Assignment No. 1

Name: Akanksha Deshpande.
Roll no: 20C0021
Class: TE Comp A
Batch: B
Sub: DSBDA

Title: Data Wrangling I

Problem Statement:

Perform the following operations using Python on any open source dataset (eg. data.csv)
(1) Import all the required Python libraries.
(2) Locate an open source data from the web (Kaggle). Provide a clear description of data and its source.
(3) Load the dataset into pandas dataframe.
(4) Data Processing - Check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions, types of variables, etc. Check dimensions of dataframe.
(5) Data Formatting and Data Normalization: Summarize the types of variables by checking the data types of variables in the data set. If not in proper data type, apply correct type conversions.
(6) Turn categorical variables into quantitative variables

In addition to the code and output, explain every

operation that you do in the above steps and explain everything that you do to import/read/scrape the dataset.

Objectives of assignment:
Students should be able to perform the data wrangling operations using Python on any open source dataset.

Prerequisite:
(1) Python programming concepts.
(2) Concepts of Data processing, data formatting, data normalization and data cleaning.

Theory:
1] Basic Terminologies
(1) Introduction to Dataset:
A dataset is a collection of records, similar to a table. Records are similar to table rows but the columns can contain not only strings or numbers but also nested data structures such as lists, maps and other records.

(2) Instance
A single row of data is called an instance.

(3) Feature
A single column of data is called a feature.

(3) Data Type: Features have datatype. They may be real or integer valued or may be categorical or ordinal value.

(4) Datasets:
A collection of instances is called as datasets.

(5) Training Datasets:
A dataset that we feed into our machine learning algorithm to train our models.

(6) Testing Dataset
The dataset that we use to validate the accuracy of our model but its not used to train the model.

2] Python Libraries for Data Science.

(1) Pandas:
It is an open source python package that provides high performance, easy to use data structures and data analysis tools for labeled data.

(2) Numpy:
Numpy is a general purpose array programming package that provides high performance to multi-dimensional arrays and tools to work with arrays.

~~(3) Mat-pottlib~~: (3) Matplotlib
With matpolib we can create stories with the data visualized. Also 2D figures can be plotted.

(4) Seaborn:
It is defined as the data visualization library based on matplotlib that provides a high level interface for drawing attractive and informative statistical graphics.

(5) Scikit learn:
It is a robust machine learning library for python. It features ML algorithms like svm, random forests, mean shift, cross validation and so on.

3] Description of dataset:
The iris dataset was used in RA Fisher's classic 1936 paper. It includes three iris species with 50 samples as well as some properties about each flower. The columns in the dataset are:
(1) Id
(2) Sepallength
(3) Sepalwidth
(4) Petal length
(5) Petal width
(6) Species

4] Dataframe Operations:

Sr.no	Dataframe function	Description
1.	dataset.head(n=5)	Returns the first n rows.
2.	dataset.tail(n=5)	Returns the last n rows.
3.	dataset.index	Returns index of dataset.
4.	dataset.columns	Returns labels of columns in the dataset.
5.	dataset.shape	Returns a tuple representing the dimensions of the dataset.
6	dataset.dtypes	Returns datatypes in database.
7	dataset.columns.values	Returns column values in array format.
8.	dataset.describe(include="all")	Generates descriptive statistics.
9.	dataset['column name']	Read data column wise.
10.	dataset.sort_index(axis=1, ascending=false)	Sort object by labels.
11.	dataset.sort_values(by="column name")	Sort values by column name.
12.	dataset.iloc[5]	Location based indexing for selection by position.
13.	dataset[0:3]	Selecting via [], which slices the rows
14.	dataset.loc["col1", "col2"]	Selection by label.

15.	dataset.iloc[:n, :]	Subset of first n rows.
16.	dataset.iloc[:, :n]	Subset of first n columns.
17.	dataset.iloc[:n, :n]	Subset of first n rows and 'n columns

Checking of missing values in dataset.

isnull()or isna() are the functions used to check missing values or null values in dataset.

a) Is there any missing value :
Dataframe.isnull().any()

b) Count of missing values across each column
dataframe.is null.sum().sum()

c) Count row wise missing values
dataframe.isnull.sum(axis=1)

d) Count missing values of specific column
dataframe.col-name.isnull().sum()

5] Data Formatting and Normalization

(a) Formatting :
Ensuring all data formats are correct is initial cleaning process
Functions used:
(1) df.dtypes
(2) df['col-name'] = df['col.name"].astype("int")

b) Data Normalization

Mapping all normal data values into a uniform scale is involved in data normalization

Algorithm:

Step 1: Import all the required Python libraries

import pandas as pd.

Step2: Load the dataset into pandas dataframe

iris = pd.csv("/content/Iris.csv")

iris

Step3: Perform the data preprocessing steps.

iris.head()

iris.info()

Iris.describe()

iris.shape

iris.dtypes

Step4: Check for null values

iris.isnull().any()

iris.isnull().sum()

isis.isnull()

Iris.isna()

Step 5: Perform Data Formatting and normalization operations.

(i) Import sklearn library for preprocessing

from sklearn import preprocessing

(ii) Create a minimum and maximum processor object.

min_max_scaler = preprocessing.MinMaxScaler().

(iii) Feature seperation from class label

X = iris.iloc[:, :4]

(iv) Create an object to transform the data to fit minmax processor.

X_scaled = min_max_scaler.fit_transform(X)

(v) Run normalizer on dataframe

iris_normalized = pd.Dataframe(X_scaled)

(vi) View the dataframe

iris_normalized.

Conclusion:

In the above lab assignment, we have explored the functions of the python library for Data Processing, & Data Wrangling techniques.

Assignment Questions:

Q1) Explain DataFrame with suitable example:

→ A DataFrame is a datastructure that organizes data into a 2-dimensional table of rows and columns, like a spreadsheet.

eg:

	Roll no	Name	Result
1.	011	Riya	Pass
2.	012	Pranav	Fail
3.	013	Karan	Pass
4.	014	TiNa	Pass

Q2) What is the limitation of label encoding method?

→ Though label encoding is straight but it has the disadvantage that the numeric values can be misinterpreted by algorithms having some sort of hierarchy in them. This ordering issue is addressed in another common alternative approach called 'One Hot Encoding'.

Q3) What is the need of data normalization?

→ 1) To eliminate redundant errors
2) Minimize data modification errors

3) Simplify the query process
4) Improve workflow
5) Lesser costs and increases security.

Q4) What are the different techniques for Handling the missing data?

Ans: Different techniques to handle the missing data are:

1) Deleting rows with missing values:
One of the ways to handle missing values is the deletion of the rows or columns that contain missing values.

2) Replacing with arbitrary values:
You can replace the missing values with some arbitary values using fillna().

3) Interpolation:
Missing values can also be handled using interpolation. Pandas interpolate method can be used to replace the missing values with different interpolation methods like 'polynomial', 'linear', 'quadratic'.

