INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

DATA WAREHOUSING AND DATA MINING LABORATORY								
V Semester: IT								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
AITC22	Core	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		1	0	2	2	30	70	100
Contact Classes: 12	**Tutorial Classes: Nil**	**Practical Classes: 45**				**Total Classes: 45**		
Prerequisites: There is no prerequisite to take this course								

I. COURSE OVERVIEW:

This course helps the students to practically understand a data warehouse, techniques and methods for data gathering and data pre-processing using different tools. The different data mining models and techniques will be discussed in this course. The main objective of this lab is to impart the knowledge on how to implement classical models and algorithms in data warehousing and data mining and to characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.

II. COURSE OBJECTIVES

The students will try to learn:

I. Build Data Warehouse and Explore WEKA
II. Perform data preprocessing tasks and demonstrate performing association rule mining on data sets.
III. Perform classification, clustering and regression on data sets

III. COURSE OUTCOMES:

At the end of the course students should be able to:

CO 1 **Perform** pre-processing statistical methods for any given raw data.
CO 2 **Apply** Association rule process for a given dataset by using A priori algorithm.
CO 3 **Demonstrate** Association rule process for a given dataset by using FP-Growth algorithm**.**
CO 4 **Experiment** Classification rule process for a given raw data by using Decision tree and ID3 algorithm
CO 5 **Analyze** Classification rule process for a given raw data by using Decision tree and naive Bayes Algorithm.
CO 6 **Categorize** a given dataset by using Clustering Algorithm**.**

IV. COURSE CONTENT

EXERCISES FOR DATA WAREHOUSING AND DATA MINING LABORATORY

Note: Students are encouraged to bring their own laptops for laboratory practice sessions.

1. Getting Started Exercises

Introduction

Pentaho Reporting is a suite (collection of tools) for creating relational and analytical reports. It can be used to transform data into meaningful information. Pentaho allows generating reports in HTML, Excel, PDF, Text, CSV, and xml. Before start proceeding with Pentaho, should have prior exposure to Core Java, Database Concepts, and SQL Queries.

Steps for Installation of Pentaho:

http://www.pentaho.com/

1. Extract the downloaded file prd-ce-3.7.0-stable.zip using an Unzip tool.
2. Copy the extracted folder (prd-ce-3.7.0-stable) into c:\ directory.
3. Open c:\prd-ce-3.7.0-stable\report-designer directory.
4. Start Pentaho Reporting Designer by double-clicking on the reportdesigner.bat file

Steps for Installation of Weka:

1. Download the software (Weka Tool Kit) as your requirements from the below given link.
   http://www.cs.waikato.ac.nz/ml/weka/downloading.html
2. The Java is mandatory for installation of WEKA so if you have already Java on your machine then download only WEKA else download the software with JVM.
3. Then open the file location and double click on the file. Click on Run.
4. Click Next to continue
5. If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Weka.
6. Check the components you want to install and uncheck the components you do not want to install. Click Next to continue
7. Setup will install Weka 3.8.6 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

1.1 Build Data Warehouse

Build a Data Warehouse/Data Mart (using open-source tools like Pentaho Data Integration tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.).

Installing Pentaho and Weka

Resource:

http://www.pentaho.com/

1. Identify source tables and populate sample data.
2. Design multi-dimensional data models namely Star, snowflake and Fact constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, Manufacturing, Automobile, etc.).
3. Write ETL scripts and implement using data warehouse tools.

Try

1. Perform various OLAP operations such slice, dice, roll up, drill up and pivot.
2. Explore visualization features of the tool for analysis like identifying trends etc.

1.2 Explore WEKA Data Mining /Machine Learning Toolkit

1. Download and/or installation of WEKA data mining toolkit.
2. Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
3. Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel Cluster panel, associate panel and Visualize panel).
4. Study the arff file format.
5. Explore the available data sets in WEKA.

Try

1. Load a data set (ex. Weather dataset, Iris dataset, etc.).
2. Load each dataset and observe the following:
    - List the attribute names and they type
    - Number of records in each dataset
    - Identify the class attribute (if any)
    - Plot Histogram
    - Determine the number of records for each class.
    - Visualize the data in various dimensions

Installing Pentaho and Weka

http://www.cs.waikato.ac.nz/ml/weka/

2. Perform Data Preprocessing tasks and Demonstrate Performing association rule mining on date sets.

2.1 Explore Various options available in Weka for Preprocessing on data set

Explore various options available in Weka for preprocessing data and apply (like Discretization Filters, Resample filter, etc.) on following dataset

Resource:

https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/iris.arff

Try

1. Preprocessing the data on https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/soybean.arff.
2. Apply the resample filter and discretization filter.

