****

**(B.TECH) Trimester-XI AY 2020-21**

**PPL Lab Assignment No. 01**

**Problem Statement:** To study Data Collection, Data Preparation, Data Handling and perform Exploratory Data Analysis.

**Name: Vishwa Mohan Singh**

**ERP No.: 1032170273**

**Panel: AML2**

**Date: 30th of November, 2020**

**Objectives:**

1. To learn python programming with different modules/libraries.
2. To understand the concept of exploratory data analysis.

**Theory:**

**Data Collection**

Collection of data is the most fundamental step in any data science process. There are many different ways of collecting data.

**1. Building dataset from scratch**

Advantages-

1. Features are included based on the purpose of the research question or task. Not vice versa.

2. This helps to only use meaningful data.

3. It is traceable how the variables were created.

Disadvantages:

1. It can be challenging to find suitable sources.

2. It takes a lot of time to gather the data.

3. Transforming features into the right format can be a lot of effort.

**2. Using government websites-**

Advantages-

1. Commonly high data quality. Data used by other researchers or practitioners.

2. Often data is well documented. Therefore, one can understand how the variables were created.

Disadvantages:

1. It can take a lot of time to gather and transform features.

2. It can be hard to understand the data sets.

3. Access is sometimes only given on a request basis.

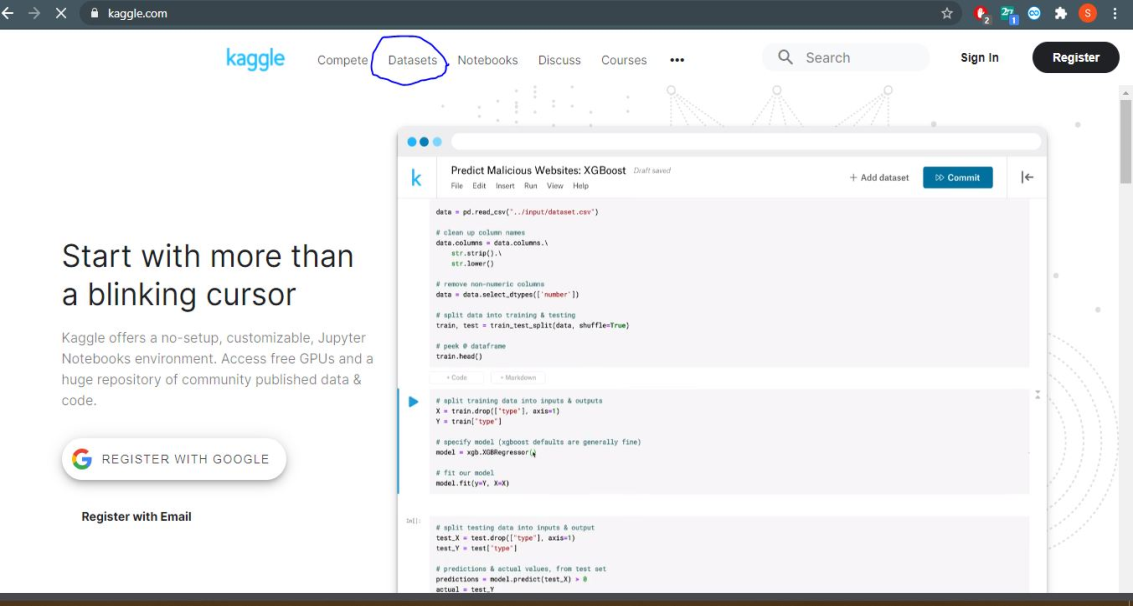
Example: https://catalog.data.gov/dataset