Assignment No. 2

Title : Data Wrangling II

Problem Statement:

Create an "Academic Performance" dataset of students and perform the following operations using Python.
1) Scan all variables for missing values and inconsistencies. If there are missing values, use any of the suitable techniques to deal with them.
2) Scan all the numeric values for outliers. If there are outliers, use any of the suitable techniques to deal with them.
3) Apply data form transformations on at least one of the variables. The purpose of transformation should be one of the following reasons : to change the scale for better understanding of the variable, to convert a non linear relation into a linear one or to decrease the skewness and convert the distribution into a normal distribution.

Objective of Assignment:
Student should be able to perform the data wrangling operation using Python on any open source dataset.

Prerequisite:
(1) Basics of Python Programming.

(2) Concept of Data processing, Data forming, Data Normalization and Data Cleaning.

Concepts of Theory:
(1) Creation of Dataset using microsoft excel.
(2) Identification and Handling of Null Values
(3) Identification and Handling of Outliers.
(4) Data Transformation for the purpose of:
   (a) To change the scale for better understanding
   (b) To decrease the skewness and convert distribution into normal distribution.

Theory:
1] Creation of Dataset using micmsoft excel.

The dataset is created in 'csv' format. Name of the dataset is 'Academic Performance'.
Features of dataset are: RollNo, Name, DSBDA, AI, CC, WT, TotalMarks, Percentage, Result.
Number of instances : 26
To fill the values in the dataset the RANDBETWEEN is used. It returns a random integer number between the range specified.
Syntax: RANDBETWEEN (bottom, top)
where,
bottom = smallest integer
top = largest integer.

2] Identification and handling of Null values:

Missing data can occur when no information is provided for one or more items or for a whole unit. In dataframe sometimes many datasets simply arrive with missing data either because it never existed or existed but not collected. In Pandas missing data is represented by two values:

(i) None : It is a python singleton object.
(ii) NaN : It is a special floating point value.

The functions used for detecting, removing and replacing null values are:

(i) Detection : isnull() ; isna() ; notnull()
(ii) Removal : dropna() ; fillna()
(iii) Replace : replace()

3] Identification and handling of outliers:

Outlier in a dataset is an observation that lies far from rest of the observations. An outlier may occur due to variability in the data. The methods used to detect the outliers are:

(1) Boxplot
(2) Z-score
(3) Inter Quartile Range (IQR)

Methods used to treat the outliers:

(1) Trimming

(2) Quantile based flooring and capping

(3) Mean/Median imputation

4] Data Transformation:

Data Transformation or ETL (Extract, Transform, Load) is the process of converting raw data into a format or structure that would be more suitable for model building and data discovery. The data transformation involves steps:

(1) Smoothing: It is used to remove noise from dataset using some algorithms. It helps in predicting patterns.

(2) Aggregation: It is a method of storing and presenting data in summary format. It maintains data accuracy.

(3) Generalization: It converts low-level data attributes to high level data attributes using concept hierarchy.

(4) Normalization: It involves converting all data variables into given range. Some of the normalization techniques are:

(1) Min-max normalization :
In this method, the original data is linearly transformed.

(2) Z-score normalization :
In this method, the values of attributes are normalized based on the mean A and its standard deviation.

(3) Normalization by decimal scaling :
It normalizes the values of an attribute by changing the position of their decimal points.

Algorithm :

Step 1 : Import libraries
```
import pandas as pd
import numpy as np
import seaborn as sns
```

Step 2 : Load the dataset into dataframe.
```
df = pd.read_csv("/Academic_performance.csv")
df
```

Step 3 : Using isnull() method to check null values.
```
df.isnull()
```

Step 4 : Handling the null values with fillna() and replace() method.

```
df.replace (np.nan, value = 0)
df.fillna(1)
df['WT'] = df['WT'].fillna(df['WT'].mean())
```

Step 5 : Identify the outliers by boxplot method.

```
df.boxplot ()
```

Step 6 : Print the outliers for each coloumn with reference to box plot, for instance assume coloumn 'WT' has outliers.

```
print(np.where(df['WT'] <25))
```

Step 7 : Removing the outliers

```
new_df = df
for i in sample_outliers :
    new_df.drop (i, inplace = True)
new_df.
```

Step 8 : For data transformation, checking the distribution with skewness.

```
import seaborn as sns
df.skew
```

Step 9 : Checking the distribution of variables using KDE plot. for all coloumns

```
sns.kdeplot (df.WT)
```

Conclusion:

In this way we have explored the functions of python library for Data identifying and handling the outliers. Data Transformations Techniques are explored with the purpose of creating the new variable and reducing the skewness from datasets.

Assignment Questions:

Q1. Explain methods to detect outliers

Ans: Outliers are observation in a given dataset that lies far from the rest of the observations. They are detected with following methods:

(1) Boxplots – Boxplot summarizes the data using 25th, 50th and 75th percentiles. It captures summary of data with box and whiskers.

(2) Scatterplots — Scatter plots present the relationship between two variables in a dataset.

(3) Inter Quantile Range (IQR) :— It measures the midspread of data. It is calculated by $IQR = Q_3 - Q_1$.

(4) Z score – It is used to convert the data into another dataset with mean = 0. Here, the z score is calculated as,

$$Z = \frac{x_i - \bar{x}}{s}$$

Q2. Explain data transformation methods.

Ans: Data transformation methods are:

(1) Smoothing: It is used to remove noise from dataset using some algorithms. It helps to detect patterns.

(2) Aggregation: It is a method of storing and

presenting data in summary format. It maintains data accuracy.

(3) Generalization: It converts low level data attributes to high level data attributes using concept hierarchy.

(4) Normalization: It involves converting all data-variable into given range.

(5) Feature construction: Here new attributes are created and applied to the given set of attributes.

Q3) Write an algorithm to display statistics of Null values.

Ans:
Step 1) Import required libraries that is pandas and numpy.
Step 2) Load the dataset in dataframe object df.
Step 3) Display dataframe.
Step 4) Use isnull() function to check null values in the dataset.
Step 5) Use isnull().isany().sum() to display statistics of null values in the dataset.

Q4) Write an algorithm to replace outlier values with mean of variable.

Ans:
Step 1) Import all required libraries.
Step 2) Load the dataset into dataframe df and display

the dataframe.

Step 3) Plot the boxplot for any column. (Consider 'WT' here)

Step 4) col = ['WT']

df.boxplot(col)

Step 4) Outliers are seen in boxplot.

Step 5) Calculate the median of 'WT' by using sorted_score

median = np.median(sorted_score)

median

Step 6) Replace the upper bound outliers using median value.

new_df = df

new_df['WT'] = np.where(new_df['WT'] > upr_bound, median, redefined_df['WT'])

Step 7) Display new_df.

Step 8) Replace the lower bound outliers using median value.

new_df = df

new_df['WT'] = np.where(new_df['WT'] < lwr_bound, median, redefined_df['WT'])

Step 9) Display new_df

Step 10) Draw boxplot for new_df

col = ['WT']

new_df.boxplot(col).

Assignment No - 3

Title: Descriptive Statistics - Measure of Central Tendancy and variability.

Problem Statement:

Perform the following operations on any open source dataset.