2.2 Find Interesting patterns from the Data set using Association Rule Mining.

Load each dataset into Weka and run Appriori algorithm with different support and confidence values. Study the rules generated.

- Load the dataset into Weka tool.
- Find the Patterns using Appriori Algorithms.
- Determine the Best Patterns with different support & confidence values for that selected dataset

Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/supermarket.arff

Try

1. Determine the Best Patterns using FP Growth Tree algorithm for different support & Confidence Values for the Dataset. https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/weather.arff

2.3 Finding the Effect of Discretization in association rule generation process

Apply different discretization filters on numerical attributes and run the Appriori association rule algorithm. Study the rules generated. Derive interesting insights and observe the effect of discretization in the rule generation process.

Data Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/breast-cancer.arff

Try

1. Apply different discretization filters on numerical attributes and run the FP Growth association rule algorithm.
2. Trace the results of using the Appriori algorithm on the grocery store example with support threshold s=33.34% and confidence threshold c=60%. Show the candidate and frequent item sets for each database scan. Enumerate all the final frequent item sets. Also indicate the association rules that are generated and highlight the strong ones, sort them by confidence.

3. Exercises on performance of classification on data sets

3.1 Load each dataset into Weka and run Id3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic

Load each dataset into Weka and run Id3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.

Data Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/glass.arff

Try

1. Perform Naïve-bayes classification and k-Nearest Neighbor classification
2. Interpret the results obtained

3.2 Apply cross-validation strategy with various fold levels compare accuracy

Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix and derive Accuracy, F-measure, TP rate, FPrate, Precision and Recall values. Apply cross-validation strategy with various fold levels and compare the accuracy results

Data Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/segment-challenge.arff

Try

1. Check the results of cross validation with 10 folds & 20 folds

3.3 Perform Naïve-bayes classification and k-Nearest Neighbor classification.

Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbor classification. Interpret the results obtained.

Data Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/vote.arff

Try

1. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset

4. Exercises on performing clustering on data sets

4.1 Perform simple k-means clustering algorithm with different values of k (number of desired clusters)

Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.

- Load the dataset into Weka tool.
- Use K-Means clustering algorithms.
- Get the sum of squared errors, centroids, No. of iterations & clustered instances as represented below

Data Resource
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/iris.arff

Try

1. Perform k-means clustering algorithm with k=2 for the data set https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/weather.arff

4.2 Explore DBSCAN clustering techniques available in Weka

Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
Load the dataset into Weka tool.
Use K-Means clustering algorithms.
Get the sum of squared errors, centroids, No. of iterations & clustered instances.

Data Resource:
https://www.kaggle.com/datasets/iamsouravbanerjee/customer- shopping-trends-dataset.

Try

1. Perform Gaussian Mixture Model clustering technique and check the accuracy.
2. Perform BIRCH Model clustering technique and check the accuracy.

4.3 Explore visualization features of Weka to visualize the clusters.

Explore visualization features of Weka to visualize the clusters. Derive interesting insights. Observe different clustering outputs for dataset by changing those X-axis, Y-axis, Color & Jitter options

Data Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/ReutersGrain-test.arff

Try

1. Visualize the Gaussian Mixture Model clustering technique output and check the accuracy.
2. Visualize the BIRCH Model clustering technique output and check the accuracy.

5. Exercises on performing Regression on data sets

5.1 Build Linear Regression model

Load each dataset into Weka and build Linear Regression model. Study the clusters formed. Use Training set option. Interpret the regression model and derive patterns and conclusions from the regression results.

- Load the dataset into Weka tool
- Use Simple Linear Regression algorithm
- The patterns should be visually displayed

Data Resource:

https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/diabetes.arff

Try

1. Find out if there is any correlation between these two variables
2. Find the best fit line for the dataset.
3. How the dependent variable is changing by changing the independent variable**.**

5.2 Perform cross-validation and percentage split in Linear Regression Model

Use options cross-validation and percentage split and repeat running the Linear Regression Model. Observe the results and derive meaningful results. Load the dataset into Weka tool, use multiple Linear Regression algorithm and cross validate with 10 folds.

Data Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/labor.arff

Try

1. Predicts the value for new test value using the model and check the accuracy.
2. Change the test data and build new regression model and check the accuracy.

