

Php Mysql School Management System

Eventually, you will unconditionally discover a further experience and deed by spending more cash. yet when? complete you take that you require to acquire those every needs considering having significantly cash? Why dont you attempt to get something basic in the beginning? Thats something that will lead you to comprehend even more all but the globe, experience, some places, in the same way as history, amusement, and a lot more?

It is your extremely own time to exploit reviewing habit. in the course of guides you could enjoy now is **Php Mysql School Management System** below.

[Database Management System Using Visual C# .NET: MySQL, SQL Server, and Access](#) - Vivian Siahaan 2020-09-14

Book 1: VISUAL C# .NET WITH MYSQL: A Definitive Guide to Develop Database-Oriented Desktop Applications In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will go through step by step to build a SALES database using MySQL. You will build each table and add associated data fields (along with the necessary keys and indexes). The first field in the Client table is ClientID. Enter the clien ID in the Name Field and select AutoNumber in the Data Type. You define primary key and other indexes which are useful for quick searching. ClientID is a primary field. You will define FamilyName as an index. You then will create Ordering table with three fields: OrderID, ClientID, and OrderDate. You then will create Purchase table with three fields: OrderID, ProductID, and Quantity. And you will create Product table with four fields: ProductID, Description, Price, and QtySold. Before designing Visual C# interface, you will build the relationships between four tables. The interface will be used to enter new orders into the database. The order form will be used to enter the following information into the database: order ID, order date, client ID, client's first name and family name, client's address, product information ordered. The form will have the ability to add new orders, find clients, add new clients. The completed order invoice will be

provided in a printed report. In chapter three, you will build a database management system where you can store information about valuables in your warehouse. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). The development of this Warehouse Inventory Project will be performed, as usual, in a step-by-step manner. You will first create the database. Furthermore, the interface will be built so that the user can view, edit, add, or add data records from the database. Finally, you add code to create a printable list of information from the database. In chapter four, you will build an application that can be used to track daily high and low pollutant PM2.5 and air quality level. The steps that need to be taken in building Siantar Air Quality Index (SAQI) database project are: Build and test a Visual C# interface; Create an empty database using code; and Report database. The designed interface will allow the user to enter max pollutant, min pollutant, and air quality for any date that the user chooses in a particular year. This information will be stored in a database. Graphical result of the data will be provided, along with summary information relating to the maximum value, minimum value, and mean value. You will use a tab control as the main component of the interface. The control has three tabs: one for viewing and editing data, one for viewing graph of pollutant data, and another

for viewing graph of air quality data. Each tab on this control operates like a Visual C# control panel. In chapter five, you will perform the steps necessary to build a MySQL book inventory database that contains 4 tables. You will build each table and add the associated fields as needed. You will have four tables in the database and define the relationship between the primary key and foreign key. You will associate AuthorID (foreign key) field in the Title_Author table with AuthorID (primary key) in the Author table. Then, you want to associate the ISBN (foreign key) field in Title_Author table with ISBN (primary key) in the Title table. Book 2: Visual C# .NET For Programmers: A Progressive Tutorial to Develop Desktop Applications In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will go through step by step to build a SALES database using Microsoft Access and SQL Server. You will build each table and add associated data fields (along with the necessary keys and indexes). The first field in the Client table is ClientID. Enter the client ID in the Name Field and select AutoNumber in the Data Type. You define primary key and other indexes which are useful for quick searching. ClientID is a primary field. If the small lock symbol is not displayed next to the ClientID row, then you need to place it there. Right click on ClientID row and select Primary Key. A small key is now displayed next to the entry indicating it is the primary key. You will define FamilyName as an index. Select the FamilyName line. On the General tab, set the Indexed property to Yes (Duplicates OK). You then will create Ordering table with three fields: OrderID, ClientID, and OrderDate. You then will create Purchase table with three fields: OrderID, ProductID, and Quantity. And you will create Product table with four fields: ProductID, Description, Price, and QtySold. Before designing Visual C# interface, you will build the relationships between four tables. In chapter three, you will build a Visual C# interface for the database. The interface will be used to enter new orders into the database. The order form will be used to enter the following information into the database: order

ID, order date, client ID, client's first name and family name, client's address, product information ordered. The form will have the ability to add new orders, find clients, add new clients. The completed order invoice will be provided in a printed report. In chapter four, you will build a database management system where you can store information about valuables in your warehouse. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). The development of this Warehouse Inventory Project will be performed, as usual, in a step-by-step manner. You will first create the database. Furthermore, the interface will be built so that the user can view, edit, add, or add data records from the database. Finally, you add code to create a printable list of information from the database. In chapter five, you will build an application that can be used to track daily high and low pollutant PM2.5 and air quality level. You will do this in stages, from database development to creation of distribution packages. These steps are the same as those used in developing a commercial database application. The steps that need to be taken in building Siantar Air Quality Index (SAQI) database project are: Build and test a Visual C# interface; Create an empty database using code; and Report database. The designed interface will allow the user to enter max pollutant, min pollutant, and air quality for any date that the user chooses in a particular year. This information will be stored in a database. Graphical result of the data will be provided, along with summary information relating to the maximum value, minimum value, and mean value. You will use a tab control as the main component of the interface. The control has three tabs: one for viewing and editing data, one for viewing graph of pollutant data, and another for viewing graph of air quality data. Each tab on this control operates like a Visual C# control panel. In chapter six, you will perform the steps necessary to build a SQL Server book inventory database that contains 4 tables using Microsoft

Visual Studio 2019. You will build each table and add the associated fields as needed. You will have four tables in the database and define the relationship between the primary key and foreign key. You will associate AuthorID (foreign key) field in the Title_Author table with AuthorID (primary key) in the Author table. Then, you want to associate the ISBN (foreign key) field in Title_Author table with ISBN (primary key) in the Title table.

REGRESSION, SEGMENTATION, CLUSTERING, AND PREDICTION PROJECTS WITH PYTHON - Vivian Siahaan 2022-02-25

PROJECT 1: TIME-SERIES WEATHER: FORECASTING AND PREDICTION WITH PYTHON Weather data are described and quantified by the variables of Earth's atmosphere: temperature, air pressure, humidity, and the variations and interactions of these variables, and how they change over time. Different spatial scales are used to describe and predict weather on local, regional, and global levels. The dataset used in this project contains weather data for New Delhi, India. This data was taken out from wunderground. It contains various features such as temperature, pressure, humidity, rain, precipitation, etc. The main target is to develop a prediction model accurate enough for forecasting temperature and predicting target variable (condition). Time-series weather forecasting will be done using ARIMA models. The machine learning models used in this project to predict target variable (condition) are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, LGBM classifier, Gradient Boosting, XGB classifier, and MLP classifier. Finally, you will plot boundary decision, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

PROJECT 2: HOUSE PRICE: ANALYSIS AND PREDICTION USING MACHINE LEARNING WITH PYTHON The dataset used in this project is taken from the second chapter of Aurélien Géron's recent book 'Hands-On Machine learning with Scikit-Learn and TensorFlow'. It serves as an excellent introduction to implementing machine learning

algorithms because it requires rudimentary data cleaning, has an easily understandable list of variables and sits at an optimal size between being too toyish and too cumbersome. The data contains information from the 1990 California census. Although it may not help you with predicting current housing prices like the Zillow Zestimate dataset, it does provide an accessible introductory dataset for teaching people about the basics of machine learning. The data pertains to the houses found in a given California district and some summary stats about them based on the 1990 census data. Be warned the data aren't cleaned so there are some preprocessing steps required! The columns are as follows: longitude, latitude, housing_median_age, total_rooms, total_bedrooms, population, households, median_income, median_house_value, and ocean_proximity. The machine learning models used in this project used to perform regression on median_house_value and to predict it as target variable are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, LGBM classifier, Gradient Boosting, XGB classifier, and MLP classifier. Finally, you will plot boundary decision, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

PROJECT 3: CUSTOMER PERSONALITY ANALYSIS AND PREDICTION USING MACHINE LEARNING WITH PYTHON Customer Personality Analysis is a detailed analysis of a company's ideal customers. It helps a business to better understand its customers and makes it easier for them to modify products according to the specific needs, behaviors and concerns of different types of customers. Customer personality analysis helps a business to modify its product based on its target customers from different types of customer segments. For example, instead of spending money to market a new product to every customer in the company's database, a company can analyze which customer segment is most likely to buy the product and then market the product only on that particular segment.

Following are the features in the dataset: ID = omahafoodtruckassociation.org on by guest

Customer's unique identifier; Year_Birth = Customer's birth year; Education = Customer's education level; Marital_Status = Customer's marital status; Income = Customer's yearly household income; Kidhome = Number of children in customer's household; Teenhome = Number of teenagers in customer's household; Dt_Customer = Date of customer's enrollment with the company; Recency = Number of days since customer's last purchase; MntWines = Amount spent on wine in the last 2 years; MntFruits = Amount spent on fruits in the last 2 years; MntMeatProducts = Amount spent on meat in the last 2 years; MntFishProducts = Amount spent on fish in the last 2 years; MntSweetProducts = Amount spent on sweets in the last 2 years; MntGoldProds = Amount spent on gold in the last 2 years; NumDealsPurchases = Number of purchases made with a discount; NumWebPurchases = Number of purchases made through the company's web site; NumCatalogPurchases = Number of purchases made using a catalogue; NumStorePurchases = Number of purchases made directly in stores; NumWebVisitsMonth = Number of visits to company's web site in the last month; AcceptedCmp3 = 1 if customer accepted the offer in the 3rd campaign, 0 otherwise; AcceptedCmp4 = 1 if customer accepted the offer in the 4th campaign, 0 otherwise; AcceptedCmp5 = 1 if customer accepted the offer in the 5th campaign, 0 otherwise; AcceptedCmp1 = 1 if customer accepted the offer in the 1st campaign, 0 otherwise; AcceptedCmp2 = 1 if customer accepted the offer in the 2nd campaign, 0 otherwise; Response = 1 if customer accepted the offer in the last campaign, 0 otherwise; and Complain = 1 if customer complained in the last 2 years, 0 otherwise. The target in this project is to perform clustering and predicting to summarize customer segments. In this project, you will perform clustering using KMeans to get 4 clusters. The machine learning models used in this project to perform regression on total number of purchase and to predict clusters as target variable are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, LGBM, Gradient Boosting, XGB, and MLP. Finally, you will plot boundary decision, distribution of

features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy. PROJECT 4: CUSTOMER SEGMENTATION, CLUSTERING, AND PREDICTION WITH PYTHON In this project, you will develop a customer segmentation, clustering, and prediction to define marketing strategy. The sample dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioral variables. Following is the Data Dictionary for Credit Card dataset: CUSTID: Identification of Credit Card holder (Categorical); BALANCE: Balance amount left in their account to make purchases; BALANCEFREQUENCY: How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated); PURCHASES: Amount of purchases made from account; ONEOFFPURCHASES: Maximum purchase amount done in one-go; INSTALLMENTSPURCHASES: Amount of purchase done in installment; CASHADVANCE: Cash in advance given by the user; PURCHASESFREQUENCY: How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased); ONEOFFPURCHASESFREQUENCY: How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased); PURCHASESINSTALLMENTSFREQUENCY: How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done); CASHADVANCEFREQUENCY: How frequently the cash in advance being paid; CASHADVANCETRX: Number of Transactions made with "Cash in Advanced"; PURCHASESTRX: Number of purchase transactions made; CREDITLIMIT: Limit of Credit Card for user; PAYMENTS: Amount of Payment done by user; MINIMUM_PAYMENTS: Minimum amount of payments made by user; PRCFULLPAYMENT: Percent of full payment paid by user; and TENURE: Tenure of credit card service for user. In this project, you will perform clustering using KMeans to get 5 clusters. The machine learning models used in this project to perform regression on total

number of purchase and to predict clusters as target variable are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, LGBM, Gradient Boosting, XGB, and MLP. Finally, you will plot boundary decision, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

SQLITE FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI - Vivian Siahaan 2022-06-27

In this project, you will use SQLite version of Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the SQLite database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10

sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997, amount feature over 1998, and all amount feature.