1) Provide summary statistics (mean, median, maximum, minimum, standard deviation) for a dataset (age, income, etc) with numeric variables grouped by one of the qualitative variables. For example, if your categorical variable is age groups and quantitative is income, then provide summary statistics of income grouped by age groups. Create a list that contains a numeric value for each response to categorical variable.

2) Write a python program to display some basic statistical details like percentile, mean, standard deviation of the species like 'iris-setosa', 'iris-versicolor', 'iris-Verginica' of iris.csv dataset.

Objective of Assignment:

Student should be able to perform the Statistical operations using Python on any open source dataset.

Prerequisites:

(1) Basics of Python

(2) Concept of statistics such as mean, median, minimum maximum, standard deviation.

Concepts / Theory :

1] Summary Statistics

Statistics :
It is the science of collecting data and analysing them to infer proportions that represent population.

Branches of Statistics :
(i) Descriptive Statistics
(ii) Inferential Statistics

(i) Descriptive Statistics
It is summarising the data at hand through certain numbers like mean, median, etc so as to make the understanding of the data easier. It does not involve any generalisation or inference beyond what is available.

Commonly Used Measures
(1) Measures of Central Tendancy
(2) Measure of Dispersion (or Variability)

(1) Measure of Central Tendancy
This is a one number summary of the data that typically describes the centre of the data. Central tendancy is measured in :

a) Mean:

Mean is defined as the ratio of the sum of all observations in the data to the total number of observations.

$\therefore$ Mean $= \bar{x} =$ Sum of all observations / Total number of observations

Python function used: df.mean()

b) Median:

Median is the point which divides the entire data into two equal halves. It is calculated by arranging the data in either ascending or descending order, then the middle observation is the median (if number of obs is odd) ~~or~~ else it is the mean of two middle obs (if number of obs is even) in sorted form

Python function used: df.median()

c) Mode:

Mode is the number which has maximum frequency in the entire data set. A data set can have more than one modes.

Python function used: df.mode()

(2) Measure of Dispersion (Variability)

It describes the spread of data around central value.

(a) Absolute Deviation from Mean

It describes variation in the data set, ie, average absolute distance of each data point.

$$\text{Mean Absolute Deviation} = \frac{1}{N}\sum_{i=1}^{N}|X_i - \bar{X}|$$

(b) Variance

Variance measures how far are data points spread out from the mean.

$$\text{Variance} = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2$$

(c) Standard Deviation

Square root of variance is called standard deviation.

$$\text{Std. Deviation} = \sqrt{\text{Variance}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

(d) Range

Range is difference between maximum value and minimum value in the dataset.

Range = Maximum - Minimum

(e) Quartile

They are the points in the dataset that divide the data set into four equal parts. Q_1, Q_2, Q_3 are first, second, third quartile of data set.

(i) 25% of data points lie below Q1 and 75% above it.

(ii) 50% of data points lie below Q2 and 50% above it.

(iii) 75% of data points lie below Q3 and 25% above it.

(f) Skewness

Measure of asymmetry in a problem of probability distribution is defined by skewness.

$$\text{Skewness} = \frac{3(\text{Mean} - \text{Median})}{\text{Std Deviation}}$$

(i) Positive Skew - indicates mean is greater than mode

(ii) Negative Skew - indicates mean is smaller than mode.

2] Types of Variables

A variable is a characteristic that can be measured and can have different values.

Variables are classified as:

(a) Categorical

(b) Numeric

(a) Categorical variables

A categorical variable is a variable that cannot be quantified. It refers to a characteristic. It is further classified as:

(i) Nominal value

A nominal variable is one that describes a name, label or category without natural order

(ii) Ordinal value

An ordinal variable is a variable whose values are defined by an order relation between the different categories.

(b) Numerical Variables

A numeric variable is a quantifiable characteristic whose values are numbers. Types of numeric variables are

(i) Continuous variables

A variable is said to be continuous if it can assume an infinite number of real values within a given interval.

(ii) Discrete variables

A discrete variable can assume only finite number of real values within a given interval.

Algorithm:

To display basic statistical details on Iris dataset.

1) Import all the required Python libraries.

2) Load the iris dataset into pandas dataframe
iris = pd.csv("/content/Iris.csv)
iris

3) Load all rows with Iris-setosa species in variable irisSet.
irisSet = (iris['Species'] == 'Iris-setosa')

4) To display basic statistical details like percentile, mean, standard deviation, etc for Iris setosa use describe
print('Iris-setosa')
print(iris[irisSet].describe())

5) Load all rows with Iris-versicolor species in variable irisVer
irisVer = (iris['Species'] == 'Iris-versicolor')

6) To display basic statistical details
print('Iris-setosa')
print(iris[irisSet].describe())

7) Load all rows with Iris-Virginica species in variable irisVir

```
irisVir = (iris['Species'] == 'Iris-virginica')
```

8) To display basic statistical details for Iris virginca use describe.

```
print ('Iris-virginica')
print (iris[irisVir].describe())
```

Conclusion:

Descriptive statistics summarises or describe the characteristics of a dataset. Descriptive statistics consists of two categories:

(i) Measure of central tendency describe the centre of a data set. It includes mean, median and mode

(ii) Measure of variability describe the dispersion of data within the set. It includes standard deviation, variance, minimum and maximum variables.

Assignment Questions:

Q1) Explain Measures of Central Tendancy with examples.

Ans: Measure of Central Tendency is a one number summary of the data that typically describes the centre of the data.

Types:

(a) Mean:

It is defined as ratio of sum of all observations to total number of observation.

Ex - Consider following data points : 17, 16, 21, 18, 15, 17, 21, 19, 11, 23

$\therefore$ Mean $= \frac{17+16+21+18+15+17+21+19+11+23}{10} = 17.8$

(b) Median:

It is the point that divides the entire data into two equal parts.

Ex - Consider following data pts: 17, 16, 21, 18, 15, 17, 21, 19, 11, 23

$\therefore$ Median $= \frac{17+18}{2} = 17.5$

(c) Mode:

Mode is the observation which has maximum frequency of occurance in the dataset.

Ex - Consider data pts : 17, 16, 21, 18, 15, 17, 21, 19, 11, 23

Here 17 and 21 occur twice.

$\therefore$ Data is bimodal with 17 and 21 as modes.

Q2) What are different types of variables. Explain with example.

Ans: A characteristic that can be measured and can assume different values is called a variable. Types of variables are:

(i) Categorical or Qualitative variable:

(a) Nominal variable

It describes a name, label or category without natural order.

Ex: In the given table 'Students in a class' is a nominal variable.

Students in a class	Total
Male	28
Female	32

(b) Ordinal variable

Values of ordinal variable are defined by an order relation

Ex: In the given table, variable 'Winners' is ordinal variable as there exists an orderly relation 'rank'.

Winners	Rank
Amey K	First
Tanvi P	Second
Karan D	Third

(ii) Numeric or Quantitative variable:

(a) Continuous variable

If a variable assumes infinite number of real values within a given interval it is continuous variable.

Ex: For instance, consider height of a student. The height can't be negative nor it can be more three metres. But between 0 to 3, the number of possible values is infinite. Hence, it is a continuous variable.

(b) Discrete variable.