**3. Accessing private datasets-**

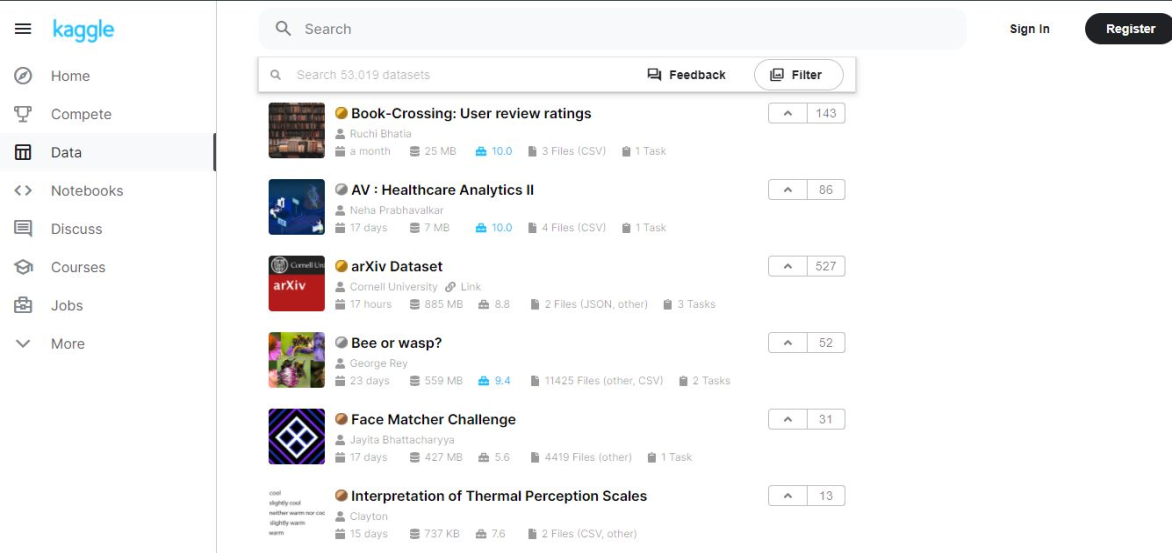
Need to substantiate quality of data before using it in any professional projects.

Example: https://www.kaggle.com/

Kaggle Homepage:



Data sets available to download:



**Various types of data**:

* **Numerical**

It represents quantitative measurement. Ex.: Height of a people, stock prices.

* **Discrete Data**

Integer based, often counts of something. Ex.: How many times did I toss “Heads”?

* **Continuous Data**

It has an infinite number of possible values. Ex.: How much rainfall on a given day?

* **Categorical Data**

Qualitative data, Ex.: Gender, Yes/No, etc. Assign some number to categorical data but

they don’t have any mathematical meaning

* **Ordinal Data**

Mixture of numerical and categorical data .Categorical data has mathematical meaning

For example: Movie rating on a scale of 1–5 .Rating must be 1,2,3,4,5

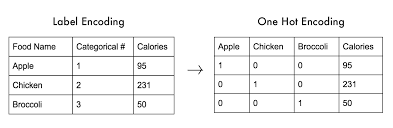
They have mathematical meaning. Like rating 1 movie is worse than rating 2 movie

**Label encoding:**

In label encoding, each category is mapped to a number or a label. The labels chosen for the categories have no relationship. So categories that have some ties or are close to each other lose such information after encoding. It supports the pandas dataframe as input and can transform data.

**One-Hot Encoding:**

A one hot encoding allows the representation of categorical data to be more expressive. Many Machine Learning algorithms cannot work with categorical data directly. The categories must be converted into numbers.



**Operations to be performed on dataset:**

**Steps in Preprocessing of Data**

1. Importing Python Modules/Libraries

2. Importing data

3. Displaying data

4. Creating the Independent and Dependent variables

5. Replacing missing value with meaningful value

6. Encoding categorical data

7. Splitting the data into training and test set

8. Doing feature scaling on data

9. Use any 3-4 graphs/plots

**Program code:**

import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler

from sklearn.compose import ColumnTransformer

from sklearn.model\_selection import train\_test\_split

import matplotlib.pyplot as plt

## Importing the data

dataset = pd.read\_csv("train.csv")

print(dataset.head())

print()

print(dataset.info())

## Array Transformation

arr = dataset.iloc[:, [2, 5, 6, 7, 9, 11, 1]]

## Handeling NaN

print("\n\nMissing values before imputing:\n", arr.isna().sum())

impu = SimpleImputer(missing\_values=np.nan, strategy="mean")

arr = arr.values

arr[:, [0, 1, 2, 3, 4]] = impu.fit\_transform(arr[:, [0, 1, 2, 3, 4]])

print("\n\nMissing values after imputing:\n", pd.DataFrame(arr).isna().sum())

# Remvoing the entries from categorical columns

arr = pd.DataFrame(arr)

arr = arr.dropna()

print("\n\nMissing values after removal:\n", pd.DataFrame(arr).isna().sum())

arr = arr.values

# OneHot and LableEncoding

lb = LabelEncoder()

arr[:, -1] = lb.fit\_transform(arr[:, -1])

print("\n\nAfter Label Encoding:\n", arr)

transformer = ColumnTransformer(

transformers=[("OneHot", OneHotEncoder(), [5])], remainder="passthrough"

)

arr = transformer.fit\_transform(arr.tolist())

print("\n\nAfter One Hot Encoding:\n", arr)

##X-Y Split

X = arr[:, :-1]

Y = arr[:, -1]

# Scaling

scaler = StandardScaler()

X = scaler.fit\_transform(X)

print("\n\nAfter Standard Scaling:\n", X)

# Train Test Split

arr\_train, arr\_test, Y\_train, Y\_test = train\_test\_split(arr, Y, test\_size=0.2)