From Zero To Hero: .NET PROGRAMMING FOR STUDENTS - Vivian Siahaan 2020-12-09

Book 1: VISUAL BASIC .NET FOR STUDENTS: A Project-Based Approach to Develop Desktop Applications In chapter one, you will get to know the properties and events of each control in a Windows Visual Basic application. You need to learn and know in order to be more familiar when applying them to some desktop applications in this book. In Tutorial 1.1, you will build a dual-mode stopwatch. The stopwatch can be started and stopped whenever desired. Two time traces: the running time when the stopwatch is active (running time) and the total time since the first stopwatch was activated. Two label controls are used to display the time (two more labels to display title information). Two button controls are used to start/stop and reset the application, one more button to exit the application. The timer control is used to periodically (every second) update the displayed time. In Tutorial 1.2, you will build a project so that children can practice basic skills in addition, subtraction, multiplication, and division operations. This Math Game project can be used to choose the types of questions and what factor you want to use. This project has three timing options. In Tutorial 1.3, you will build Bank Code game. The storage box is locked and can only be opened if you enter the correct digit combination. Combinations can be 2 to 4 non-repetitive digits (range of digits from 1 to 9). After a guess is given, you will be notified of how many digits are right and how many digits are in the right position. Based on this information, you will give another guess. You continue to guess until you get the right combination or until you stop the game. In Tutorial 1.4, you will build Horse Racing game. This is a simple game. Up to 10 horses will race to the finish line. You guessed two horses that you thought could win the race. By clicking on the Start button, the race will start. All horses will race speed to get to the finish line. In chapter two, you will learn

the basic concepts of classes and objects. Next, it will demonstrate how to define class and type of enumeration, which shows how both are used in the application. In Tutorial 2.1, you will create a two-level application that uses a form to pass input user to the People class. The form class is the level of representation and the People class is the middle level. You will add controls to the form so people can enter ID, last name, and their height. When the user clicks the Save button, the code will assign input values to the People class properties. Finally, you will display the People object on a label. Figure below shows the form after the user clicks the Save button. In Tutorial 2.2, you will add a parameterized constructor to the People class. The application will ask the user to enter values, which will then be passed to the People constructor. Then, the application will display the values stored on the People object. In Tutorial 2.3, you will create an application that utilizes enumeration type. The user will choose one type of account that is listed in a ListBox control and what he chooses is then displayed in a Label control. In Tutorial 2.4, you will create a simple Bank application. This application has one class, BankAcc, and a startup form. In Tutorial 2.5, you will improve the simple Bank application, by implementing the following two properties in the BankAcc class: TotalDeposit- Total money saved in current account; TotalWithdraw- Total funds that have been withdrawn from current account. In Tutorial 2.6, you will create an application to calculate the time needed for a particular aircraft to reach takeoff speed. You will also calculate how long the runway will be required. For each type of aircraft, you are given (1) the name of the aircraft, (2) the required take-off speed (feet/sec), and (3) how fast the plane accelerates (feet/sec²). In Tutorial 2.7, you will provide a number of programming training for those who want to improve their programming skills. Your task here is to write an object-oriented application so that training manager can display and edit the training services offered. There are several training categories: (1) Application Development, (2) Database, (3) Networking, and (4) System Administration. The training itself consists of: (1) title, (2) training days, (3) category, and (4) cost. Create a class named Training that contains this information,

along with its properties and a ToString() method. In chapter three, several tutorials will be presented to build more complex projects. You will build them gradually and step by step. In Tutorial 3.1, you will build Catching Ball game. The bird flew and dropped ball from the sky. User is challenged to position man under the fallen ball to catch it. In Tutorial 3.2, you will build Smart Tic Tac Toe game. The aim of this game is to win the game on a 3 x 3 grid with the victory of three identical symbols (X or O) on horizontal, diagonal, or vertical lines. The players will play alternately. In this game given two game options: player 1 against player 2 or human player against computer. A smart but simple strategy will be developed for computer logic to be a formidable opponent for human. In Tutorial 3.3, you will build a Matching Images game. Ten pairs of images hidden on the game board. The object of the game is to find image pairs. In Two Players mode, players will get turns in turn. In One Player mode, there are two options to choose from: Playing Alone or Against Computer. When Play Alone option is selected, the player will play alone without an opponent. If Against Computer option is selected, then the level of computer intelligence is given with several levels according to the level of difficulty of the game. In Tutorial 3.4, you will build Throwing Fire program. This program can be played by two human players or human player versus computer. In chapter four, tutorials will be presented to build two advanced projects. You will build them gradually and step by step. In Tutorial 4.1, you will build Roasted Duck Delivery simulation. In this simulation, a number of decisions are needed. The basic idea is to read the order by incoming telephone and tell the delivery scooter to go to the location of the order. You also need to make sure that you always provide a roasted duck ready to be transported by the delivery scooter. The delivery area is a 20 by 20 square grid. The more roasted duck is sold, the more profit it gets. In Tutorial 4.2, you will build a Drone Simulation. In this simulation, you control both vertical and horizontal thrusters to maneuver the ride to the landing pad. You will adjust the landing speed so that it is slow enough so that no accident occurs.

Applications In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# applications. You need to learn and know in order to be more familiar when applying them to some desktop applications in this book. In chapter two, you will build Throwing Fire program. This program can be played by two human players or human player versus computer. You will use 12 labels, a large control panel, and three control buttons on the form. In the control panel, a smaller panel with two group box controls and a button control are placed. In the first group box, you will use 2 radio buttons; in the second box group, place 4 radio buttons. Next, two timer controls are added to the project. All label controls are used for titles and provide scoring and game information. The large panel (Panel1) is the playing field. Three button controls are used to start / stop a program, set options, and exit the program. One timer control is used to control game animation and another is used to represent the computer's decision process. The second control panel (Panel2) is used to select game options. One group box contains radio buttons which are used to select number of players. A group box contains radio buttons to select the level of difficulty of the game, when playing against a computer. A small button is used to close the options panel. The default properties are set for one-player games with the easiest game difficulty. In chapter three, you will build Roasted Duck Delivery simulation. In this simulation, a number of decisions are needed. The basic idea is to read the order by incoming telephone and tell the delivery scooter to go to the location of the order. You also need to make sure that you always provide a roasted duck ready to be transported by the delivery scooter. The delivery area is a 20 by 20 square grid. The more roasted duck is sold, the more profit it gets. The panel control on the left side of the form contains the delivery grid. On the upper right are group boxes with two label controls to display the time or hour and sale results. The computer monitor (in a picture box) displays order and delivery status using a list box and label control. Another group box contains a roasting oven when the roasted ducks are displayed using eight picture box controls. Two button controls on the group box control the

operation of the oven. Group boxes under the oven show how many ducks are ready to be delivered and how many are in the delivery scooter (a button control is to load the roasted duck into the scooter). The two button controls beneath are used to start/pause the game and to stop the game or exit the game. In the area under the form there are several timers for controlling a number of aspects in the program. The delivery grid consists of 400 label controls on 20 rows (marked with numbers) and 20 columns (marked with letters). Here, you will learn how to place controls on a form (or panel in this case) using code (when the program runs, not when designing the form). This mechanism can save time designing the form. In chapter four, you will build a Drone Simulation. In this simulation, you control both vertical and horizontal thrusters to maneuver the ride to the landing pad. You will adjust the landing speed so that it is slow enough so that no accident occurs. You build the form in two stages, the first stage creates two option group boxes, and then the second stage uses both those group boxes as landing controls. Two control panels are placed on the left side of the form: one panel for drawing and another panel for the edge. On the right side of the form, place the two group control boxes. In the first group box, five radio buttons and a check box are added. In the second group box, two radio buttons are placed. In the below section of the form, three buttons are added. Finally, one timer control is added. Then in the form, a group box is added overlap panel. Then, 11 label controls are added to the group box. After that, a progress bar is added. Under the bar, two control panels are added, one high panel and one short panel. In the second (short) panel control, two small label controls are added. Underneath, three button controls are placed. Under these three buttons, a label control is added. For each label control, set the AutoSize property to False to be resized and set (temporarily) the BorderStyle property to FixedSingle so that you can see the edges to facilitate the layout process. In this chapter, you will build Jumper game. In this game, you will move the jumper across the busy road, avoid the tiger, and cross the river with the changing current to get to house safely. You will place four label controls on the top part of the form.

(set the `AutoSize` property to `False` so that it can be resized and the `BorderStyle` property temporarily becomes `FixedSingle` so you can see the edges). Then, you use five panel controls below the labels. These panels will be a place for image graphics. Each panel has a width of 16 jumpers or 640 pixels, because one jumper will be given a width of 40 pixels. The first panel will be the jumper house, which will be given a height of 80 pixels. The next panel will become a river, with a height of 120 pixels. The next panel will be a place for tiger, 40 pixels high. Under the snake panel, there is a road panel. This panel will contain three boat lanes. Each boat has a height of 40 pixels, but you will give it a height of 140 pixels (not 120 pixels) to make room for lane markers. The fifth panel is the place where the jumper will begin its journey or leap. This panel will be given a height of 40 pixels. Add the last control panel below the form with three button controls. Then, finally, add four timer controls. Adjust the size of the form so that the panel controls can occupy according to the width of the form.

BOOK 3: VISUAL C# .NET : A Step By Step, Project-Based Guide to Develop Desktop Applications

In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will build a project so that children can practice basic skills in addition, subtraction, multiplication, and division operations. This Math Game project can be used to choose the types of questions and what factors you want to use. This project has three timing options. Random math problems using values from 0 to 9 will be presented. Timing options are provided to measure accuracy and speed. There are many controls used. Two label controls are used for title information, two for displaying scores. There is a wide label in the middle of the form to display math questions. And, long skinny label is used as separator. Two button controls are used to start and stop question and one button to exit the project. There are three group control boxes. The first group box holds four check box controls that are used to select the type of questions. The second group box holds eleven radio buttons that are used to select values that are used as

factors in calculations. The third group box contains three radio button controls for timing options. A scroll bar control rod is used to change the time. In chapter three, you will build Bank Code game. The storage box is locked and can only be opened if you enter the correct digit combination. Combinations can be 2 to 4 non-repetitive digits (range of digits from 1 to 9). After a guess is given, you will be notified of how many digits are right and how many digits are in the right position. Based on this information, you will give another guess. You continue to guess until you get the right combination or until you stop the game. On the left side of the form is a large picture box control. On the right side, two group box controls and two button controls are placed. In the picture box, a control panel is placed. In the panel, there are four label controls (set the `AutoSize` property to `False`) and nine button controls. In the first group box control, place three radio buttons. In the second group box control, a text box control is placed. The picture box contains an image of bank and a panel. The label controls in the panel are used to display the combinations entered (the `BorderStyle` property set to `FixedSingle` to display the label size). The nine buttons on the panel are used to enter combinations. Radio buttons are used to set options. The buttons (one to start and stop the game and another to exit the project) are used to control game operations. The text box displays the results of the combinations entered. In chapter four, you will build Horse Racing game. This is a simple game. Up to 10 horses will race to the finish line. You guessed two horses that you thought could win the race. By clicking on the Start button, the race will start. All horses will race speed to get to the finish line. Labels are used to display instructions and number of horses in a race. Four button controls are used: two buttons to change number of horses, one button to start the game, and one other button to stop the game. The picture box control is used to load the horse image. A timer control is used to update the horse's movement during the race. In chapter five, you will build Catching Ball game. The bird flew and dropped ball from the sky. Users are challenged to position man under the fallen ball to catch it. Labels are used for instructions and to display game information (remaining time,

number of balls captured, and game difficulty level). Two buttons are used to change the game difficulty level, one button to start the game, and another button to stop the game. Picture box controls hold images for man, bird, and ball. In chapter six, you will build Smart Tic Tac Toe game. That said, this is the first game ever programmed on a computer and one that had been programmed by Bill Gates himself when he was a teenager while attending Lakeside School in Seattle. The aim of this game is to win the game on a 3 x 3 grid with the victory of three identical symbols (X or O) on horizontal, diagonal, or vertical lines. The players will play alternately. In this game given two game options: player 1 against player 2 or human player against computer. A smart but simple strategy will be developed for computer logic to be a formidable opponent for humans. In chapter seven, you will build Fighting Plane program. This program can be played by two human players or human player versus computer. The controls of the player are done via the keyboard. Player 1 presses A key to move up, Z key to move down, and S key to throw rudal. When you choose Two players from the Options button, this game can be played by two human players. Player 1 presses the same keys, while player 2 presses key K to move up, M to move down, and key J to throw rudal. All label controls are used for titles and provide scoring and game information. The large panel (Panel1) is the playing field. Three button controls are used to start / stop a program, set options, and exit the program. One timer control is used to control game animation and another is used to represent the computer's decision process. The second control panel (Panel2) is used to select game options. One group box contains radio buttons which are used to select number of players. A group box contains radio buttons to select the level of difficulty of the game, when playing against a computer. A small button is used to close the options panel. The default properties are set for one-player games with the easiest game difficulty.