5.3 Explore Simple linear regression technique that only looks at one variable

Load each dataset into Weka and build Linear Regression model. Study the clusters formed. Use Training set option. Interpret the regression model and derive patterns and conclusions from the regression results.

- Load the dataset into Weka tool
- Use Simple Linear Regression algorithm
- The patterns should be visually displayed

Data Resource:
https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/labor.arff

Try

1. Check the model for new test date and check the accuracy.
2. Build a new regression model and check the accuracy.

6. Credit Risk Assessment – The German Credit Data

6.1 Find all the categorical (or nominal) attributes and the real-valued attributes separately.

List all the categorical (or nominal) attributes and the real-valued attributes separately.

Data Resource:

https://www.kaggle.com/datasets/uciml/german-credit

Try

1. List all the categorical (or nominal) attributes and the real-valued attributes separately https://www.kaggle.com/datasets/zain280/car-dataset

6.2 What attributes do you think might be crucial in making the credit assessment.

What attributes do you think might be crucial in making the credit assessment. come up with some simple rules in plain English using your selected attributes

Data Resource:

https://www.kaggle.com/datasets/uciml/german-credit

Try

1. Use another dataset and find the accuracy using same model

6.3 Identification of outliers in the data

Identify outliers using z-score method on above data frame.

Hint:

```
stats. Score () function can be used to identify outliers of a data frame.
```

Try

1. Write a program to identify outliers using IQR (Interquartile Range) Method on above data frame.
2. Write a program to identify outliers using Tukey's Method on above data frame.

7. Handling missing values

7.1 Handling missing values - Removing

Create a data frame 'data' containing the data from the following dataset https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv

Hint:

```
# Identify missing values in the data frame using '.is null ()' method
```

Consider given a data frame 'data' with missing values, use the. Drona() method to remove all rows

containing missing values. Display the resulting data frame. Also verify the same by printing the summary reports. Also print the rest rows and columns summary.

Hint:

```
#Use the '. Drona ()' method to remove all rows

Write code
```

Try

1. Consider the above data frame with missing values, calculate the percentage of missing values in each column and create a summary table showing the columns with the highest percentage of missing values.
2. Consider the above data frame, calculate and display the number of missing values in each column individually. Sort the columns by the number of missing values in descending order.

7.2 Handling missing values - Imputation

Impute the missing values in the numerical column of the given data frame 'data' with the mean of the non-missing values.

Hints

```
# Sample Data frame with missing values

# Impute missing values with the mean
```

Try

1. Consider given data frame with missing values in a numerical column, impute the missing values in that column with the median of the non-missing values.
2. Consider given data frame with missing values in a numerical column, impute the missing values in that column with the median of the non-missing values.
3. Consider given data frame with missing values, impute missing values in a specific column with a constant value of your choice (e.g., 0).

7.3 Handling missing values - Interpolation

Consider the given data frame with missing values in a numerical column, use linear interpolation to impute the missing values in that column.

Hint:

```
# import libraries

# Perform linear interpolation on the 'Value' column using '. Interpolate ()' method
```

Try

1. Custom interpolation function that takes into account domain-specific knowledge or specific data characteristics to impute missing values in a data frame.
2. Consider above data frame with time-ordered data and a numerical column containing missing values, use interpolation techniques that consider the time steps between observations to impute missing values more accurately.

8. Exercises on Time Series Data

8.1 Time series data preparation

Load a time series dataset daily-minimum-temperatures-in-me.csv into a pandas data frame from the https://www.kaggle.com/datasets/shenba/time-series-datasets. Preprocess the data by setting the datetime column as the index and ensuring that it's in the correct datetime format.

Hint:

```
#import libraries

# preprocess the data by setting the datetime column as the index in correct format using '.to_datetime()' method
```

Try

1. Load a time series dataset with irregular time intervals (timestamps) into a Pandas data frame. Resample the data to have a regular time interval (e.g., daily, hourly) and fill any missing data with appropriate values (e.g., forward fill, backward fill, interpolation) on the following data:
   data = {'Timestamp': ['2022-01-01 08:00:00', '2022-01-01 10:30:00', '2022-01-02 09:15:00'], 'Value': [10, 15, 20]}
2. Load above time series dataset that includes time zone information. Convert the timestamps to a common time zone (e.g., UTC) in pandas.
3. Load a time series dataset that exhibits seasonality (e.g., monthly sales data). Create additional columns to represent the year, quarter, month, day of the week, or any other seasonal components to facilitate seasonal analysis on the following data:
   data = {'Date': ['2022-01-15', '2022-02-20', '2022-03-10', '2022-04-05', '2022-05-18'], 'Sales': [1000, 1200, 800, 1100, 1500]}