# Plotting Graphs

plt.pie(

[len(Y == 1), len(Y == 0)],

labels=["Survived", "Did not survive"],

autopct="%1.1f%%",

)

plt.title("Class distribution")

plt.show()

dataset.groupby("Sex").Age.plot(kind="kde")

plt.title("Gender Wise Age distribution")

plt.legend()

plt.show()

plt.boxplot(arr[:, 4])

plt.title("Ship Fair BoxPlot")

plt.legend()

plt.show()

**Dataset used:**

**Titanic:** A Kaggle competition dataset that is used for a machine learning competition aimed at predicting who might have survived the sinking. It includes passenger information like class, embarkment port, fare, cabin etc.

**Link:** https://www.kaggle.com/c/titanic/data

**Output:**

PassengerId Survived Pclass ... Fare Cabin Embarked

0 1 0 3 ... 7.2500 NaN S

1 2 1 1 ... 71.2833 C85 C

2 3 1 3 ... 7.9250 NaN S

3 4 1 1 ... 53.1000 C123 S

4 5 0 3 ... 8.0500 NaN S

[5 rows x 12 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null object

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

None

Missing values before imputing:

Pclass 0

Age 177

SibSp 0

Parch 0

Fare 0

Embarked 2

Survived 0

dtype: int64

Missing values after imputing:

0 0

1 0

2 0

3 0

4 0

5 2

6 0

dtype: int64

Missing values after removal:

0 0

1 0

2 0

3 0

4 0

5 0

6 0

dtype: int64

After Label Encoding:

[[3.0 22.0 1.0 ... 7.25 'S' 0]

[1.0 38.0 1.0 ... 71.2833 'C' 1]

[3.0 26.0 0.0 ... 7.925 'S' 1]

...

[3.0 29.69911764705882 1.0 ... 23.45 'S' 0]

[1.0 26.0 0.0 ... 30.0 'C' 1]

[3.0 32.0 0.0 ... 7.75 'Q' 0]]

After One Hot Encoding:

[[0.0 0.0 1.0 ... 0.0 7.25 0]

[1.0 0.0 0.0 ... 0.0 71.2833 1]

[0.0 0.0 1.0 ... 0.0 7.925 1]

...

[0.0 0.0 1.0 ... 2.0 23.45 0]

[1.0 0.0 0.0 ... 0.0 30.0 1]

[0.0 1.0 0.0 ... 0.0 7.75 0]]

After Standard Scaling:

[[-0.48271079 -0.30794088 0.61679395 ... 0.43135024 -0.47432585

-0.50023975]

[ 2.07163382 -0.30794088 -1.62128697 ... 0.43135024 -0.47432585

0.78894661]

[-0.48271079 -0.30794088 0.61679395 ... -0.47519908 -0.47432585

-0.48664993]

...

[-0.48271079 -0.30794088 0.61679395 ... 0.43135024 2.00611934

-0.17408416]

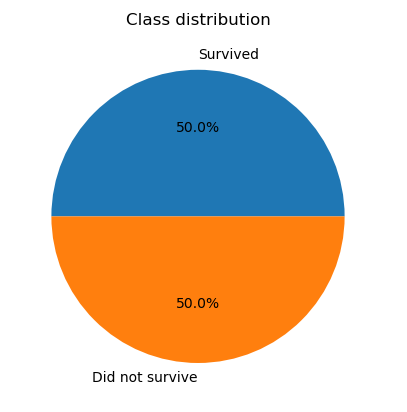
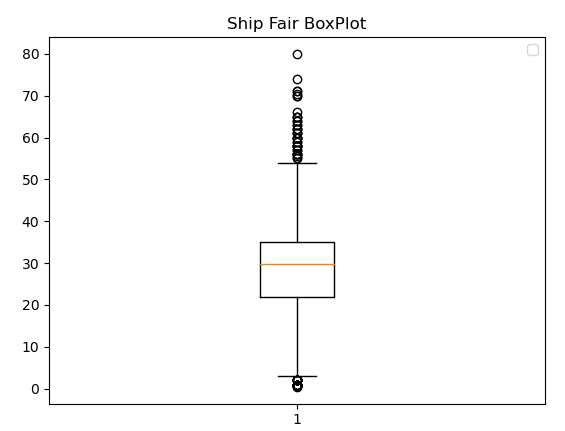
[ 2.07163382 -0.30794088 -1.62128697 ... -0.47519908 -0.47432585