If a variable assumes only a finite number of real values within a given interval

Ex: The score given by a judge in a competition is between 0 to 10. The score is always given to one decimal. Hence, it can be a discrete variable.

Assignment No. 4

Title : Data Analytics I

Problem Statement :

Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset. The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 features variables in this dataset. The objective is to predict the value of prices of house using given features.

Objectives of Assignment :

Student should be able to do data analysis using linear regression using Python for any open source dataset.

Prerequisites :

(1) Basics of Python programming
(2) Concepts of Regression.

Concepts for Theory :

(1) Linear Regression : Univariate and Multivariate
(2) Least Square Method for Linear Regression.
(3) Measuring Performance of Linear Regression
(4) Example of Linear Regression
(5) Training dataset and testing dataset

Theory:

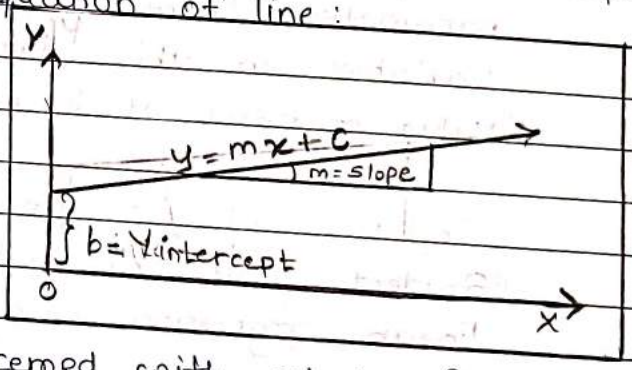
1] Linear Regression:

It is a machine learning algorithm based on supervised learning. It targets prediction values on the basis of independent variables. A linear relationship between a dependent variable is continuous while independent variable may be continuous or discrete. A linear regression model is popular because the cost function is Mean Squared Error. It is shown as equation of line:

$Y = m(x) + c$

where, m = slope

c = intercept



Univariate Regression is concemed with study of single predictor variable.

Multivariate Regression is concemed with study of two or more predictor variable.

Regression line equation is given by,

$y = \beta_0 + \beta_1 x$

where y = dependent variable.

x = independent variable.

β_0 = y intercept

β_1 = slope of line.

2] Least Square Method :

This method is used to guess the values of the parameters based on sample set. This technique estimates parameters β_0 and β_1 and tries to minimise the square of errors at all the points in sample set. The error is the deviation of actual sample data point from the regression line.

3] Measuring performance of Linear Regression :

(1) Mean Square Error (MSE) :

MSE represents the error of the estimator or predictive model created based on the given set of observations in the sample. The lesser is the MSE value, better is the regression model. MSE can be calculated as:

$$MSE = \frac{1}{n} \sum (y - \bar{y})^2$$

where y = actual value

$\bar{y}$ = predicted value.

An MSE of zero represents the fact that the predictor is a perfect predictor.

(2) Root Mean Squared Error (RMSE) :

It calculates the least square error and take a root of summed values. Mathematically, it is the square root of the sum of all errors divided by the total number of

values.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{1}{n}(y_i - \hat{y}_i)^2}$$

(3) R-squared

R-squared is the ratio of the sum of squares regression (SSR) and sum of squares total (SST). A value of R-squared closer to 1 would mean that regression model covers most part of the variance of the values of the response variable and can be termed as a good model. It is given as:

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y} - \bar{y})^2}{\sum(y - \bar{y})^2}$$

4] Training dataset and Testing dataset

Training dataset: It is a dataset having attributes and class labels and used for training machine learning algorithms to prepare models.

Testing dataset: It is a dataset for which class label is unknown. It is tested using built model.

In ML, the fit method is called on training set to build a model. Further, the fit model is applied on testing set to estimate the target value and evaluate

the model's performance.

Algorithm:

Step 1: Import libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Step 2: Import the Boston Housing Dataset.

```
from google.colab import files
files.upload(boston housing.csv)
```

Step 3: Initialize the dataframe

```
df = pd.read_csv("/boston_housing.csv")
df
```

Step 4: Perform Data Processing

```
df.isna().sum()
```

Step 5: Split dependent and independent variables

```
x = df.drop(['medv'], axis = 1)
y = df['medv']
```

Step 6: Splitting data to training and testing dataset.

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0)
```

Step 6: Use linear regression to create model.

```
import sklearn
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
model = lm.fit(xtrain, ytrain)
```

Step 7: Predict the ypred for all values of xtrain, xtest

```
ytrain_pred = lm.predict(xtrain)
ytest_pred = lm.predict(xtest)
ytrain_pred.
```

Step 8: Calculate MSE, RMSE and R2score for ytrain.

```
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = (np.sqrt(mean_squared_error(ytrain, ytrain_pred)))
r2 = r2_score(ytrain, ytrain_pred)
```

Step 9: Calculate MSE, RMSE and R2score for ytest

```
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = (np.sqrt(mean_squared_error(ytest, ytest_pred)))
r2 = r2_score(ytest, ytest_pred)
```

Step 10: Plotting the linear regression model.

```
plt.scatter(ytrain, ytrain_pred, c='blue', label='Training data')
plt.scatter(ytest, ytest_pred, c='green', label='Test data')
plt.xlabel('True values')
plt.ylabel('Predicted values')
plt.legend(loc='upperleft')
plt.plot()
plt.show()
```

4.4

Conclusion:

In this way we have done data analysis using linear regression for Boston Housing Dataset and predicted the price of house using its features.

Assignment Questions

Q1) Explain SST, SSE, SSR, MSE, RMSE, R2 Score

Ans: 1) SST: Total sum of squares
It is the squared difference between the observed dependent variable and its mean.

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

2) SSE: Sum of squares error
It is the squared difference between observed value and predicted value.

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

3) SSR: Regression sum of squares
It is the squared difference between the predicted value and the mean of the dependent variable

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

4) MSE: Mean Squared Error
It is the average sum of squared difference between the actual and predicted value.

$$MSE = \frac{1}{n}\sum (y - \hat{y})^2$$

5) RMSE: Root Mean Squared Error
It is the square root of the sum of all errors divided by the total number of values.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{1}{n} (\hat{y}_i - y_i)^2}$$

6) R-Squared Score:
It is the ratio of sum of squares regression (SSR) and sum of squares total. (SST).

$$R^2 = \frac{SSR}{SST} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

Q2) Write python code to calculate the RS quare for Boston Dataset.

Ans: Step1 : Import libraries.

```
import numpy as np
import pandas as pd
import ~~matplot~~ sklearn
```

Step 2: Load the dataset.

```
from sklearn.datasets import load_boston
boston = load_boston()
data = pd.DataFrame(boston.data)
```

Step 3: Split dependent and independent variable.

```
x = data.drop(['PRICE'], axis=1)
y = data.['PRICE']
```

Step4: Spliting data as training and testing dataset.

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=
0.2, random_state=0)
```

Step5: Creating model

```
import sklearn
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
model = lm.fit(xtrain, ytrain)
```

Step6: Predict the y_pred.

```
ytrain_pred = lm.predict(xtrain)
ytest_pred = lm.predict(xtest)
```

Step7: Evaluate R-Square

```
from sklearn.metrics import r2_score
r2_score(y, ytrain_pred)
r2_score(y, ytest_pred)
```

Assignment No 5

Title : Data Analytics II

Problem Statement:

(1) Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
(2) Compute confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Objective of the assignment:
Student should be able to data analysis using logistic regression using Python for any open source dataset.