8.2 Time series aggregation

Load a daily time series dataset and aggregate it to monthly intervals. Calculate summary statistics such as the monthly mean, sum, minimum, and maximum values for the following data:
data = {'Date': pd. date_range (start='2022-01-01', end='2022-12-31', freq='D'), 'Value': [10, 15, 20, 18, 25, 30] * 61} # Six values repeated for each day in 2022

Hint:

```
# Create a dataframe from the daily time series data
# Aggregate to monthly intervals and calculate summary statistics using '.agg() method
```

Try

1. Load an hourly time series dataset and aggregate it to daily intervals. Calculate the daily total, average, or maximum values for the following data:
   data = {'Timestamp': pd.date_range(start='2022-01-01', end='2022-01-03', freq='H'), 'Value': [10, 15, 20, 18, 25, 30, 22, 28, 35, 40, 32, 38, 45, 50, 42, 48, 55, 60]}
2. Load a weekly time series dataset and aggregate it to monthly intervals. Calculate monthly summaries, such as the total sales for each month.
3. Load a time series dataset as following with seasonal patterns (e.g., daily temperature readings). Aggregate the data to a higher level of granularity (e.g., monthly) while preserving important seasonal

information.
data = {'Date': pd. date_range (start='2022-01-01', end='2022-12-31', freq='D'), 'Temperature': [30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52] * 30 # Daily temperature values (seasonal pattern)}

9. Exercises on working with Data Tables

9.1 Exercises on joining data tables

Join two data tables along rows from the dataset as used two times 'https://github.com/17rsuraj/data-urious/blob/master/TowardsDataScience/Dummy_course_data.xlsx'.

Hint:

```
import matplotlib library
# Use 'join ()' function to add two data tables
```

Try

1. Join two data tables along columns from the dataset using 'merge () function.
2. Join two data tables along rows from the dataset using 'join () function.
3. Join two data tables along columns from the given dataset using 'merge () ' function.

9.2 Exercises on pivot tables

Create a Pivot table with multiple indexes from the data set of "https://github.com/ven-27/datasets/blob/master/titanic.csv" file.

Hint:

```
# Import necessary libraries

# Create a data frame

# Use 'pivot table()' function to create a pivot table
```

Try

1. Create a Pivot table on the given dataset and find survival rate by gender.
2. Create a Pivot table on the given dataset and find survival rate by gender, age wise of various classes.
3. Create a Pivot table on the given dataset and calculate number of women and men were in a particular cabin class.

9.3 Exercises on contingency tables

Make a contingency table between the 'Age' and 'Survived' columns on the above dataset.

Hint:

```
# Import necessary libraries

# Use the 'crosstab ()' function to create contingency table
```

Try

1. Make a contingency table between the p class and 'Sex' columns on the above dataset.
2. Make a contingency table between the 'passenger Id' + 'Sex' and 'Survived' columns on the above dataset.

9.4 Exercises on grouping data

Split the above dataset into groups based on passenger class and also check the type of Group By () object.

Hint:

```
# Use the 'requests' and the 'Beautiful Soup' libraries for web scraping and data extraction
```

Try

1. Split the above data frame into groups based on survived and passenger class.
2. Split the above data frame into groups based on single column and multiple columns. Find the size of the grouped data.

10. Exercises on Web Scraping

10.1 Scraping a list of items from a website

Scrape a list of articles from a news website https://ywepaper.thehindu.com/reader.

Hint:

```
# Use the 'requests' and the 'Beautiful Soup' libraries for web scraping and data extraction
```

Try

1. Scrape a List of Products from an E-commerce Website https://www.amazon.in
2. Scrape a List of Job Postings from a Job Search Website https://www.naukri.com/engineering-jobs?src=discovery_trendingWdgt_homepage_srch.
3. Scrape a List of Books from an Online Bookstore https://www.bookswagon.com

10.2 Scraping data from a table

Scrape data from a table on a world health organization website:

https://rho.emro.who.int/Indicator/TermID/46.

Hint:

```
# Use the 'requests' and the 'Beautiful Soup' libraries for web scraping and data extraction
```

Try

1. Scrape a Table with Headers on an above world health organization website.
2. Scrape a Table with Pagination on an above world health organization website.