TIME-SERIES WEATHER: FORECASTING AND PREDICTION WITH PYTHON - Vivian Siahaan 2022-02-18

Weather data are described and quantified by the variables of Earth's atmosphere:

temperature, air pressure, humidity, and the variations and interactions of these variables, and how they change over time. Different spatial scales are used to describe and predict weather on local, regional, and global levels. The dataset used in this project contains weather data for New Delhi, India. This data was taken out from wunderground. It contains various features such as temperature, pressure, humidity, rain, precipitation, etc. The main target is to develop a prediction model accurate enough for forecasting temperature and predicting target variable (condition). Time-series weather forecasting will be done using ARIMA models. The machine learning models used in this project to predict target variable (condition) are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, LGBM classifier, Gradient Boosting, XGB classifier, and MLP classifier. Finally, you will plot boundary decision, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

Hands-On Guide To IMAGE

CLASSIFICATION Using Scikit-Learn, Keras, And TensorFlow with PYTHON GUI - Vivian Siahaan 2021-05-27

In this book, implement deep learning-based image classification on detecting face mask, classifying weather, and recognizing flower using TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries. In Chapter 1, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform detecting face mask using Face Mask Detection Dataset provided by Kaggle

(<https://www.kaggle.com/omkargurav/face-mask-dataset/download>). In Chapter 2, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform how to classify weather using Multi-class Weather Dataset provided by Kaggle (<https://www.kaggle.com/pratik2901/multiclass-weather-dataset/download>). In Chapter 3, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other

Downloaded from
omahafoodtruckassociation.org on by
guest

libraries to perform how to recognize flower using Flowers Recognition dataset provided by Kaggle (<https://www.kaggle.com/alxmamaev/flowers-recognition/download>).

DATA SCIENCE WORKSHOP: Liver Disease Classification and Prediction Using Machine Learning and Deep Learning with Python GUI - Vivian Siahaan 2021-11-25

Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors. This dataset contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90". Columns in the dataset: Age of the patient; Gender of the patient; Total Bilirubin; Direct Bilirubin; Alkaline Phosphatase; Alamine Aminotransferase; Aspartate Aminotransferase; Total Protiens; Albumin; Albumin and Globulin Ratio; and Dataset: field used to split the data into two sets (patient with liver disease, or no disease). The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

THE APPLIED DATA SCIENCE WORKSHOP: Urinary biomarkers Based Pancreatic Cancer Classification and Prediction Using Machine Learning with Python GUI - Vivian Siahaan 2021-12-04

Pancreatic cancer is an extremely deadly type of cancer. Once diagnosed, the five-year survival rate is less than 10%. However, if pancreatic

cancer is caught early, the odds of surviving are much better. Unfortunately, many cases of pancreatic cancer show no symptoms until the cancer has spread throughout the body. A diagnostic test to identify people with pancreatic cancer could be enormously helpful. In a paper by Silvana Debernardi and colleagues, published this year in the journal PLOS Medicine, a multi-national team of researchers sought to develop an accurate diagnostic test for the most common type of pancreatic cancer, called pancreatic ductal adenocarcinoma or PDAC. They gathered a series of biomarkers from the urine of three groups of patients: Healthy controls, Patients with non-cancerous pancreatic conditions, like chronic pancreatitis, and Patients with pancreatic ductal adenocarcinoma. When possible, these patients were age- and sex-matched. The goal was to develop an accurate way to identify patients with pancreatic cancer. The key features are four urinary biomarkers: creatinine, LYVE1, REG1B, and TFF1. Creatinine is a protein that is often used as an indicator of kidney function. YVLE1 is lymphatic vessel endothelial hyaluronan receptor 1, a protein that may play a role in tumor metastasis. REG1B is a protein that may be associated with pancreas regeneration. TFF1 is trefoil factor 1, which may be related to regeneration and repair of the urinary tract. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, and MLP classifier. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

HOUSEHOLD ELECTRIC POWER CONSUMPTION: ANALYSIS, CLUSTERING, AND PREDICTION WITH PYTHON - Vivian Siahaan 2022-03-03

In this project, you will perform analysis, clustering, and prediction on household electric power consumption with python. The dataset used in this project contains 2075259 measurements gathered between December 2006 and November 2010 (47 months).

Following are the attributes in the dataset: date: Date in format dd/mm/yyyy; time: time in format hh:mm:ss; globalactivepower: household global minute-averaged active power (in kilowatt); globalreactivepower: household global minute-averaged reactive power (in kilowatt); voltage: minute-averaged voltage (in volt); global_intensity: household global minute-averaged current intensity (in ampere); submetering1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered); submetering2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light; and submetering3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner. In this project, you will perform clustering using KMeans to get 5 clusters. The machine learning models used in this project to perform regression on total number of purchase and to predict clusters as target variable are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, LGBM, Gradient Boosting, XGB, and MLP. Finally, you will plot boundary decision, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

The Applied Data Science Workshop On Medical Datasets Using Machine Learning and Deep Learning with Python GUI - Vivian Siahaan

Workshop 1: Heart Failure Analysis and Prediction Using Scikit-Learn, Keras, and TensorFlow with Python GUI Cardiovascular diseases (CVDs) are the number 1 cause of death globally taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as

hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning models can be of great help. Dataset used in this project is from Davide Chicco, Giuseppe Jurman. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). Attribute information in the dataset are as follows: age: Age; anaemia: Decrease of red blood cells or hemoglobin (boolean); creatinine_phosphokinase: Level of the CPK enzyme in the blood (mcg/L); diabetes: If the patient has diabetes (boolean); ejection_fraction: Percentage of blood leaving the heart at each contraction (percentage); high_blood_pressure: If the patient has hypertension (boolean); platelets: Platelets in the blood (kiloplatelets/mL); serum_creatinine: Level of serum creatinine in the blood (mg/dL); serum_sodium: Level of serum sodium in the blood (mEq/L); sex: Woman or man (binary); smoking: If the patient smokes or not (boolean); time: Follow-up period (days); and DEATH_EVENT: If the patient deceased during the follow-up period (boolean). The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy. WORKSHOP 2: Cervical Cancer Classification and Prediction Using Machine Learning and Deep Learning with Python GUI About 11,000 new cases of invasive cervical cancer are diagnosed each year in the U.S. However, the number of new cervical cancer cases has been declining steadily over the past decades. Although it is the most preventable type of cancer, each year cervical cancer kills about 4,000 women in the U.S. and about 300,000 women worldwide. Numerous studies report that high poverty levels are linked with low screening rates. In addition, lack of

health insurance, limited transportation, and language difficulties hinder a poor woman's access to screening services. Human papilloma virus (HPV) is the main risk factor for cervical cancer. In adults, the most important risk factor for HPV is sexual activity with an infected person. Women most at risk for cervical cancer are those with a history of multiple sexual partners, sexual intercourse at age 17 years or younger, or both. A woman who has never been sexually active has a very low risk for developing cervical cancer. Sexual activity with multiple partners increases the likelihood of many other sexually transmitted infections (chlamydia, gonorrhea, syphilis). Studies have found an association between chlamydia and cervical cancer risk, including the possibility that chlamydia may prolong HPV infection. Therefore, early detection of cervical cancer using machine and deep learning models can be of great help. The dataset used in this project is obtained from UCI Repository and kindly acknowledged. This file contains a List of Risk Factors for Cervical Cancer leading to a Biopsy Examination. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

WORKSHOP 3: Chronic Kidney Disease Classification and Prediction Using Machine Learning and Deep Learning with Python GUI Chronic kidney disease is the longstanding disease of the kidneys leading to renal failure. The kidneys filter waste and excess fluid from the blood. As kidneys fail, waste builds up. Symptoms develop slowly and aren't specific to the disease. Some people have no symptoms at all and are diagnosed by a lab test. Medication helps manage symptoms. In later stages, filtering the blood with a machine (dialysis) or a transplant may be required. The dataset used in this project was taken over a 2-month period in India with 25 features (eg, red

blood cell count, white blood cell count, etc). The target is the 'classification', which is either 'ckd' or 'notckd' - ckd=chronic kidney disease. It contains measures of 24 features for 400 people. Quite a lot of features for just 400 samples. There are 14 categorical features, while 10 are numerical. The dataset needs cleaning: in that it has NaNs and the numeric features need to be forced to floats. Attribute Information: Age(numerical) age in years; Blood Pressure(numerical) bp in mm/Hg; Specific Gravity(categorical) sg - (1.005,1.010,1.015,1.020,1.025); Albumin(categorical) al - (0,1,2,3,4,5); Sugar(categorical) su - (0,1,2,3,4,5); Red Blood Cells(categorical) rbc - (normal,abnormal); Pus Cell (categorical) pc - (normal,abnormal); Pus Cell clumps(categorical) pcc - (present, notpresent); Bacteria(categorical) ba - (present,notpresent); Blood Glucose Random(numerical) bgr in mgs/dl; Blood Urea(numerical) bu in mgs/dl; Serum Creatinine(numerical) sc in mgs/dl; Sodium(numerical) sod in mEq/L; Potassium(numerical) pot in mEq/L; Hemoglobin(numerical) hemo in gms; Packed Cell Volume(numerical); White Blood Cell Count(numerical) wc in cells/cumm; Red Blood Cell Count(numerical) rc in millions/cmm; Hypertension(categorical) htn - (yes,no); Diabetes Mellitus(categorical) dm - (yes,no); Coronary Artery Disease(categorical) cad - (yes,no); Appetite(categorical) appet - (good,poor); Pedal Edema(categorical) pe - (yes,no); Anemia(categorical) ane - (yes,no); and Class (categorical) class - (ckd,notckd). The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

WORKSHOP 4: Lung Cancer Classification and Prediction Using Machine Learning and Deep Learning with Python GUI The effectiveness of

cancer prediction system helps the people to know their cancer risk with low cost and it also helps the people to take the appropriate decision based on their cancer risk status. The data is collected from the website online lung cancer prediction system. Total number of attributes in the dataset is 16, while number of instances is 309. Following are attribute information of dataset: Gender: M(male), F(female); Age: Age of the patient; Smoking: YES=2 , NO=1; Yellow fingers: YES=2 , NO=1; Anxiety: YES=2 , NO=1; Peer_pressure: YES=2 , NO=1; Chronic Disease: YES=2 , NO=1; Fatigue: YES=2 , NO=1; Allergy: YES=2 , NO=1; Wheezing: YES=2 , NO=1; Alcohol: YES=2 , NO=1; Coughing: YES=2 , NO=1; Shortness of Breath: YES=2 , NO=1; Swallowing Difficulty: YES=2 , NO=1; Chest pain: YES=2 , NO=1; and Lung Cancer: YES , NO. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy. WORKSHOP 5: Alzheimer's Disease Classification and Prediction Using Machine Learning and Deep Learning with Python GUI Alzheimer's is a type of dementia that causes problems with memory, thinking and behavior. Symptoms usually develop slowly and get worse over time, becoming severe enough to interfere with daily tasks. Alzheimer's is not a normal part of aging. The greatest known risk factor is increasing age, and the majority of people with Alzheimer's are 65 and older. But Alzheimer's is not just a disease of old age. Approximately 200,000 Americans under the age of 65 have younger-onset Alzheimer's disease (also known as early-onset Alzheimer's). The dataset consists of a longitudinal MRI data of 374 subjects aged 60 to 96. Each subject was scanned at least once. Everyone is right-handed. 206 of the subjects were grouped as 'Nondemented' throughout the study. 107 of the subjects were grouped as 'Demented' at the time

of their initial visits and remained so throughout the study. 14 subjects were grouped as 'Nondemented' at the time of their initial visit and were subsequently characterized as 'Demented' at a later visit. These fall under the 'Converted' category. Following are some important features in the dataset: EDUC:Years of Education; SES: Socioeconomic Status; MMSE: Mini Mental State Examination; CDR: Clinical Dementia Rating; eTIV: Estimated Total Intracranial Volume; nWBV: Normalize Whole Brain Volume; and ASF: Atlas Scaling Factor. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy. WORKSHOP 6: Parkinson Classification and Prediction Using Machine Learning and Deep Learning with Python GUI The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD. The data is in ASCII CSV format. The rows of the CSV file contain an instance corresponding to one voice recording. There are around six recordings per patient, the name of the patient is identified in the first column. Attribute information of this dataset are as follows: name - ASCII subject name and recording number; MDVP:Fo(Hz) - Average vocal fundamental frequency; MDVP:Fhi(Hz) -