Prerequisites:
1) Basic of Python programming
2) Concept of Regression.

Concepts for Theory:
1) Logistic Regression
2) Sigmoid function
3) Types of logistic regression
4) Confusion Matrix Evaluation Metrics.

(1) Logistic Regression

Logistic regression is one of the most simple and commonly used Machine learning algorithm for two class classification. It is a statistical method for predicting binary classes. The outcomes or target variable is dichotomous in nature. It computes the probability of event occuring. It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. It uses a logit function.

Equation of linear regression: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots \beta_n x_n$
where, y is a dependent variable and x_n are exploratory variables.

Sigmoid function: $p = 1/1 + e^{-y}$

Apply sigmoid function on linear equation:

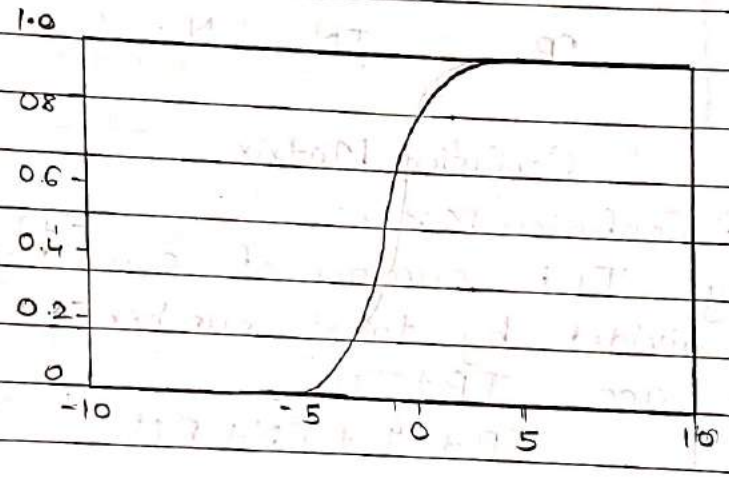
$\therefore p = 1/1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_n x_n)}$

(2) Sigmoid Function

The sigmoid function also called logistic function, gives an 'S' shaped curve that can take any

real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1 and if the curve goes to negative infinity, y predicted will become 0.

Equation : $f(x) = \frac{1}{1+e^{-x}}$



(3) Types of Logistic Regression

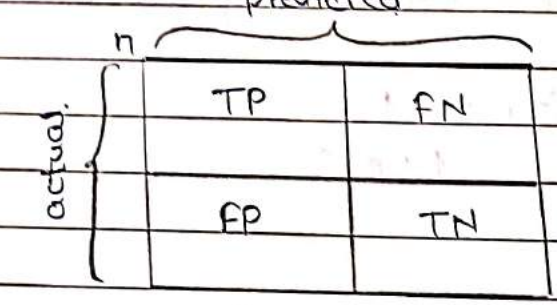
(i) Binary Logistic Regression:
The target variable has only two possible outcomes such as Cancer or No Cancer.

(ii) Multinomial Logistic Regression:
The target variable has three or more nominal.

(iii) Ordinal Logistic Regression:
The target variable has three or more ordinal categories

(4) Confusion Matrix:

It contains information about actual and predicted classifications done by a ~~sth~~ system.



Confusion Matrix

Measures of Confusion Matrix:

(i) Accuracy: It is number of correctly classified instances divided by total number of instances.

$$\text{Accuracy} = acc = \frac{TP+TN}{TP+FP+TN+FN}$$

(ii) Error Rate: It is number of incorrectly classified instances divided by total number of instances.

$$\text{Error Rate} = err = \frac{FP+FN}{TP+FP+TN+FN} = 1 - acc.$$

(iii) Precision: It is number of correctly classified positive instances divided by total number of instances which are predicted positive.

$$\text{precision} = \frac{TP}{TP+FP}$$

Algorithm:

Step 1: Import libraries and create alias for Pandas, Numpy and Matplotlib.

Step 2: Import the Social_Media_Adv Dataset.

Step 3: Initialize the dataframe.

Step 4: Perform Data Processing.
(1) Convert categorical to numeric values.
(2) Check for null values
(3) Divide the dataset into independent (x) and dependent (y). variable.
(4) Split the dataset into training and testing dataset.

Step 5: Use Logistic Regression to create model.
from sklearn.linear_model import LogisticRegression.
logreg = LogisticRegression()
logreg.fit (xtrain, ytrain)
y_pred = logreg.predict (xtest)

Step 6: Predict the y_pred for all values of train_x and ~~train_y~~ test_x.

Step 7: Evaluate the performance of model for train_y and test_y.

Step 8: Calculate the required evaluation parameters.

```
from sklearn.metrics import
precision_score, confusion_matrix, accuracy_score, recall_score
cm = confusion_matrix(y_test, y_pred)
```

Conclusion:

In this way we have done data analysis using logistic regression for Social Media Adv. and evaluated the performance of model.

Assignment Questions:

Consider binary classification. Find out TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall.

N = 165	Predicted Yes	Predicted No
Actual YES	TP = 150	FN = 10
Actual No	FP = 20	TN = 100

∴ True Positive = TP = 150
∴ False Positive = FP = 20
∴ True Negative = TN = 100
∴ False Negative = FN = 10

$$\therefore \text{Accuracy} = acc = \frac{TP+TN}{TP+FP+TN+FN} = \frac{150+100}{150+20+100+10}$$

∴ Accuracy = 0.8928

$$\therefore \text{Error rate} = err = 1 - \text{accuracy} = 1 - 0.8928$$

∴ Error rate = 0.1074

$$\therefore \text{Precision} = \frac{TP}{TP+FP} = \frac{150}{150+120} = 0.8823$$

$$\therefore \text{Recall} = \frac{TP}{TP+FN} = \frac{150}{150+10} = 0.9375.$$

Q2) Comment on whether the model is best fit or not based on calculated values.

Ans: For the above classification,
Accuracy = 0.8928
Error rate = 0.1074
Precision = 0.8823
Recall = 0.9375

Therefore, the given classification ~~is~~ model is best fit based on the calculated values.

Q3) Difference between linear regression and logistic regression.

Linear Regression	Logistic Regression
1) It is a supervised regression model	1) It is a supervised classification model.
2) It is based on least square estimation	2) It is based on maximum likelihood estimation
3) Equation of linear regression: $y = a_0 + a_1x_1 + a_2x_2 \dots + a_nx_n$	3) Equation of logistic regression $y(x) = e^{(a_0 + a_1x_1 + \dots a_nx_n)}/(1 + e^{(a_0 + a_1x_1 + a_2x_2 \dots + a_nx_n)})$.
4) In linear regression, no threshold value is needed.	4) In logistic regression threshold value is needed.