10.3 Scraping images from a website

Locate the all-images URLs in the webpage's HTML to do scraping from the above website.

Hint:

```
# Use the 'requests' and the 'Beautiful Soup' libraries for web scraping and
data extraction
```

Try

1. Scrape Images from a Gallery on the world health organization website https://www.emro.who.int/media/multimedia/posters.html.
2. Scrape Images from a Search Result Page as https://www.emro.who.int/search/en/index.htm?q=images.

10.4 Scraping data with pagination

Scrape data with pagination from a website involves navigating through multiple pages to collect all the desired information.

Hint:

```
# Use the 'requests' and the 'Beautiful Soup' libraries for web scraping and
data extraction
# Set the number of pages to scrape
# Print the scraped data
```

Try

1. Scrape Data with AJAX Pagination.
2. Scrape Images from a Search Results Page as https://www.emro.who.int/search/en/index.htm?q=images.
3. Write a program to implement error handling to gracefully handle HTTP errors, such as 404 or 503, and exceptions that may occur during scraping, such as timeouts.

10.5 Scraping with user-agent

scrape a webpage with a custom User-Agent as:
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36'}

Hint:

```
# use the 'requests' library to set a User-Agent header to simulate a web
browser request and avoid being blocked
```

Try

1. Modify the program to use a different User-Agent string. You can find various User- Agent strings online, or you can generate your own to mimic different browsers or devices.
2. Extend the program to scrape a specific page on the website with the custom User- Agent header.
3. Modify the program to scrape multiple pages on the website with the custom User- Agent header. Use a loop to iterate through different URLs.
4. Create a list of User-Agent strings and randomly select one for each request. This exercise simulates a more dynamic scraping behavior.

11. Exercises on Visualization-I

11.1 Exercises on line plot

Plot a line plot on the following dataset
https://gist.githubusercontent.com/seankross/a412dfbd88b3db70b74b/raw/5f23f993cd87c283ce766e7ac6b329ee7cc2e1d1/mtcars.csv.

Hint:

```
# Use the Matplotlib library

# The method Pl. Plot (x, y) can be used to plot the line plot
```

Try

1. Line plot customizing Line Styles and Markers**.**
2. Line plot adding Grid Lines.
3. Line plot adding Logarithmic Scale on the Y-axis.

11.2 Exercises on scatter plot

Plot scatter plot to visualize the relationship between two variables on the followingdataset"https://github.com/CoreyMSchafer/code_snippets/blob/master/Python/Matplotlib/07-ScatterPlots/2019-05-31-data.csv**".**

Hint:

```
# Use the Matplotlib library

# The method plt.scatter(x, y) can be use to plot the line plot
```

Try

1. Plot a scatter Plot with Color and Marker Customization.
2. Plot a scatter Plot with Size and Transparency.
3. Calculate correlations between variables to quantify the strength and direction of a relationship between the variables.

11.3 Exercises on bar chart

Plot bar plot to visualize the relationship between two variables on the following dataset"https://github.com/CoreyMSchafer/code_snippets/blob/master/Python/Matplotlib/07-ScatterPlots/2019-05-31-data.csv".

Hints

```
# Use the Matplotlib library
# Create a basic vertical bar chart using method 'plt.bar(categories, values)' with parameters
```

Try

1. Plot a vertical bar plot on the above data.
2. Plot a grouped bar plot on the above data.
3. Plot a bar plot on the above data using seaborn library.

11.4 Stacked bar plot

Plot Stacked bar plot to visualize the relationship between two variables on the following dataset "https://github.com/CoreyMSchafer/code_snippets/blob/master/Python/Matplotlib/07- Scatterplots/2019-05-31-data.csv".

Hint:

```
# Use the Matplotlib library
# Create a basic stacked bar chart using method 'plt.bar(categories, values1)' and
'plt.bar(categories, values2, bottom=values1)' parameters
```

Try

1. Plot a horizontal stacked bar plot on the above data using Matplotlib library.
2. Plot a horizontal stacked bar plot on the above data using seaborn library.

12. Exercises on Visualization-II

12.1 Exercises on histogram

Plot a Histogram for visualizing the distribution of a dataset using Matplotlib library on the following dataset
https://gist.githubusercontent.com/seankross/a412dfbd88b3db70b74b/raw/5f23f993cd87c283ce766e7ac6b329ee7cc2e1d1/mtcars.csv.