Maximum vocal fundamental frequency; MDVP:Flo(Hz) - Minimum vocal fundamental frequency; MDVP:Jitter(%); MDVP:Jitter(Abs); MDVP:RAP; MDVP:PPQ; Jitter:DDP - Several measures of variation in fundamental frequency; MDVP:Shimmer; MDVP:Shimmer(dB); Shimmer:APQ3; Shimmer:APQ5; MDVP:APQ; Shimmer:DDA - Several measures of variation in amplitude; NHR; HNR - Two measures of ratio of noise to tonal components in the voice; status - Health status of the subject (one) - Parkinson's, (zero) - healthy; RPDE,D2 - Two nonlinear dynamical complexity measures; DFA - Signal fractal scaling exponent; and spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

WORKSHOP 7: Liver Disease Classification and Prediction Using Machine Learning and Deep Learning with Python GUI Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors. This dataset contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90". Columns in the dataset: Age of the patient; Gender of the patient; Total Bilirubin; Direct Bilirubin; Alkaline Phosphatase; Alamine Aminotransferase; Aspartate Aminotransferase; Total Proteins; Albumin; Albumin and Globulin Ratio; and Dataset: field used to split the data

into two sets (patient with liver disease, or no disease). The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

Step by Step Tutorial IMAGE CLASSIFICATION Using Scikit-Learn, Keras, And TensorFlow with PYTHON GUI - Vivian Siahaan 2021-06-03

In this book, implement deep learning-based image classification on classifying monkey species, recognizing rock, paper, and scissor, and classify airplane, car, and ship using TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries. In Chapter 1, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform how to classify monkey species using 10 Monkey Species dataset provided by Kaggle (<https://www.kaggle.com/slothkong/10-monkey-species/download>). In Chapter 2, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform how to recognize rock, paper, and scissor using 10 Monkey Species dataset provided by Kaggle (<https://www.kaggle.com/sanikamal/rock-paper-scissors-dataset/download>). In Chapter 3, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform how to classify airplane, car, and ship using Multiclass-image-dataset-airplane-car-ship dataset provided by Kaggle (<https://www.kaggle.com/abtabm/multiclassimage-datasetairplanecar>).

Access Database with JDBC - Vivian Siahaan 2019-10-12

This is a Java GUI crash course. This book will help you quickly write efficient, high-quality access-database-driven code with Java. It's an ideal way to begin, whether you're new to

programming or a professional developer versed in other languages. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch two access database management systems using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will create School database and six tables. In chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create Crime database and its tables. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eighth, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter nine, you will add two tables: Police and

Investigator. These two tables will later be joined to Suspect table through another table, Case_File, which will be built in the seventh chapter. The Police has six columns: police_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and Case_File. The Case_File table will connect four other tables: Suspect, Police, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

DATA SCIENCE WORKSHOP: Lung Cancer Classification and Prediction Using Machine Learning and Deep Learning with Python GUI - Vivian Siahaan 2021-11-14

The effectiveness of cancer prediction system helps the people to know their cancer risk with low cost and it also helps the people to take the appropriate decision based on their cancer risk status. The data is collected from the website online lung cancer prediction system. Total number of attributes in the dataset is 16, while number of instances is 309. Following are attribute information of dataset: Gender: M(male), F(female); Age: Age of the patient; Smoking: YES=2 , NO=1; Yellow fingers: YES=2 , NO=1; Anxiety: YES=2 , NO=1; Peer_pressure: YES=2 , NO=1; Chronic Disease: YES=2 , NO=1; Fatigue: YES=2 , NO=1; Allergy: YES=2 , NO=1; Wheezing: YES=2 , NO=1; Alcohol: YES=2 , NO=1; Coughing: YES=2 , NO=1; Shortness of Breath: YES=2 , NO=1; Swallowing Difficulty: YES=2 , NO=1; Chest pain: YES=2 , NO=1; and Lung Cancer: YES , NO. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and

CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

A Heuristic Database Programming with PHP and PostgreSQL - Vivian Siahaan 2019-04-16
PostgreSQL was designed to run on UNIX-like platforms. However, PostgreSQL was then also designed to be portable so that it could run on various platforms such as Mac OS X, Solaris, and Windows. PostgreSQL is free and open source software. Its source code is available under PostgreSQL license, a liberal open source license. You are free to use, modify and distribute PostgreSQL in any form. PostgreSQL requires very minimum maintained efforts because of its stability. Therefore, if you develop applications based on PostgreSQL, the total cost of ownership is low in comparison with other database management systems. In Chapter 2, you will learn: Connecting to a PostgreSQL database - shows you how to setup a simple PHP application structure and connect to a PostgreSQL database; Creating new PostgreSQL database tables - walks you through the steps of creating database tables in PostgreSQL using PHP; Inserting data into PostgreSQL tables - guides you how to use insert data into a table using PHP PDO; Updating data in the table - provides you with the steps of updating data in the database tables; Querying data from a table - shows you various ways to query data in the PostgreSQL database from PHP; Performing transactions - explains the transaction concept and shows you how to perform transactions in PHP; Working with the binary large objects (BLOB) - shows you how to insert, select, and delete the large objects in the PostgreSQL using PHP; Calling PostgreSQL stored procedures - explains you the steps of calling PostgreSQL stored procedures from PHP; Deleting data in a PostgreSQL table using PHP PDO - teaches you how to delete data from the PostgreSQL table in the PHP application using PDO. In Chapter 3, you will learn managing table structure and views including postgresql data types, postgresql create table, postgresql select into statement, postgresql create table as, using

postgresql serial to create auto-increment column, identity column, alter table, drop table, truncate table, check constraint, not-null constraint, foreign key, primary key, unique constraint, managing postgresql views, creating updatable views, materialized views, creating updatable views using the with check option clause, and recursive view. In Chapter 4, you will learn statements, operators, and clauses including select, order by, select distinct, limit, fetch, in, between, postgresql like, is null, alias, joins, inner join, postgresql left join, self-join, full outer join, cross join, natural join, group by, having, intersect operator, except operator, grouping sets, cube, and rollup. In Chapter 5, you will learn postgresql trigger, aggregate, and string functions including creating the first trigger in postgresql, managing postgresql trigger, aggregate functions, avg function, max function, min function, sum function, postgresql concat function, ascii function, trim function, length function, substring function, regexp_matches function, regexp_replace function, replace function, to_number function, and to_char function.

PHP Mysql For Advanced Learning - Hirdesh Bhardwaj 2021-01-07

[FIVE PROJECTS: POSTGRESQL AND PYTHON GUI FOR DATA ANALYSIS](#) - Vivian Siahaan 2022-11-03

PROJECT 1: ZERO TO MASTERY: THE COMPLETE GUIDE TO LEARNING POSTGRESQL WITH PYTHON GUI This book uses the PostgreSQL version of MySQL-based Northwind database. The Northwind database is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind database has since been ported to a variety of non-Microsoft databases, including PostgreSQL. The Northwind dataset includes sample data for the

Downloaded from
omahafoodtruckassociation.org on by
guest

following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; and Orders and Order_Details: Sales Order transactions taking place between the customers & the company. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997, amount feature over 1998, and all amount feature. PROJECT 2: FULL SOURCE CODE: POSTGRESQL AND DATA SCIENCE FOR PROGRAMMERS WITH PYTHON GUI This project uses the PostgreSQL version of MySQL-based Sakila sample database which is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, film_actor, customer, rental, payment and inventory among others. You can download the database from <https://dev.mysql.com/doc/sakila/en/>. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year,

month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue costumers; plot which store have most sales; plot average payment amount by month with mean and EWM; and plot payment amount over June 2005. PROJECT 3: FULL SOURCE CODE: POSTGRESQL FOR DATA ANALYTICS AND VISUALIZATION WITH PYTHON GUI In this project, we provide you with a PostgreSQL version of an Oracle sample database named OT which is based on a global fictitious company that sells computer hardware including storage, motherboard, RAM, video card, and CPU. The company maintains the product information such as name, description standard cost, list price, and product line. It also tracks the inventory information for all products including warehouses where products are available. Because the company operates globally, it has warehouses in various locations around the world. The company records all customer information including name, address, and website. Each customer has at least one contact person with detailed information including name, email, and phone. The company also places a credit limit on each customer to limit the amount that customer can owe. Whenever a customer issues a purchase order, a sales order is created in the database with the pending status. When the company ships the order, the order status becomes shipped. In case the customer cancels an order, the order status becomes canceled. In addition to the sales information, the employee data is recorded with some basic information such as name, email, phone, job title, manager, and hire date. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10

sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by status, top 10 sales by status, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2016, amount feature over 2017, and amount payment in all years. PROJECT 4: FULL SOURCE CODE: POSTGRESQL FOR DATA SCIENTISTS AND DATA ANALYSTS WITH PYTHON GUI In this project, we will use the PostgreSQL version of SQL Server based BikeStores as a sample database to help you work with PostgreSQL quickly and effectively. The detailed structure of database can be found at:

<https://www.sqlservertutorial.net/sql-server-sample-database/>. The stores table includes the store's information. Each store has a store name, contact information such as phone and email, and an address including street, city, state, and zip code. The staffs table stores the essential information of staffs including first name, last name. It also contains the communication information such as email and phone. A staff works at a store specified by the value in the store_id column. A store can have one or more staffs. A staff reports to a store manager specified by the value in the manager_id column. If the value in the manager_id is null, then the staff is the top manager. If a staff no longer works for any stores, the value in the active column is set to zero. The categories table stores the bike's categories such as children bicycles, comfort bicycles, and electric bikes. The products table stores the product's information such as name, brand, category, model year, and list price. Each product belongs to a brand specified by the brand_id column. Hence, a brand may have zero or many products. Each product also belongs a category specified by the category_id column. Also, each category may have zero or many products. The customers table stores customer's information including first name, last name, phone, email, street, city, state, zip code, and photo path. The orders table stores the sales order's header information including customer, order status, order date, required date, shipped date. It also stores the

information on where the sales transaction was created (store) and who created it (staff). Each sales order has a row in the sales_orders table. A sales order has one or many line items stored in the order_items table. The order_items table stores the line items of a sales order. Each line item belongs to a sales order specified by the order_id column. A sales order line item includes product, order quantity, list price, and discount. The stocks table stores the inventory information i.e. the quantity of a particular product in a specific store. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by brand, top 10 sales by brand, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2017, amount feature over 2018, and all amount feature. PROJECT 5: FULL SOURCE CODE: THE COMPLETE GUIDE TO LEARNING POSTGRESQL AND DATA SCIENCE WITH PYTHON GUI In this project, we provide you with the PostgreSQL version of SQLite sample database named chinook. The chinook sample database is a good database for practicing with SQL, especially PostgreSQL. The detailed description of the database can be found on: <https://www.sqlitetutorial.net/sqlite-sample-database/>. The sample database consists of 11 tables: The employee table stores employees data such as employee id, last name, first name, etc. It also has a field named ReportsTo to specify who reports to whom; customers table stores customers data; invoices & invoice_items tables: these two tables store invoice data. The invoice table stores invoice header data and the invoice_items table stores the invoice line items data; The artist table stores artists data. It is a simple table that contains only the artist id and

name; The album table stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums; The media_type table stores media types such as MPEG audio and AAC audio files; genre table stores music types such as rock, jazz, metal, etc; The track table stores the data of songs. Each track belongs to one album; playlist & playlist_track tables: The playlist table store data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The relationship between the playlist table and track table is many-to-many. The playlist_track table is used to reflect this relationship. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the bottom/top 10 sales by employee, the bottom/top 10 sales by customer, the bottom/top 10 sales by customer, the bottom/top 10 sales by artist, the bottom/top 10 sales by genre, the bottom/top 10 sales by play list, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the payment amount by month with mean and EWM, the average payment amount by every month, and amount payment in all years.

SQLITE FOR DATA ANALYST AND DATA SCIENTIST WITH PYTHON GUI - Vivian Siahaan 2022-07-08

In this project, we will use the SQLite version of BikeStores database as a sample database to help you work with MySQL quickly and effectively. The stores table includes the store's information. Each store has a store name, contact information such as phone and email, and an address including street, city, state, and zip code. The staffs table stores the essential information of staffs including first name, last name. It also contains the communication information such as email and phone. A staff works at a store specified by the value in the store_id column. A store can have one or more staffs. A staff reports to a store manager specified by the value in the manager_id column. If the value in the manager_id is null, then the

staff is the top manager. If a staff no longer works for any stores, the value in the active column is set to zero. The categories table stores the bike's categories such as children bicycles, comfort bicycles, and electric bikes. The products table stores the product's information such as name, brand, category, model year, and list price. Each product belongs to a brand specified by the brand_id column. Hence, a brand may have zero or many products. Each product also belongs a category specified by the category_id column. Also, each category may have zero or many products. The customers table stores customer's information including first name, last name, phone, email, street, city, state, zip code, and photo path. The orders table stores the sales order's header information including customer, order status, order date, required date, shipped date. It also stores the information on where the sales transaction was created (store) and who created it (staff). Each sales order has a row in the sales_orders table. A sales order has one or many line items stored in the order_items table. The order_items table stores the line items of a sales order. Each line item belongs to a sales order specified by the order_id column. A sales order line item includes product, order quantity, list price, and discount. The stocks table stores the inventory information i.e. the quantity of a particular product in a specific store. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by brand, top 10 sales by brand, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2017, amount feature over 2018, and all amount feature.