	Linear Regression	Logistic Regression.
(5)	Here, when we plot the training datasets, a straight line can be drawn that touches maximum plots.	(5) Here, if we plot the training dataset, for positive slope an S-shaped (sigmoid) curve is plotted and for negative slope Z-shaped curve is plotted.
(6)	Linear Regression assumes the normal or gaussian distribution of the dependent variable.	(6) Logistic regression assumes binomial distribution of the dependent variable.

Assignment No 6

Title: Data Analytics III

Problem Statement:

(1) Implement Simple Naive Bayes Classification algorithm using Python/R on iris.csv dataset.
(2) Compute confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Objective of assignment:

Student should be able to data analysis using Naive Bayes algorithm using Python for any open source dataset.

Prerequisites:
1) Basic of Python program.
2) Concept of Naive Bayes Classification.

Theory:

1] Concepts used in Naive Bayes Classifier.

(1) It can be used for Classification of categorical data. The Naive Bayes classifier depends on Baye's

rule from probability theory.

Prior probabilities: Probabilities which are calculated for some event based on no other information are called prior probabilities.

Conditional Probabilities:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad \text{if } P(B) \neq 0 \quad -(1)$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad -(2)$$

From (1) & (2)

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

where,

P(A|B) is posterior probability.
P(B|A) is likelihood probability
P(A) is prior probability
P(B) is marginal probability.

Types of Naive Bayes Model.

(1) Gaussian: It assumes that features follow a normal distribution.
(2) Multinomial: It is used when the data is multinomial distributed.
(3) Bernoulli: It is similar to multinomial but predictor variables are independent variables.

Algorithm:

Step 1 : Import libraries and create alias for Pandas, Numpy, Matplotlib

Step 2 : Import the Iris dataset and load it into a dataframe.

Step 3 : Perform Data Processing
(1) Convert categorical to numerical values.
(2) Check for null values
(3) Divide the dataset into Independent (x) and dependent (y) variable.
(4) Split the dataset into training and testing.

Step 4 : Use Naive Bayes algorithm to create model

```
#import class
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit (xtrain, ytrain)
```

Step 5 : Predict y_pred for all values of train_x and test_x.

```
y_pred = gaussian.predict (xtest)
```

Step 6 : Evaluate the performance of model for train_y test_y.

```
accuracy = accuracy_score (y_test, y_pred)
```

```
precision = precision_score (y_test, y_pred, average = 'micro')
recall = recall_score (y_test, y_pred, average = 'micro')
```

Step 7 : Calculate the required evaluation parameters.

```
from sklearn.metrics import
precision_score, confusion matrix, accuracy_score, recall_score
cm = confusion_matrix (y_test, y_pred)
```

Conclusion :

In this way we have done data analysis using Naive Bayes Algorithm for Iris dataset and evaluated the performance model.

FOR EDUCATIONAL USE

Assignment Questions:

Q1) Write python code for the preprocessing mentioned in step 4. Explain every step in detail.

Ans: After importing the ~~data~~ Iris dataset and initializing it to the dataframe, the next step is data preprocessing.
Data Preprocessing involves following steps:

(i) Convert Categorical to ~~Null values~~ Numeric Values.

```
from sklearn import preprocessing
df['Species'].unique()
label_encoder = preprocessing.LabelEncoder()
df['Species'] = label_encoder.fit_transform(df['Species'])
df['Species'].unique()
```

In the above step the categorical variable 'Species' in iris dataset is converted into numeric values.

(ii) Check for Null Values

```
df.isna().any().sum()
```

The above step gives statistics for null values if any in the dataset. If null values are present removal is must.

(iii) Divide the dataset into Independent (x) and Dependent (Y) variables.

```
~~data~~ df.columns = iris.feature_names
x = df.columns.values()
y = iris.data()
```

(iv) Split the dataset into training and testing dataset

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size = 0.2, random_state = 0)
```

Assignment No. 7

Title: Text Analytics

Problem Statement:

(1) Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, Stop words removal, Stemming and Lemmatization.
(2) Create representation of document by calculating Term Frequency and Inverse Document Frequency.

Objective :

Student should be able to perform Text Analysis using TF IDF Algorithm.

Prerequisite :

(1) Basics of Python Programming
(2) Basics of English Language.

Theory:

1] Text Analysis and Operations

(1) Text Analysis:

It is the process of exploring sizable textual data and finding patterns. In Text Analytics, statistical and machine learning algorithms are used to classify information.

Operation in Text Analysis:

(1) Tokenization:
The process of breaking down a text paragraph into smaller chunks such as words or sentences is called Tokenization. Token is a single entity that is the building blocks for a sentences or paragraph.
(i) Sentence tokenization: Split a paragraph into list of sentences using sent_tokenize() method.
(ii) Word tokenization: Split a sentence into list of words using word_tokenize() method.

(2) Stop words removal
Stopwords considered as noise in the text. Text may contain stop words such as is, am, are, this, a, an, etc. In NLTK for removing stopwords, you need to create a list of stopwords and filter out your list of stopwords and filter out your list of tokens from these words.

(3) Stemming:
Stemming is a normalization technique where lists of tokenized words are converted into shortened root words to remove redundancy. A computer program that stems word may be called a stemmer.

Ex: A stemmer reduces words like fishing, fished, fisher to the stem fish.

mmatization:
mmatization in NLTK is the algorithmic process of ding the lemma of a word depending on its aning and context. It usually refers to the orphological analysis of words, which aims to remove fectional endings.
g. lemma for studies is study.

) POS Tagging:
OS (Part of Speech) tell us about grammatical nformation of words of the sentences by assigning pecific token as tag to each words. Words can have nore than one POS depending upon the context where is used.

2] Text Analysis Model using TF-IDF

Term Frequency- Inverse Document Frequency (TFIDF). s a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

(1) Term Frequency (TF):
TF is defined as the ratio of a words occurances in a document to the total number of words in a document.

$$TF(w,d) = \frac{\text{occurances of } w \text{ in document } d}{\text{total no. of words in document } d}$$

(2) Inverse Document Frequency (IDF)
IDF is the measure of importance of a word. It provides weightage to each word based on its frequency in the corpus D.

$$IDF(w, D) = \ln \left[\frac{\text{Total no. of documents (N) in corpus D}}{\text{number of documents containing w.}} \right]$$

(3) Term frequency - Inverse Document frequency (TFIDF)
It is the product of TF and IDF. It gives more weightage to the word that is more in the corpus

$$TFIDF(w, d, D) = Tf(w, d) * IDF(w, D).$$

3] Bag of Words (BoW)
Bag of Words is a technique for extracting features from text is to place all of the words that occur in the text in a bucket.

Algorithm:
(1) Tokenization, POS Tagging, Stop words removal, stemming and lemmatization.

Step 1: Download the packages.
nltk.download('punkt')

```
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_tagger')
```

Step 2 : Initialize the text.

Step 3. Perform Tokenization

```
from nltk.tokenize import sent_tokenize
tokenized_word = sent_tokenize(text)
print(tokenized_word)
```

Step 4 : Removing Punctuations and stop words

Step 5 : Perform Stemming