Hint:

```
# Use the Matplotlib library

# Create a histogram with bin parameters in 'plt.hist(data, bins)' method
```

Try

1. Plot a histogram with different Customization.
2. Plot a Stacked Histograms on the above data points.

12.2 Exercises on density plots

Plot a density plot on given the dataset 'car_crashes' in which is the most common speed due to which most of the car crashes happened.

Hint:

```
# Use the Matplotlib & seaborn library

# Create a histogram with bin parameters in 'plot.density()' method
```

Try

1. Plot a density plot on given the dataset 'tips' and calculate what was the most common tip given by a customer.

12.3 Exercises on box plot

Plot a box plot on given the dataset 'https://github.com/ven- 27/datasets/blob/master/titanic.csv' in Which whether the person has survived or not during the sink of titanic and different details of the person.

Hint:

```
# Use the Matplotlib & seaborn library

# Create a box plot with bin parameters in 'plt.boxplot( )' method
```

Try

1. Plot a box plot of distribution of survived based on the age of the passenger on given the dataset.
2. Box plot of distribution of survived based on the Class of the passenger on given the dataset.

12.4 Exercises on pair plots

Write a program to generate pair plots for pairwise relationship of variables on the 'penguins' dataset from seaborn module.

Hint:

```
# Use the seaborn library

# Create a pair plot using 'seaborn.pairplot( )' method
```

Try

1. Generate pair plots for pairwise relationship of variables on the above dataset to assigning a hue variable as 'species.
2. Generate pair plots for pairwise relationship of variables on the above dataset to assigning the kind parameter as 'kde' to draw KDEs using kdeplot( ) funtion.
3. Generate pair plots for pairwise relationship of variables on the above dataset to assigning a vars value as x vars and y vars to select the variables to plot.

13. Final Notes

Dealing with the raw data which has no longer been accessible and cannot be utilized should be processed to use efficiently. Then here comes data-wrangling which helps to turn non-resourceful (raw) data into valuable data which in turn returns valuable information. Each step in the process of wrangling the data is to the best possible analysis. The outcome is expected to generate a robust and reliable analysis of the data.

The problems in this tutorial are certainly NOT challenging. Check out these sites:

- Educative - https://www.educative.io/courses/data-wrangling-with-python/coding-challenges-on-
- iMocha - https://www.imocha.io/tests/data-wrangling-with-python-test
- Kaggle - https://www.kaggle.com/code/prakharrathi25/data-wrangling
- Github - https://moderndive.github.io/moderndive_labs/static/PS/PS03_data_wrangling.html
- Rpubs - https://rpubs.com/Rokshana_Data_knowledge/891691
- Towardsdatascience - https://towardsdatascience.com/conquer-the-python-coding-round-in-data-science-interviews-5e27c4513be3
- Into the minds - https://www.intotheminds.com/blog/en/the-11-challenges-of-data-preparation-and-data-wrangling/

Student must have any one of the following certifications:

- Coursera - Data Mining and AB Testing with SQL
- Coursera - Data Warehouse and Data Mining
- Coursera – Data Warehouse and Data Wrangling with Pandas
- Coursera - Fundamental Tools of Data Ming

- GeeksforGeeks – Data Wrangling in Python
- Stanford – Data Mining
- Udemy – Data Wrangling with Python
- Pluralsight - Data Wrangling with Python 3
- Harvard University - Data Science: Preprocessing
- Udacity - Advanced Data Mining

V. TEXT BOOKS:

1. J.Han,M.Kamber, "Data Mining: Concept and Techniques", Academic Press, Morgan Kanfman Publishers, 3rd Edition, 2019.

VI. REFERENCE BOOKS:

1. Dr. Tirthajyoti Sarkar, Shubhadeep, "Data Wrangling with Python: Creating actionable data from raw sources", Packt Publishing Ltd, 2019.
2. Wes McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", O'Reilly, 2nd edition,2018.
3. Alex Berson, Stephen J.Smith, "Data Warehousing, Data Mining & OLAP", Tata McGraw-Hill, 10th edition, 2007
4. PieterAdrians, DolfZantinge, "Data Mining", Addison Wesley, Peter V, 2000.

VII. ELECTRONICS RESOURCES

1. https://www.tutorialspoint.com
2. http://www.anderson.ucla.edu
3. https://www.smartzworld.com
4. http://iiscs.wssu.edu

VIII. MATERIALS ONLINE

1. Course template
2. Lab Manual