VISUAL C# .NET FOR STUDENTS: A

Downloaded from
omahafoodtruckassociation.org on by
 guest

Project-Based Approach to Develop Desktop Applications - Vivian Siahaan 2020-11-23

In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# applications. You need to learn and know in order to be more familiar when applying them to some desktop applications in this book. In chapter two, you will build Throwing Fire program. This program can be played by two human players or human player versus computer. You will use 12 labels, a large control panel, and three control buttons on the form. In the control panel, a smaller panel with two group box controls and a button control are placed. In the first group box, you will use 2 radio buttons; in the second box group, place 4 radio buttons. Next, two timer controls are added to the project. All label controls are used for titles and provide scoring and game information. The large panel (Panel1) is the playing field. Three button controls are used to start / stop a program, set options, and exit the program. One timer control is used to control game animation and another is used to represent the computer's decision process. The second control panel (Panel2) is used to select game options. One group box contains radio buttons which are used to select number of players. A group box contains radio buttons to select the level of difficulty of the game, when playing against a computer. A small button is used to close the options panel. The default properties are set for one-player games with the easiest game difficulty. In chapter three, you will build Roasted Duck Delivery simulation. In this simulation, a number of decisions are needed. The basic idea is to read the order by incoming telephone and tell the delivery scooter to go to the location of the order. You also need to make sure that you always provide a roasted duck ready to be transported by the delivery scooter. The delivery area is a 20 by 20 square grid. The more roasted duck is sold, the more profit it gets. The panel control on the left side of the form contains the delivery grid. On the upper right are group boxes with two label controls to display the time or hour and sale results. The computer monitor (in a picture box) displays order and delivery status using a list box and label control. Another group box contains a roasting oven when the roasted ducks are

displayed using eight picture box controls. Two button controls on the group box control the operation of the oven. Group boxes under the oven show how many ducks are ready to be delivered and how many are in the delivery scooter (a button control is to load the roasted duck into the scooter). The two button controls beneath are used to start/pause the game and to stop the game or exit the game. In the area under the form there are several timers for controlling a number of aspects in the program. The delivery grid consists of 400 label controls on 20 rows (marked with numbers) and 20 columns (marked with letters). Here, you will learn how to place controls on a form (or panel in this case) using code (when the program runs, not when designing the form). This mechanism can save time designing the form. In chapter four, you will build a Drone Simulation. In this simulation, you control both vertical and horizontal thrusters to maneuver the ride to the landing pad. You will adjust the landing speed so that it is slow enough so that no accident occurs. You build the form in two stages, the first stage creates two option group boxes, and then the second stage uses both those group boxes as landing controls. Two control panels are placed on the left side of the form: one panel for drawing and another panel for the edge. On the right side of the form, place the two group control boxes. In the first group box, five radio buttons and a check box are added. In the second group box, two radio buttons are placed. In the below section of the form, three buttons are added. Finally, one timer control is added. Then in the form, a group box is added overlap panel. Then, 11 label controls are added to the group box. After that, a progress bar is added. Under the bar, two control panels are added, one high panel and one short panel. In the second (short) panel control, two small label controls are added. Underneath, three button controls are placed. Under these three buttons, a label control is added. For each label control, set the AutoSize property to False to be resized and set (temporarily) the BorderStyle property to FixedSingle so that you can see the edges to facilitate the layout process. In this chapter, you will build Jumper game. In this game, you will move the jumper across the busy road, avoid the tiger, and cross the river with the changing

current to get to house safely. You will place four label controls on the top part of the form (set the `AutoSize` property to `False` so that it can be resized and the `BorderStyle` property temporarily becomes `FixedSingle` so you can see the edges). Then, you use five panel controls below the labels. These panels will be a place for image graphics. Each panel has a width of 16 jumpers or 640 pixels, because one jumper will be given a width of 40 pixels. The first panel will be the jumper house, which will be given a height of 80 pixels. The next panel will become a river, with a height of 120 pixels. The next panel will be a place for tiger, 40 pixels high. Under the snake panel, there is a road panel. This panel will contain three boat lanes. Each boat has a height of 40 pixels, but you will give it a height of 140 pixels (not 120 pixels) to make room for lane markers. The fifth panel is the place where the jumper will begin its journey or leap. This panel will be given a height of 40 pixels. Add the last control panel below the form with three button controls. Then, finally, add four timer controls. Adjust the size of the form so that the panel controls can occupy according to the width of the form.

WIND POWER ANALYSIS AND FORECASTING USING MACHINE LEARNING WITH PYTHON - Vivian Siahaan 2022-05-18

Renewable energy remains one of the most important topics for a sustainable future. Wind, being a perennial source of power, could be utilized to satisfy our power requirements. With the rise of wind farms, wind power forecasting would prove to be quite useful. It contains various weather, turbine and rotor features. Data has been recorded from January 2018 till March 2020. Readings have been recorded at a 10-minute interval. The aim is to predict the wind power that could be generated from the windmill for the next 15 days. A longterm wind forecasting technique is thus required. The attributes in the dataset are as follows:

ActivePower, AmbientTemperature, BearingShaftTemperature, Blade1PitchAngle, Blade2PitchAngle, Blade3PitchAngle, ControlBoxTemperature, GearboxBearingTemperature, GearboxOilTemperature, GeneratorRP, GeneratorWinding1Temperature, GeneratorWinding2Temperature,

HubTemperature, MainBoxTemperature, NacellePosition, ReactivePower, RotorRPM, TurbineStatus, WTG, WindDirection, and WindSpeed. To perform forecasting based on regression active power, you will use: Linear Regression, Random Forest regression, Decision Tree regression, Support Vector Machine regression, Naïve Bayes regression, K-Nearest Neighbor regression, Adaboost regression, Gradient Boosting regression, Extreme Gradient Boosting regression, Light Gradient Boosting regression, Catboost regression, and MLP regression. To perform clustering, you will use K-Means algorithm. The machine learning models used predict categorized active power as target variable are K-Nearest Neighbor classifier, Random Forest classifier, Naive Bayes classifier, Logistic Regression classifier, Decision Tree classifier, Support Vector Machine classifier, LGBM classifier, Gradient Boosting classifier, XGB classifier, and MLP classifier. Finally, you will plot boundary decision, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy. [Access Database Crash Course - Vivian Siahaan 2019-10-12](#)

This book is a access database crash course which overs microsoft acces-based GUI programming using Python. In chapter one, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the

initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have LONGBINARY data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

A PRACTICAL GUIDE TO Database Programming with PHP/MySQL - Vivian Siahaan
2019-04-14

You will learn PHP/MySQL fast, easy and fun. This book provides you with a complete MySQL guidance presented in an easy-to-follow manner. Each chapter has practical examples with SQL script and screenshots available. If you go through the entire chapters, you will know how to manage MySQL databases and manipulate data using various techniques such as MySQL queries, MySQL stored procedures, database views, triggers. In the first part of the book, you will learn basic MySQL statements including how to implement querying data, sorting data, filtering data, joining tables, grouping data, subquerying data, dan setting operators. Aside from learning basic SQL statements, you will also learn step by step how to develop stored procedures in MySQL. First, we introduce you to the stored procedure concept and discuss when you should use it. Then, we show you how to use the basic elements of the procedure code such as create procedure statement, if-else, case, loop, stored procedure's parameters. In the next chapter, we will discuss the database views, how they are implemented in MySQL, and how to use them more effectively. After that, you will learn how to work with the MySQL triggers. By definition, a trigger or database trigger is a stored program executed automatically to respond to a specific event e.g., insert, update or delete occurred in a table. The database trigger is powerful tool for protecting the integrity of the data in your MySQL databases. In addition, it is useful to automate some database operations such as logging, auditing, etc. Then, you will learn about MySQL index including creating indexes, removing indexes, listing all indexes of a table and other important features of indexes in MySQL. MySQL uses indexes to quickly find rows with specific column values. Without an index, MySQL must scan the whole table to locate the relevant rows. The larger table, the slower it searches. After that, you will find a lot of useful MySQL administration techniques including MySQL server startup and shutdown, MySQL server security, MySQL database maintenance, and backup. The last chapter gives you the most commonly used MySQL functions including aggregate functions,

string functions, date time functions, control flow functions, etc.

Database and Image Processing Using SQL Server and Python - Vivian Siahaan 2019-10-31

The book details how programmers and database professionals can develop SQL Server-based Python applications that involves database and image processing. This book is SQL Server-based python programming. Microsoft SQL Server is robust relational database management system used by so many organizations of various sizes including top fortune 100 companies. SQL Server is a relational database management system (RDBMS) developed and marketed by Microsoft. As a database server, the primary function of the SQL Server is to store and retrieve data used by other applications. Deliberately designed for various levels of programming skill, this book is suitable for students, engineers, and even researchers in various disciplines. There is no need for advanced programming experience, and school-level programming skills are needed. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school

database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have VARBINARY(MAX) data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

GOLD PRICE ANALYSIS AND FORECASTING USING MACHINE LEARNING WITH PYTHON - Vivian Siahaan 2022-05-23

The challenge of this project is to accurately predict the future adjusted closing price of Gold ETF across a given period of time in the future. The problem is a regression problem, because the output value which is the adjusted closing

price in this project is continuous value. Data for this study is collected from November 18th 2011 to January 1st 2019 from various sources. The data has 1718 rows in total and 80 columns in total. Data for attributes, such as Oil Price, Standard and Poor's (S&P) 500 index, Dow Jones Index US Bond rates (10 years), Euro USD exchange rates, prices of precious metals Silver and Platinum and other metals such as Palladium and Rhodium, prices of US Dollar Index, Eldorado Gold Corporation and Gold Miners ETF were gathered. The dataset has 1718 rows in total and 80 columns in total. Data for attributes, such as Oil Price, Standard and Poor's (S&P) 500 index, Dow Jones Index US Bond rates (10 years), Euro USD exchange rates, prices of precious metals Silver and Platinum and other metals such as Palladium and Rhodium, prices of US Dollar Index, Eldorado Gold Corporation and Gold Miners ETF were gathered. To perform forecasting based on regression adjusted closing price of gold, you will use: Linear Regression, Random Forest regression, Decision Tree regression, Support Vector Machine regression, Naïve Bayes regression, K-Nearest Neighbor regression, Adaboost regression, Gradient Boosting regression, Extreme Gradient Boosting regression, Light Gradient Boosting regression, Catboost regression, and MLP regression. The machine learning models used predict gold daily returns as target variable are K-Nearest Neighbor classifier, Random Forest classifier, Naive Bayes classifier, Logistic Regression classifier, Decision Tree classifier, Support Vector Machine classifier, LGBM classifier, Gradient Boosting classifier, XGB classifier, MLP classifier, and Extra Trees classifier. Finally, you will plot boundary decision, distribution of features, feature importance, predicted values versus true values, confusion matrix, learning curve, performance of the model, and scalability of the model.