```
from nltk.stem import PorterStemmer
e_words = ["wait", "waiting", "waited"]
ps = PorterStemmer()
for w in e_words:
    rootWord = ps.stem(w)
print(rootWord)
```

Step 6 : Perform Lemmatization.

Step 7 : Apply POS Tagging to text.

(2) Algorithm for creating representation of document by calculating TFIDF.

Step 1: Import necessary libraries.
import pandas apd
from sklearn.feature_extraction.text import TfidfVectorizer.

Step 2: Initialize the document.

Step 3: Create Bag of Words (BoW) for document A and B.

Step 4: Create collection of unique words from document A and B.

Step 5: Create a dictionary of words and their occurances for each document in the corpus.

Step 6: Compute the term frequency for each of our documents.

Step 7: Compute the term Inverse Document frequency.

Step 8: Compute the term TF-IDF for all words

Conclusion:
In this way we have done text data analysis using TFIDF algorithm.

Assignment Question

Q1) Perform Stemming for text = "studies studying cries cry". Compare result generated with Lemmatization Comment on how they differ from each other.

Ans:

text = "studies studying cries cry".

Results of Stemming :
Root word for "studies", "studying" is "study".
Root word for "cries", "cry" is "cry".

Result of Lemmatization :
Lemma for "studies" is "study"
Lemma for "studying" is "studying"
Lemma for "cries" is "cry"
Lemma for "cry" is "cry".

Q2) Calculate TF, IDF, TFIDF for each document.

ns:

Document A = 'Jupiter is the largest planet'

Document B = 'Mars is the fourth planet from the Sun'.

Document	Text	Total number of words
A	Jupiter is the largest planet.	5
B	Mars is the fourth planet from the sun	8

Words	TF (for A)	TF (for B)	IDF.
Jupiter	1/5	0	$\ln(2/1) = 0.69$
Is	1/5	1/8	$\ln(2/2) = 0$
The	1/5	2/8	$\ln(2/2) = 0$
Largest	1/5	0	$\ln(2/1) = 0.69$
Planet	1/5	1/8	$\ln(0) = 0$
Mars	0	1/8	$\ln(2/1) = 0.69$
Fourth	0	1/8	$\ln(2/1) = 0.69$
From	0	1/8	$\ln(2/1) = 0.69$
Sun	0	1/8	$\ln(2/1) = 0.69$

Words	TFIDF(A)	TFIDF(B)
Jupiter	0.138	0
Is	0	0
The	0	0
~~Planet~~ Largest	0.138	0
~~Mars~~ Planet	0.138	0
~~fourth~~ Mars	0	0.86
~~From~~ fourth	0	0.86
From	0	0.86
Sun	0	0.86

Q3) Differentiate between Stemming and Lemmatization

Ans: 1) Stemming:

(1) Stemming is a process that stems or removes the suffix from a word.

(2) Stemming is a general operation.

(3) Ex: A stemmer reduces the words like "wait", "waiting", to the root word wait.

2) Lemmatization:

(1) Lemmatization is a process that converts the words to its meaningful base form, called lemma.

(2) It is an intelligent operation.

(3) Ex: After lemmatizing the word 'Caring', it would return 'Care'

(4) Lemmatization helps in forming better machine learning features.

Assignment No. 8

Title : Data Visualization I

Problem Statement :
1) Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate titanic ship. Use the seaborn library to see if we can find any patterns in the data.

2) Write a code to check how the price of the ticket (coloumn name : 'fare') for each passenger is distributed by plotting a histogram.

Objective of the assignment:
Students should be able to perform the data visualization using python on any open source dataset.

Prerequisite:
1) Basics of Python programming
2) Seaborn Library, concept of data visualization.

Theory:

1] Seaborn Library
The Seaborn library is built on top of Matplotlib and offers many advanced data visualization functions. It can be used to draw a variety of.

charts such as matrix plots, grid plots, etc and also to draw distributional and categorial plots.

2] Titanic Dataset.

The Titanic dataset is included in the Seaborn library. It contains 891 rows and 15 columns and contains information about the passengers who boarded the unfortunate titanic ship. The original task is to predict whether the passenger survived or not depending on various features such as age, ticket, cabin, class.

a) Distributional Plots

(1) Dist Plot

The distplot() shows the histogram distribution of data for a single column. The column name is passed as a parameter to the distplot() function. To check how the price of the ticket for each passenger is distributed, execute the following:

```
sns.distplot(dataset['fare'])
```

(2) Joint Plot

The jointplot() is used to display the mutual distribution of each column. The first parameter passed is the column name which display the

distribution on x-axis. The second parameter is the column name which display the distribution of data on y-axis and third parameter is the name of the dataframe. Plot a joint plot of age and fare columns to see if there is any relationship between the two.

sns.jointplot(x='age', y='fare', data=dataset)

From the output, a joint plot has three parts, a distribution plot at the top for the column on x-axis, a distribution plot on right for the column on y-axis and a scatter plot in between that shows the mutual distribution of data for both columns.

(3) Pair Plot:

The pairplot() is a type of distribution plot that basically plots a joint plot for all the possible combination of numeric and boolean columns in dataset. The name of your dataset need to pass as the parameter to the pairplot() function.

sns.paiplot(dataset)

To add information from categorical column to the pair plot, name of that column is passed as parameter to the 'hue' parameter).

sns.pairplot(dataset, hue='sex')

Algorithm:

Step 1: Download the data set of Titanic

Step 2: Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Step 3: Initializing the dataframe

```
dataset = sns.load_dataset ('titanic')
dataset.head()
```

Step 4: Draw distribution plot.

```
sns.distplot (dataset ['fare'])
```

Step 5: Removal of kernal Density Line.

```
sns.distplot (dataset ['fare'], kde = False)
```

Step 6: Draw histogram

```
sns.distplot (dataset ['fare'], kde = False, bins = 10)
```

Step 7: Draw Joint Plot

```
sns.jointplot (x = 'age', y = 'fare', data = dataset)
sns.jointplot (x = 'age', y = 'fare', data = dataset, kind = 'hex')
```

Step 8: Draw Pair Plot.

```
sns.pairplot (dataset)
```

Conclusion:

Seaborn is an advanced data visualization library built on top of Matplotlib library. In this assignment, we have explored distributional and categorical plots using Seaborn library.

Assignment Questions:

Q1) List out different types of plots to find patterns of data.

Ans: 1) Different types of plots to find patterns of data are:

(a) Distribution Plot: The distplot() shows the histogram distribution of data for a single column.

(b) Joint Plot: The jointplot() is used to display the mutual distribution of each columns.

(c) Pair Plot: The pairplot() is a type of distribution plot that basically plots a joint plot for all the possible combination of numeric and boolean columns in dataset.

Q2) Explain when you will use distribution plot and when you will use categorical plots.

Ans: 1) Distributional plots, as the name suggests are type of plots that show the statistical distribution of data.

2) Categorical plot, as the name suggests are used to plot categorical data.

3) The categorical plots plot the values in the

column against another categorical column or a numeric column.

4) On the contrary, the distribution plot take only qualitative type of data to plot the distribution plot.

Q3) Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

Ans:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
dataset = sns.load_dataset('titanic')
dataset.head()
# To check how the price of ticket is distributed :
sns.distplot(dataset['fare'])
```

Q4) Which parameter is used to add another categorical variable to violin plot, Explain the syntax with example.