Data Science and Deep Learning Workshop For Scientists and Engineers - Vivian Siahaan 2021-11-04

WORKSHOP 1: In this workshop, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to implement deep learning on recognizing traffic signs using GTSRB dataset, detecting brain

tumor using Brain Image MRI dataset, classifying gender, and recognizing facial expression using FER2013 dataset In Chapter 1, you will learn to create GUI applications to display line graph using PyQt. You will also learn how to display image and its histogram. In Chapter 2, you will learn how to use TensorFlow, Keras, Scikit-Learn, Pandas, NumPy and other libraries to perform prediction on handwritten digits using MNIST dataset with PyQt. You will build a GUI application for this purpose. In Chapter 3, you will learn how to perform recognizing traffic signs using GTSRB dataset from Kaggle. There are several different types of traffic signs like speed limits, no entry, traffic signals, turn left or right, children crossing, no passing of heavy vehicles, etc. Traffic signs classification is the process of identifying which class a traffic sign belongs to. In this Python project, you will build a deep neural network model that can classify traffic signs in image into different categories. With this model, you will be able to read and understand traffic signs which are a very important task for all autonomous vehicles. You will build a GUI application for this purpose. In Chapter 4, you will learn how to perform detecting brain tumor using Brain Image MRI dataset provided by Kaggle (<https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>) using CNN model. You will build a GUI application for this purpose. In Chapter 5, you will learn how to perform classifying gender using dataset provided by Kaggle (<https://www.kaggle.com/cashutosh/gender-classification-dataset>) using MobileNetV2 and CNN models. You will build a GUI application for this purpose. In Chapter 6, you will learn how to perform recognizing facial expression using FER2013 dataset provided by Kaggle (<https://www.kaggle.com/nicolejyt/facialexpressionrecognition>) using CNN model. You will also build a GUI application for this purpose. WORKSHOP 2: In this workshop, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to implement deep learning on classifying fruits, classifying cats/dogs, detecting furnitures, and classifying fashion. In Chapter 1, you will learn to create GUI applications to display line graph using PyQt. You will also learn how to display

image and its histogram. Then, you will learn how to use OpenCV, NumPy, and other libraries to perform feature extraction with Python GUI (PyQt). The feature detection techniques used in this chapter are Harris Corner Detection, Shi-Tomasi Corner Detector, and Scale-Invariant Feature Transform (SIFT). In Chapter 2, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform classifying fruits using Fruits 360 dataset provided by Kaggle (<https://www.kaggle.com/moltean/fruits/code>) using Transfer Learning and CNN models. You will build a GUI application for this purpose. In Chapter 3, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform classifying cats/dogs using dataset provided by Kaggle (<https://www.kaggle.com/chetankv/dogs-cats-images>) using Using CNN with Data Generator. You will build a GUI application for this purpose. In Chapter 4, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform detecting furnitures using Furniture Detector dataset provided by Kaggle (<https://www.kaggle.com/akkithetechie/furniture-detector>) using VGG16 model. You will build a GUI application for this purpose. In Chapter 5, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform classifying fashion using Fashion MNIST dataset provided by Kaggle (<https://www.kaggle.com/zalando-research/fashionmnist/code>) using CNN model. You will build a GUI application for this purpose. WORKSHOP 3: In this workshop, you will implement deep learning on detecting vehicle license plates, recognizing sign language, and detecting surface crack using TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries. In Chapter 1, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform detecting vehicle license plates using Car License Plate Detection dataset provided by Kaggle (<https://www.kaggle.com/andrewmvd/car-plate-detection/download>). In Chapter 2, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to

perform sign language recognition using Sign Language Digits Dataset provided by Kaggle (<https://www.kaggle.com/ardamavi/sign-language-digits-dataset/download>). In Chapter 3, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform detecting surface crack using Surface Crack Detection provided by Kaggle (<https://www.kaggle.com/arunrk7/surface-crack-detection/download>). WORKSHOP 4: In this workshop, implement deep learning-based image classification on detecting face mask, classifying weather, and recognizing flower using TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries. In Chapter 1, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform detecting face mask using Face Mask Detection Dataset provided by Kaggle (<https://www.kaggle.com/omkargurav/face-mask-dataset/download>). In Chapter 2, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform how to classify weather using Multi-class Weather Dataset provided by Kaggle (<https://www.kaggle.com/pratik2901/multiclass-weather-dataset/download>). WORKSHOP 5: In this workshop, implement deep learning-based image classification on classifying monkey species, recognizing rock, paper, and scissor, and classify airplane, car, and ship using TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries. In Chapter 1, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform how to classify monkey species using 10 Monkey Species dataset provided by Kaggle (<https://www.kaggle.com/slothkong/10-monkey-species/download>). In Chapter 2, you will learn how to use TensorFlow, Keras, Scikit-Learn, OpenCV, Pandas, NumPy and other libraries to perform how to recognize rock, paper, and scissor using 10 Monkey Species dataset provided by Kaggle (<https://www.kaggle.com/sanikamal/rock-paper-scissors-dataset/download>). WORKSHOP 6: In this worksshop, you will implement two data science projects using Scikit-Learn, Scipy, and

other libraries with Python GUI. In Chapter 1, you will learn how to use Scikit-Learn, Scipy, and other libraries to perform how to predict traffic (number of vehicles) in four different junctions using Traffic Prediction Dataset provided by Kaggle (<https://www.kaggle.com/fedesoriano/traffic-prediction-dataset/download>). This dataset contains 48.1k (48120) observations of the number of vehicles each hour in four different junctions: 1) DateTime; 2) Junction; 3) Vehicles; and 4) ID. In Chapter 2, you will learn how to use Scikit-Learn, NumPy, Pandas, and other libraries to perform how to analyze and predict heart attack using Heart Attack Analysis & Prediction Dataset provided by Kaggle (<https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset/download>).

WORKSHOP 7: In this workshop, you will implement two data science projects using Scikit-Learn, Scipy, and other libraries with Python GUI. In Project 1, you will learn how to use Scikit-Learn, NumPy, Pandas, Seaborn, and other libraries to perform how to predict early stage diabetes using Early Stage Diabetes Risk Prediction Dataset provided by Kaggle (<https://www.kaggle.com/ishandutta/early-stage-diabetes-risk-prediction-dataset/download>). This dataset contains the sign and symptom data of newly diabetic or would be diabetic patient. This has been collected using direct questionnaires from the patients of Sylhet Diabetes Hospital in Sylhet, Bangladesh and approved by a doctor. You will develop a GUI using PyQt5 to plot distribution of features, feature importance, cross validation score, and predicted values versus true values. The machine learning models used in this project are Adaboost, Random Forest, Gradient Boosting, Logistic Regression, and Support Vector Machine. In Project 2, you will learn how to use Scikit-Learn, NumPy, Pandas, and other libraries to perform how to analyze and predict breast cancer using Breast Cancer Prediction Dataset provided by Kaggle (<https://www.kaggle.com/merishnasuwal/breast-cancer-prediction-dataset/download>). Worldwide, breast cancer is the most common type of cancer in women and the second highest in terms of mortality rates. Diagnosis of breast cancer is performed when an abnormal lump is

found (from self-examination or x-ray) or a tiny speck of calcium is seen (on an x-ray). After a suspicious lump is found, the doctor will conduct a diagnosis to determine whether it is cancerous and, if so, whether it has spread to other parts of the body. This breast cancer dataset was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. You will develop a GUI using PyQt5 to plot distribution of features, pairwise relationship, test scores, predicted values versus true values, confusion matrix, and decision boundary. The machine learning models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, and Support Vector Machine.

WORKSHOP 8: In this workshop, you will learn how to use Scikit-Learn, TensorFlow, Keras, NumPy, Pandas, Seaborn, and other libraries to implement brain tumor classification and detection with machine learning using Brain Tumor dataset provided by Kaggle. This dataset contains five first order features: Mean (the contribution of individual pixel intensity for the entire image), Variance (used to find how each pixel varies from the neighboring pixel 0, Standard Deviation (the deviation of measured Values or the data from its mean), Skewness (measures of symmetry), and Kurtosis (describes the peak of e.g. a frequency distribution). It also contains eight second order features: Contrast, Energy, ASM (Angular second moment), Entropy, Homogeneity, Dissimilarity, Correlation, and Coarseness. The machine learning models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, and Support Vector Machine. The deep learning models used in this project are MobileNet and ResNet50. In this project, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, training loss, and training accuracy.

WORKSHOP 9: In this workshop, you will learn how to use Scikit-Learn, Keras, TensorFlow, NumPy, Pandas, Seaborn, and other libraries to perform COVID-19 Epitope Prediction using COVID-19/SARS B-cell Epitope Prediction dataset provided in Kaggle. All of three datasets consists of information of protein and peptide:

parent_protein_id : parent protein ID;
protein_seq : parent protein sequence;
start_position : start position of peptide;
end_position : end position of peptide;
peptide_seq : peptide sequence; chou_fasman :
peptide feature; emini : peptide feature, relative
surface accessibility; kolaskar_tongaonkar :
peptide feature, antigenicity; parker : peptide
feature, hydrophobicity; isoelectric_point :
protein feature; aromacity: protein feature;
hydrophobicity : protein feature; stability :
protein feature; and target : antibody valence
(target value). The machine learning models
used in this project are K-Nearest Neighbor,
Random Forest, Naive Bayes, Logistic
Regression, Decision Tree, Support Vector
Machine, Adaboost, Gradient Boosting, XGB
classifier, and MLP classifier. Then, you will
learn how to use sequential CNN and VGG16
models to detect and predict Covid-19 X-RAY
using COVID-19 Xray Dataset (Train & Test
Sets) provided in Kaggle. The folder itself
consists of two subfolders: test and train.
Finally, you will develop a GUI using PyQt5 to
plot boundary decision, ROC, distribution of
features, feature importance, cross validation
score, and predicted values versus true values,
confusion matrix, training loss, and training
accuracy. WORKSHOP 10: In this workshop, you
will learn how to use Scikit-Learn, Keras,
TensorFlow, NumPy, Pandas, Seaborn, and other
libraries to perform analyzing and predicting
stroke using dataset provided in Kaggle. The
dataset consists of attribute information: id:
unique identifier; gender: "Male", "Female" or
"Other"; age: age of the patient; hypertension: 0
if the patient doesn't have hypertension, 1 if
the patient has hypertension; heart_disease: 0
if the patient doesn't have any heart diseases,
1 if the patient has a heart disease; ever_married:
"No" or "Yes"; work_type: "children", "Govt_jov",
"Never_worked", "Private" or "Self-employed";
Residence_type: "Rural" or "Urban";
avg_glucose_level: average glucose level in
blood; bmi: body mass index; smoking_status:
"formerly smoked", "never smoked", "smokes" or
"Unknown"; and stroke: 1 if the patient had a
stroke or 0 if not. The models used in this
project are K-Nearest Neighbor, Random Forest,
Naive Bayes, Logistic Regression, Decision Tree,
Support Vector Machine, Adaboost, LGBM

classifier, Gradient Boosting, XGB classifier,
MLP classifier, and CNN 1D. Finally, you will
develop a GUI using PyQt5 to plot boundary
decision, ROC, distribution of features, feature
importance, cross validation score, and
predicted values versus true values, confusion
matrix, learning curve, performace of the model,
scalability of the model, training loss, and
training accuracy. WORKSHOP 11: In this
workshop, you will learn how to use Scikit-
Learn, Keras, TensorFlow, NumPy, Pandas,
Seaborn, and other libraries to perform
classifying and predicting Hepatitis C using
dataset provided by UCI Machine Learning
Repository. All attributes in dataset except
Category and Sex are numerical. Attributes 1 to
4 refer to the data of the patient: X (Patient
ID/No.), Category (diagnosis) (values: '0=Blood
Donor', '0s=suspect Blood Donor', '1=Hepatitis',
'2=Fibrosis', '3=Cirrhosis'), Age (in years), Sex
(f,m), ALB, ALP, ALT, AST, BIL, CHE, CHOL,
CREA, GGT, and PROT. The target attribute for
classification is Category (2): blood donors vs.
Hepatitis C patients (including its progress ('just'
Hepatitis C, Fibrosis, Cirrhosis). The models
used in this project are K-Nearest Neighbor,
Random Forest, Naive Bayes, Logistic
Regression, Decision Tree, Support Vector
Machine, Adaboost, LGBM classifier, Gradient
Boosting, XGB classifier, MLP classifier, and
ANN 1D. Finally, you will develop a GUI using
PyQt5 to plot boundary decision, ROC,
distribution of features, feature importance,
cross validation score, and predicted values
versus true values, confusion matrix, learning
curve, performace of the model, scalability of the
model, training loss, and training accuracy.
**PYTHON GUI PROJECTS WITH MACHINE
LEARNING AND DEEP LEARNING** - Vivian
Siahaan 2022-01-16
PROJECT 1: THE APPLIED DATA SCIENCE
WORKSHOP: Prostate Cancer Classification and
Recognition Using Machine Learning and Deep
Learning with Python GUI Prostate cancer is
cancer that occurs in the prostate. The prostate
is a small walnut-shaped gland in males that
produces the seminal fluid that nourishes and
transports sperm. Prostate cancer is one of the
most common types of cancer. Many prostate
cancers grow slowly and are confined to the
prostate gland, where they may not cause

serious harm. However, while some types of prostate cancer grow slowly and may need minimal or even no treatment, other types are aggressive and can spread quickly. The dataset used in this project consists of 100 patients which can be used to implement the machine learning and deep learning algorithms. The dataset consists of 100 observations and 10 variables (out of which 8 numeric variables and one categorical variable and is ID) which are as follows: Id, Radius, Texture, Perimeter, Area, Smoothness, Compactness, Diagnosis Result, Symmetry, and Fractal Dimension. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