Ans: To add another categorical variable to the violin plot, 'hue' parameter is used.

The 'hue' parameter takes value as name of the another categorical variable.

ans. violinplot(x='sex', y='age', data=dataset, hue='survived')

Assignment No. 9

Title : Data Visualization II

Problem Statement :
1) Use the inbuilt dataset 'Titanic'. Plot a boxplot for distribution of age with respect to each gender along with the information about whether they survived or not.

2) Write observations on the inference from above statistics

Objectives of the assignment :

Student should be able to perform data visualization using Python on any open source dataset.

Prerequisites :
1) Basics of Python Programming
2) Seaborn Library, concept of Data Visualization.

Theory :

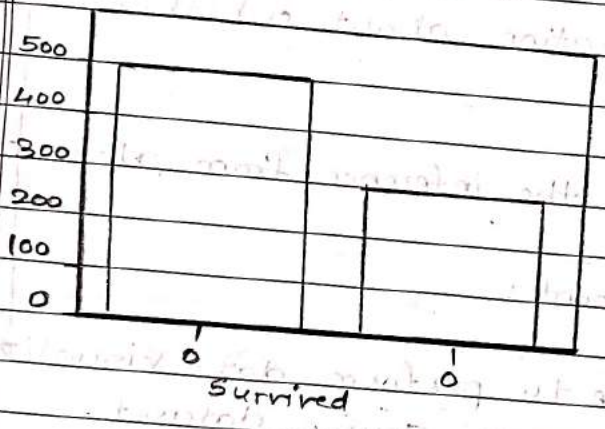
1] Categorical Data :

A variable that has text-based information is referred to as categorical variables. Following are various plots which we can use for visualizing categorical data.

(1) Count Plot:

The count plot plots the count of each category in a seperate bar. It is basically a count of frequency plot in form of a bar graph.

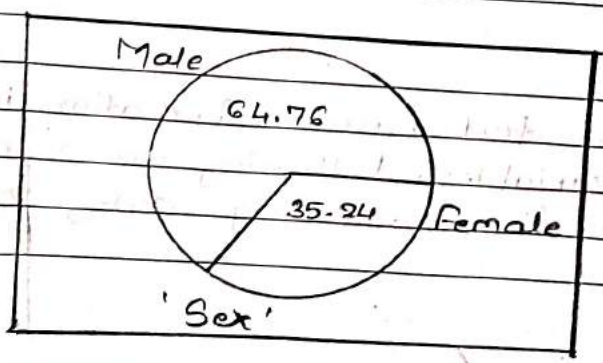
```
sns.countplot(data['Survived'])
plt.show()
```



(2) Pie Chart:

Pie Chart is count plot that gives additional information about the percentage presence of each category.

```
data['Sex'].value_counts().plot(kind="pie", autopct="%.2f")
plt.show()
```

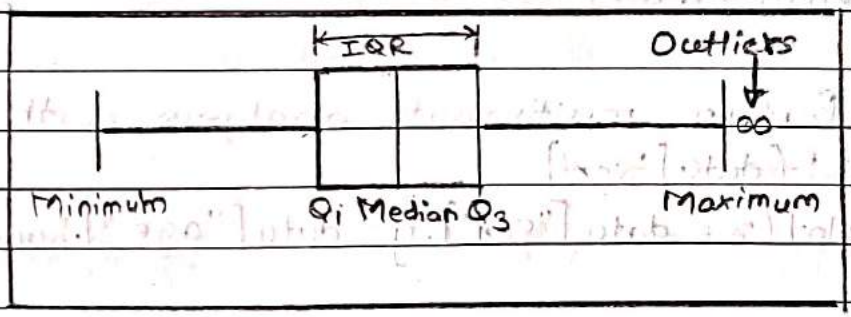


2] Numerical Data:

A variable that has number based information is numerical data. Analyzing Numerical data is important because understanding the distribution of variables helps to further process the data.

(1) Boxplot:

A boxplot is a five number summary plot. The five number summary includes minimum, first quartile [Q_1], median, third quartile [Q_3] and maximum. They are a standardized way of displaying the distribution of data based on five number summary.



Both multivariate analysis and univariate analysis are possible using boxplot.

For multivariate analysis with boxplot:
Along with x, y, we use additional parameter 'hue'.
sns.boxplot(data['Sex'], data["Age"], data["Survived"])

Algorithm:

Step 1: Import necessary libraries.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Step 2: Load the dataset and initialize a dataframe.

```
from seaborn import load_dataset
data = load_dataset("titanic")
data.
```

Step 3: Box Plot for titanic dataset.

```
sns.boxplot(data)
```

Step 4: Perform multivariate analysis with boxplot.

~~sns.boxplot(data['sex']~~

```
sns.boxplot(x = data["sex"], y = data["age"], hue = data["Survived"]
```

Conclusion:

In this way by using the builtin titanic dataset boxplot distribution is studied.

Assignment Questions:

Q1) What is Data Visualization?

Ans: 1) Data visualization is the process of representing complex data or information using visual elements such as charts, graphs and maps.

2) The goal of data visualization is to present data in a way that is easy to understand and to help viewers identify patterns, trends and outliers that may not be immediately apparent in raw data.

3) Data visualization can be used to explore and analyze data, communicate findings to others and support decision making processes.

4) Some common types of data visualization include bar charts, line graphs, heatmaps and so on.

Q2) How to calculate min, max, range and standard deviation?

Ans: 1) To calculate minimum, maximum and range of dataset:
(1) Sort the data from smallest to largest.
(2) The minimum value is smallest number in dataset.
(3) The maximum value is largest number in dataset.
(4) The range is the difference between maximum and minimum values.

Ex: Dataset = {3, 5, 2, 8, 6, 4, 1, 9}
Sorted_dataset = {1, 2, 3, 4, 5, 6, 7, 8, 9}

Therefore, the minimum value is 1, maximum value is 9 and range is $9-1=8$.

2) To calculate standard deviation:
(1) Calculate the mean of dataset
(2) Find sum of squared deviations.
(3) Divide the sum by number of values in dataset.
(4) Take squareroot of the result obtained is step (3) to get standard deviation.

$$S.D = \sqrt{\frac{\sum(x-\bar{x})^2}{N}}$$

Q3) How to create boxplot for each feature in dataset?

Ans:1) To create a boxplot for each feature in a dataset:

(1) Import all the required packages and library
(2) Load the dataset into dataframe
(3) Select a feature or column from the dataset to create a boxplot for
(4) Use the chosen library to create a boxplot for selected feature. It should show minimum, maximum, median and IQR of the values in selected feature.
(5) Repeat the above steps (3) & (4) for each feature in dataset.

Q4) How to create histogram ?

Ans: 1) To create a histogram for a dataset :

(1) Load the dataset into the dataframe.
(2) Choose appropriate library or package and import it. Here import 'matplotlib'.
(3) Select a feature from dataset to create a histogram for.
(4) Use the chosen library to create a histogram for selected feature. The histogram should show the distribution of values in selected feature.