PROJECT 2: THE APPLIED DATA SCIENCE WORKSHOP: Urinary Biomarkers Based Pancreatic Cancer Classification and Prediction Using Machine Learning with Python GUI

Pancreatic cancer is an extremely deadly type of cancer. Once diagnosed, the five-year survival rate is less than 10%. However, if pancreatic cancer is caught early, the odds of surviving are much better. Unfortunately, many cases of pancreatic cancer show no symptoms until the cancer has spread throughout the body. A diagnostic test to identify people with pancreatic cancer could be enormously helpful. In a paper by Silvana Debernardi and colleagues, published this year in the journal PLOS Medicine, a multinational team of researchers sought to develop an accurate diagnostic test for the most common type of pancreatic cancer, called pancreatic ductal adenocarcinoma or PDAC. They gathered a series of biomarkers from the urine of three groups of patients: Healthy controls, Patients with non-cancerous pancreatic conditions, like chronic pancreatitis, and Patients with pancreatic ductal adenocarcinoma. When possible, these patients were age- and sex-matched. The goal was to develop an accurate

way to identify patients with pancreatic cancer. The key features are four urinary biomarkers: creatinine, LYVE1, REG1B, and TFF1. Creatinine is a protein that is often used as an indicator of kidney function. YVLE1 is lymphatic vessel endothelial hyaluronan receptor 1, a protein that may play a role in tumor metastasis. REG1B is a protein that may be associated with pancreas regeneration. TFF1 is trefoil factor 1, which may be related to regeneration and repair of the urinary tract. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, and MLP classifier. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

PROJECT 3: DATA SCIENCE CRASH COURSE: Voice Based Gender Classification and Prediction Using Machine Learning and Deep Learning with Python GUI

This dataset was created to identify a voice as male or female, based upon acoustic properties of the voice and speech. The dataset consists of 3,168 recorded voice samples, collected from male and female speakers. The voice samples are pre-processed by acoustic analysis in R using the seewave and tuneR packages, with an analyzed frequency range of 0hz-280hz (human vocal range). The following acoustic properties of each voice are measured and included within the CSV: meanfreq: mean frequency (in kHz); sd: standard deviation of frequency; median: median frequency (in kHz); Q25: first quantile (in kHz); Q75: third quantile (in kHz); IQR: interquantile range (in kHz); skew: skewness; kurt: kurtosis; sp.ent: spectral entropy; sfm: spectral flatness; mode: mode frequency; centroid: frequency centroid (see specprop); peakf: peak frequency (frequency with highest energy); meanfun: average of fundamental frequency measured across acoustic signal; minfun: minimum fundamental frequency measured across acoustic signal; maxfun: maximum fundamental frequency measured across acoustic signal; meandom: average of dominant frequency

measured across acoustic signal; mindom: minimum of dominant frequency measured across acoustic signal; maxdom: maximum of dominant frequency measured across acoustic signal; dfrange: range of dominant frequency measured across acoustic signal; modindx: modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range; and label: male or female. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

PROJECT 4: DATA SCIENCE CRASH COURSE: Thyroid Disease Classification and Prediction Using Machine Learning and Deep Learning with Python GUI

Thyroid disease is a general term for a medical condition that keeps your thyroid from making the right amount of hormones. Thyroid typically makes hormones that keep body functioning normally. When the thyroid makes too much thyroid hormone, body uses energy too quickly. The two main types of thyroid disease are hypothyroidism and hyperthyroidism. Both conditions can be caused by other diseases that impact the way the thyroid gland works. Dataset used in this project was from Garavan Institute Documentation as given by Ross Quinlan 6 databases from the Garavan Institute in Sydney, Australia. Approximately the following for each database: 2800 training (data) instances and 972 test instances. This dataset contains plenty of missing data, while 29 or so attributes, either Boolean or continuously-valued. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC,

distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

STUDENT ACADEMIC PERFORMANCE ANALYSIS AND PREDICTION USING MACHINE LEARNING WITH PYTHON -

Vivian Siahaan 2022-03-20

The dataset used in this project consists of student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school-related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). In the two datasets were modeled under binary/five-level classification and regression tasks. Important note: the target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. It is more difficult to predict G3 without G2 and G1, but such prediction is much more useful. Attributes in the dataset are as follows: school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira); sex - student's sex (binary: 'F' - female or 'M' - male); age - student's age (numeric: from 15 to 22); address - student's home address type (binary: 'U' - urban or 'R' - rural); famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3); Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart); Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education); Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education); Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other'); Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other'); reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course'

preference or 'other'); guardian - student's guardian (nominal: 'mother', 'father' or 'other'); traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour); studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours); failures - number of past class failures (numeric: n if $1 \leq n < 3$, else 4); schoolsup - extra educational support (binary: yes or no); famsup - family educational support (binary: yes or no); paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no); activities - extra-curricular activities (binary: yes or no); nursery - attended nursery school (binary: yes or no); higher - wants to take higher education (binary: yes or no); internet - Internet access at home (binary: yes or no); romantic - with a romantic relationship (binary: yes or no); famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent); freetime - free time after school (numeric: from 1 - very low to 5 - very high); goout - going out with friends (numeric: from 1 - very low to 5 - very high); Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high); Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high); health - current health status (numeric: from 1 - very bad to 5 - very good); absences - number of school absences (numeric: from 0 to 93); G1 - first period grade (numeric: from 0 to 20); G2 - second period grade (numeric: from 0 to 20); and G3 - final grade (numeric: from 0 to 20, output target). The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, and XGB classifier. Three feature scaling used in machine learning are raw, minmax scaler, and standard scaler. Finally, you will develop a GUI using PyQt5 to plot cross validation score, predicted values versus true values, confusion matrix, learning curve, decision boundaries, performance of the model, scalability of the model, training loss, and training accuracy.

DATA SCIENCE WORKSHOP: Alzheimer's Disease Classification and Prediction Using Machine Learning and Deep Learning with Python GUI - Vivian Siahaan 2021-11-20

Alzheimer's is a type of dementia that causes problems with memory, thinking and behavior. Symptoms usually develop slowly and get worse over time, becoming severe enough to interfere with daily tasks. Alzheimer's is not a normal part of aging. The greatest known risk factor is increasing age, and the majority of people with Alzheimer's are 65 and older. But Alzheimer's is not just a disease of old age. Approximately 200,000 Americans under the age of 65 have younger-onset Alzheimer's disease (also known as early-onset Alzheimer's). The dataset consists of a longitudinal MRI data of 374 subjects aged 60 to 96. Each subject was scanned at least once. Everyone is right-handed. 206 of the subjects were grouped as 'Nondemented' throughout the study. 107 of the subjects were grouped as 'Demented' at the time of their initial visits and remained so throughout the study. 14 subjects were grouped as 'Nondemented' at the time of their initial visit and were subsequently characterized as 'Demented' at a later visit. These fall under the 'Converted' category. Following are some important features in the dataset: EDUC:Years of Education; SES: Socioeconomic Status; MMSE: Mini Mental State Examination; CDR: Clinical Dementia Rating; eTIV: Estimated Total Intracranial Volume; nWBV: Normalize Whole Brain Volume; and ASF: Atlas Scaling Factor. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

CRYPTOGRAPHY AND IMAGE PROCESSING with Java GUI and SQLite - Vivian Siahaan 2019-10-31

The book details how programmers and database professionals can develop SQLite-based Java GUI applications that involves cryptography and image processing. In this book, you will learn how to build from scratch a criminal records management database system

using Java/SQLite. All Java code for digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. In chapter one, you will create Bank database and its four tables. In chapter two, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. In chapter three, you will learn how to create and store salt passwords and verify them. You will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter four, you will create an Account table. This account table has the following ten fields: account_id (primary key), client_id (primary key), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter five, you will create a Client_Data table, which has the following seven fields: client_data_id (primary key), account_id (primary key), birth_date, address, mother_name, telephone, and photo_path. In chapter six, you will create Crime database and its six tables. In chapter seven, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key),

suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter ten, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

Access Database for Pragmatic Programmers: A Step by Step Guide to Create Database-Driven Application Using Python - Vivian Siahaan 2019-10-11

This book covers microsoft acces-based GUI programming using pyqt. Intentionally designed for various levels of interest and ability of learners, this book is suitable for students, engineers, and even researchers in a variety of disciplines. No advanced programming experience is needed, and only a few school-level programming skill are needed. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the

Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In third chapter, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In fourth chapter, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In the last chapter, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables.

DATA SCIENCE WORKSHOP: Chronic Kidney Disease Classification and Prediction Using Machine Learning and Deep Learning with Python GUI - Vivian Siahaan 2021-11-12

Chronic kidney disease is the longstanding disease of the kidneys leading to renal failure. The kidneys filter waste and excess fluid from the blood. As kidneys fail, waste builds up. Symptoms develop slowly and aren't specific to the disease. Some people have no symptoms at all and are diagnosed by a lab test. Medication helps manage symptoms. In later stages, filtering the blood with a machine (dialysis) or a transplant may be required. The dataset used in this project was taken over a 2-month period in India with 25 features (eg, red blood cell count, white blood cell count, etc). The target is the 'classification', which is either 'ckd' or 'notckd' - ckd=chronic kidney disease. It contains measures of 24 features for 400 people. Quite a lot of features for just 400 samples. There are 14 categorical features, while 10 are numerical. The dataset needs cleaning: in that it has NaNs and the numeric features need to be forced to floats. Attribute Information: Age(numerical) age

in years; Blood Pressure(numerical) bp in mm/Hg; Specific Gravity(categorical) sg - (1.005,1.010,1.015,1.020,1.025); Albumin(categorical) al - (0,1,2,3,4,5); Sugar(categorical) su - (0,1,2,3,4,5); Red Blood Cells(categorical) rbc - (normal,abnormal); Pus Cell (categorical) pc - (normal,abnormal); Pus Cell clumps(categorical) pcc - (present, notpresent); Bacteria(categorical) ba - (present,notpresent); Blood Glucose Random(numerical) bgr in mgs/dl; Blood Urea(numerical) bu in mgs/dl; Serum Creatinine(numerical) sc in mgs/dl; Sodium(numerical) sod in mEq/L; Potassium(numerical) pot in mEq/L; Hemoglobin(numerical) hemo in gms; Packed Cell Volume(numerical); White Blood Cell Count(numerical) wc in cells/cumm; Red Blood Cell Count(numerical) rc in millions/cmm; Hypertension(categorical) htn - (yes,no); Diabetes Mellitus(categorical) dm - (yes,no); Coronary Artery Disease(categorical) cad - (yes,no); Appetite(categorical) appet - (good,poor); Pedal Edema(categorical) pe - (yes,no); Anemia(categorical) ane - (yes,no); and Class (categorical) class - (ckd,notckd). The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

SQLITE AND DATA SCIENCE: QUERIES AND VISUALIZATION WITH PYTHON GUI - Vivian Siahaan 2022-06-15

In this project, you will develop GUI with PyQt5 to: utilize Push Button, Combo Box, Table Widget, Line Edit, and Widget, read and create SQLite database and every table in it, plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of

Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue costumers; plot which customer have least and most overdue days; plot which store have most sales; plot average payment amount by month with mean and EWM; and plot payment amount over June 2005. This project uses the Sakila sample database which is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, film_actor, customer, rental, payment and inventory among others. You can download the SQLite from <https://dev.mysql.com/doc/sakila/en/>.

Classification and Prediction Projects with Machine Learning and Deep Learning - Vivian Siahaan 2022-02-06

PROJECT 1: DATA SCIENCE CRASH COURSE: Drinking Water Potability Classification and Prediction Using Machine Learning and Deep Learning with Python Access to safe drinking water is essential to health, a basic human right, and a component of effective policy for health protection. This is important as a health and development issue at a national, regional, and local level. In some regions, it has been shown that investments in water supply and sanitation can yield a net economic benefit, since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the interventions. The drinkingwaterpotability.csv file contains water quality metrics for 3276 different water bodies. The columns in the file are as follows: ph, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, Turbidity, and Potability. Contaminated water and poor sanitation are linked to the transmission of diseases such as cholera, diarrhea, dysentery, hepatitis A, typhoid, and polio. Absent, inadequate, or inappropriately managed water and sanitation services expose individuals to preventable health risks. This is particularly the

case in health care facilities where both patients and staff are placed at additional risk of infection and disease when water, sanitation, and hygiene services are lacking. The machine learning models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy. PROJECT 2: DATA SCIENCE CRASH COURSE: Skin Cancer Classification and Prediction Using Machine Learning and Deep Learning Skin cancer develops primarily on areas of sun-exposed skin, including the scalp, face, lips, ears, neck, chest, arms and hands, and on the legs in women. But it can also form on areas that rarely see the light of day — your palms, beneath your fingernails or toenails, and your genital area. Skin cancer affects people of all skin tones, including those with darker complexions. When melanoma occurs in people with dark skin tones, it's more likely to occur in areas not normally exposed to the sun, such as the palms of the hands and soles of the feet. Dataset used in this project contains a balanced dataset of images of benign skin moles and malignant skin moles. The data consists of two folders with each 1800 pictures (224x244) of the two types of moles. The machine learning models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. The deep learning models used are CNN and MobileNet.

Database and Image Processing Using Python and Access - Vivian Siahaan 2019-10-31

The book details how programmers and database professionals can develop access-based Python applications that involves database and image processing. This book is a access database crash course which overs microsoft acces-based GUI programming using Python. In chapter one, you will learn to use several

widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have LONGBINARY data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The

Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

Database and Image Processing Using Java GUI and Microsoft Access - Vivian Siahaan
2019-11-01

The book details how programmers and database professionals can develop Access-based Java GUI applications that involves database and image processing. This book will help you quickly write efficient, high-quality access-database-driven code with Java. It's an ideal way to begin, whether you're new to programming or a professional developer versed in other languages. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications).The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch two access database management systems using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will create School database and six tables. In chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join

and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create Crime database and its tables. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eighth, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter nine, you will add two tables: Police and Investigator. These two tables will later be joined to Suspect table through another table, Case_File, which will be built in the seventh chapter. The Police has six columns: police_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and Case_File. The Case_File table will connect four other tables: Suspect, Police, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

DATA SCIENCE WORKSHOP: Parkinson Classification and Prediction Using Machine Learning and Deep Learning with Python GUI - Vivian Siahaan 2021-11-22

The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD. The data is in ASCII CSV format. The rows of the CSV file contain an instance corresponding to one voice recording. There are around six recordings per patient, the name of the patient is identified in the first column. Attribute information of this dataset are as follows: name - ASCII subject name and recording number; MDVP:Fo(Hz) - Average vocal fundamental frequency; MDVP:Fhi(Hz) - Maximum vocal fundamental frequency; MDVP:Flo(Hz) - Minimum vocal fundamental frequency; MDVP:Jitter(%); MDVP:Jitter(Abs); MDVP:RAP; MDVP:PPQ; Jitter:DDP - Several measures of variation in fundamental frequency; MDVP:Shimmer; MDVP:Shimmer(dB); Shimmer:APQ3; Shimmer:APQ5; MDVP:APQ; Shimmer:DDA - Several measures of variation in amplitude; NHR; HNR - Two measures of ratio of noise to tonal components in the voice; status - Health status of the subject (one) - Parkinson's, (zero) - healthy; RPDE,D2 - Two nonlinear dynamical complexity measures; DFA - Signal fractal scaling exponent; and spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation. The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, XGB classifier, MLP classifier, and CNN 1D. Finally, you will develop a GUI using PyQt5 to plot boundary decision, ROC,

distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy. [LEARN FROM SCRATCH VISUAL C# .NET WITH SQL SERVER To Develop Database-Driven Desktop Applications](#) - VIVIAN SIAHAAN 2020-10-10

In Tutorial 1, you will start building a Visual C# interface for database management system project with SQL Server. The database, named DBMS, is created. The designed interface in this tutorial will be used as the main terminal in accessing other forms. This tutorial will also discuss how to create login form and login table. In Tutorial 2, you will build a project, as part of database management system, where you can store information about valuables in school. In Tutorial 3 up to Tutorial 4, you will perform the steps necessary to add 6 tables into DBMS database. You will build each table and add the associated fields as needed. In this tutorial, you will create a library database project, as part of database management system, where you can store all information about library including author, title, and publisher. In Tutorial 5 up to Tutorial 7, you will perform the steps necessary to add 6 more tables into DBMS database. You will build each table and add the associated fields as needed. In this tutorial, you will create a high school database project, as part of database management system, where you can store all information about school including parent, teacher, student, subject, and, title, and grade.

CUSTOMER SEGMENTATION, CLUSTERING, AND PREDICTION WITH PYTHON - Vivian Siahaan 2022-02-24

In this project, you will develop a customer segmentation, clustering, and prediction to define marketing strategy. The sample dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioral variables. Following is the Data Dictionary for Credit Card dataset: CUSTID: Identification of Credit Card holder (Categorical); BALANCE: Balance amount left in their account to make purchases; BALANCEFREQUENCY: How frequently the

Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated); PURCHASES: Amount of purchases made from account; ONEOFFPURCHASES: Maximum purchase amount done in one-go; INSTALLMENTSPURCHASES: Amount of purchase done in installment; CASHADVANCE: Cash in advance given by the user; PURCHASESFREQUENCY: How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased); ONEOFFPURCHASESFREQUENCY: How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased); PURCHASESINSTALLMENTSFREQUENCY: How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done); CASHADVANCEFREQUENCY: How frequently the cash in advance being paid; CASHADVANCETRX: Number of Transactions made with "Cash in Advanced"; PURCHASESTRX: Number of purchase transactions made; CREDITLIMIT: Limit of Credit Card for user; PAYMENTS: Amount of Payment done by user; MINIMUM_PAYMENTS: Minimum amount of payments made by user; PRCFULLPAYMENT: Percent of full payment paid by user; and TENURE: Tenure of credit card service for user. In this project, you will perform clustering using KMeans to get 5 clusters. The machine learning models used in this project to perform regression on total number of purchase and to predict clusters as target variable are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, LGBM, Gradient Boosting, XGB, and MLP. Finally, you will plot boundary decision, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, learning curve, performance of the model, scalability of the model, training loss, and training accuracy.

[VISUAL BASIC .NET AND DATABASE: PRACTICAL TUTORIALS](#) - Vivian Siahaan 2020-10-31

This book aims to develop a MySQL-driven desktop application that readers can develop for their own purposes to implement library project using Visual Basic .NET. In Tutorial 1, you will

build a Visual Basic interface for the database. This interface will be used as the main terminal in accessing other forms. This tutorial will also discuss how to create login form and login table. You will create login form. Place on the form one picture box, two labels, one combo box, one text box, and two buttons. In Tutorial 2, you will build a school inventory project where you can store information about valuables in school. The table will have nine fields: Item (description of the item), Quantity, Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). In Tutorial 3, you will perform the steps necessary to add 5 new tables using phpMyAdmin into Academy database. You will build each table and add the associated fields as needed. Every table in the database will need input form. In this tutorial, you will build such a form for Author table. Although this table is quite simple (only four fields: AuthorID, Name, BirthDate, and PhotoFile), it provides a basis for illustrating the many steps in interface design. SQL statement is required by the Command object to read fields (sorted by Name). Then, you will build an interface so that the user can maintain the Publisher table in the database (Academy). The Publisher table interface is more or less the same as Author table interface. This Publisher table interface only requires more input fields. So you will use the interface for the Author table and modify it for the Publisher table. In Tutorial 4, you will perform the steps necessary to design and implement title form, library member form, and book borrowing form. You start by designing and testing the basic entry form for book titles. The Title table has nine fields: BookTitle, PublishYear, ISBN, PublisherID, AuthorID, Description, Note, Subject, and Comment. Then, you will build such a form for Member table. This table has twelve fields: MemberID, FirstName, LastName, BirthDate, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, and PhotoFile). You need thirteen label controls, one picture box, six text boxes, four combo boxes, one check box, one date time picker, one openfiledialog, and one

printpreviewdialog. You also need four buttons for navigation, six buttons for controlling editing features, one button for searching member's name, and one button to upload member's photo. Finally, you will build such a form for Borrow table. This table has seven fields: BorrowID, MemberID, BorrowCode, ISBN, BorrowDate, ReturnDate, and Penalty. In this form, you need fourteen label controls, seven text boxes, two combo boxes, two date time pickers, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for other utilities, one button to generate borrowing code, and one button to return book.

SQLITE QUERIES, ANALYSIS, AND VISUALIZATION WITH PYTHON - Vivian Siahaan 2022-06-01

Sakila for SQLite is a part of the sakila-sample-database-ports project intended to provide ported versions of the original MySQL database for other database systems, including: Oracle, SQL Server, SQLite, Interbase/Firebird, and Microsoft Access. Sakila for SQLite is a port of the Sakila example database available for MySQL, which was originally developed by Mike Hillyer of the MySQL AB documentation team. The project is designed to help database administrators to decide which database to use for development of new products. In this project, you will: read sqlite database and every table in it; read every actor in actor table, read every film in films table; plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue costumers; plot which customer have least and most overdue days; plot which store have most sales; plot average payment amount

by month with mean and EWM; and plot payment amount over June 2005.

HIGHER EDUCATION STUDENT ACADEMIC PERFORMANCE ANALYSIS AND PREDICTION USING MACHINE LEARNING WITH PYTHON GUI - Vivian Siahaan

2022-04-24

The dataset used in this project was collected from the Faculty of Engineering and Faculty of Educational Sciences students in 2019. The purpose is to predict students' end-of-term performances using ML techniques. Attribute information in the dataset are as follows: Student ID; Student Age (1: 18-21, 2: 22-25, 3: above 26); Sex (1: female, 2: male); Graduated high-school type: (1: private, 2: state, 3: other); Scholarship type: (1: None, 2: 25%, 3: 50%, 4: 75%, 5: Full); Additional work: (1: Yes, 2: No); Regular artistic or sports activity: (1: Yes, 2: No); Do you have a partner: (1: Yes, 2: No); Total salary if available (1: USD 135-200, 2: USD 201-270, 3: USD 271-340, 4: USD 341-410, 5: above 410); Transportation to the university: (1: Bus, 2: Private car/taxi, 3: bicycle, 4: Other); Accommodation type in Cyprus: (1: rental, 2: dormitory, 3: with family, 4: Other); Mother's education: (1: primary school, 2: secondary school, 3: high school, 4: university, 5: MSc., 6: Ph.D.); Father's education: (1: primary school, 2: secondary school, 3: high school, 4: university, 5: MSc., 6: Ph.D.); Number of sisters/brothers (if available): (1: 1, 2: 2, 3: 3, 4: 4, 5: 5 or above); Parental status: (1: married, 2: divorced, 3: died - one of them or both); Mother's occupation: (1: retired, 2: housewife, 3: government officer, 4: private sector employee, 5: self-employment, 6: other); Father's occupation: (1: retired, 2: government officer, 3: private sector employee, 4: self-employment, 5: other); Weekly study

hours: (1: None, 2: <5 hours, 3: 6-10 hours, 4: 11-20 hours, 5: more than 20 hours); Reading frequency (non-scientific books/journals): (1: None, 2: Sometimes, 3: Often); Reading frequency (scientific books/journals): (1: None, 2: Sometimes, 3: Often); Attendance to the seminars/conferences related to the department: (1: Yes, 2: No); Impact of your projects/activities on your success: (1: positive, 2: negative, 3: neutral); Attendance to classes (1: always, 2: sometimes, 3: never); Preparation to midterm exams 1: (1: alone, 2: with friends, 3: not applicable); Preparation to midterm exams 2: (1: closest date to the exam, 2: regularly during the semester, 3: never); Taking notes in classes: (1: never, 2: sometimes, 3: always); Listening in classes: (1: never, 2: sometimes, 3: always); Discussion improves my interest and success in the course: (1: never, 2: sometimes, 3: always); Flip-classroom: (1: not useful, 2: useful, 3: not applicable); Cumulative grade point average in the last semester (/4.00): (1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49); Expected Cumulative grade point average in the graduation (/4.00): (1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49); Course ID; and OUTPUT: Grade (0: Fail, 1: DD, 2: DC, 3: CC, 4: CB, 5: BB, 6: BA, 7: AA). The models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, LGBM classifier, Gradient Boosting, and XGB classifier. Three feature scaling used in machine learning are raw, minmax scaler, and standard scaler. Finally, you will develop a GUI using PyQt5 to plot cross validation score, predicted values versus true values, confusion matrix, learning curve, decision boundaries, performance of the model, scalability of the model, training loss, and training accuracy.