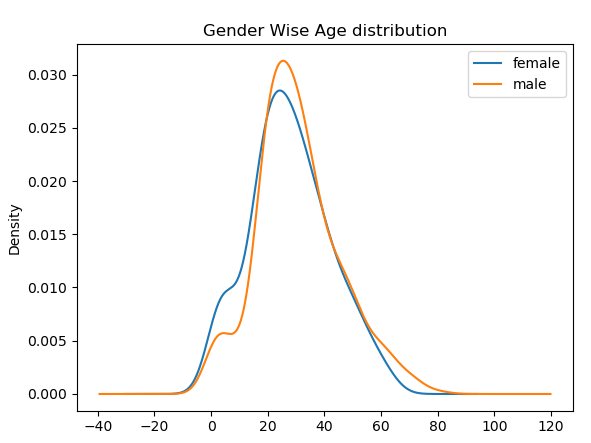
-0.0422126 ]

[-0.48271079 3.24737656 -1.62128697 ... -0.47519908 -0.47432585

-0.49017322]]

**Plots:**



**FAQs:**

1) List two common libraries for data manipulation. Give an example for each library.

=>

**Pandas**: This is the most commonly used libraries to import and manipulate data stored in different format. We can impute data, interpolate missing values for continuous variable, drop or fill missing values etc.

Example: dataset = pd.read\_csv("train.csv")

dataset.dropna()

**Sci-kit Learn**: This is the most popular machine learning libraries that comes with provisions to handle numpy arrays. This includes Train-Test splitting, scalers, encoders, imputers and a lot more, all can be used using a standard fit-transform technique.

Examples: X = sklearn.preprocessing.StandardScaler.fit\_transform(X)

2) Give an example on how ordinal data is handled in a Machine Learning algorithm.

=>

To handle ordinal categorical data, we assign them numbers tantamount to the significance of it. This can be done using a technique called LableEncoding.

**Example:**

Spiciness = [low , high , medium, low, low, high]

Encoded = sklearn.preprocessing.LableEncoding(Spiciness)

~ Encoded = [0,2,1,0,0,2]

3) Can one hot encoding be used for continuous data. If yes, give an example.

=>

The main purpose of OneHotEncoder is to encode categorical data, which therefore cannot be directly applied to continuous data. Although, if we can convert the continuous data into nominal data, we can practically use one hot encoding on it. One method of doing this is called binning.

**Example of Binning:**

Age = [9,25,27,13,15,76,14]

Binned\_ages = [“1-10”, “20-30”, “20-30”, “10-20”, “10-20”, “70-80”, “10-20”]

This data can now be One Hot Encoded.

4) Why is it necessary to encode strings?

=>

Most machine learning models are mathematical i.e. they perform mathematical calculations to predict the output. Therefore, all the inputs need to numeric. Therefore, we need to encode the data and convert them to numbers before passing them thorough a model.

5) State the significance of exploratory data analysis.

=>

Although machine learning models can learn most patterns themselves, there is some amount of processing and manipulation required before passing it through the process. We need some information like:

- What columns are more significant?

- How many classes does the dataset has?

- Is there a statistically significant relationship between a column and the target/ is a noticed trend statistically significant?

- Are there any missing values or outliers?

- Are there any problematic trends like multi-collinearity or a dummy-variable trap?

To get the answer to questions like these, we need exploratory data analysis. This includes methods like:

- Graphs and plots

- Ad Hoc testing and comparison

- Clustering

- Chi-Squared testing

- Statistics like R-Squared and Cramer’s V

6) ‘Handling missing values of data is an important step in Data preprocessing.’ Comment on the statement.

Missing data in a dataset is represented using NaN. This value cannot be processed by any type of machine learning model. Therefore, these need to be handled in order to proceed with creating a model. The most common technique is to drop the records with an NaN value in any column. However, this method then leads to loosing a lot of data in some cases. This can then be handled using methods like:

- Imputing with mean, median or other central tendency.

- Interpolating the data.

- Filling the missing slots with 0 or a *Missing* category

7) State any 4 graphical techniques/plots used for exploratory data analysis.

**Exploratory Charts:**

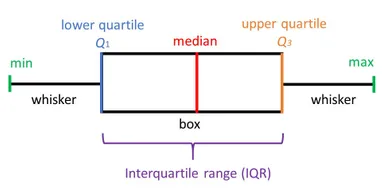
- Correlation Heatman

- Boxplot

- Histogram

- Barplot

8) Describe the ***box-and-whisker*** plot.



**Making the Plot:**

The whiskers are first drawn at the min and max points of the data. The first split (Q2) is made at the median of the data column. The next 2 splits are made at the median between the min and Q2 (Q1) and Q2 and the max (Q4). The difference between Q4 and Q1 is called the interquartile range. The range of the data is the difference between Maximum and Minimum.

**Interpreting The plot:**

Values lying in the interquartile range are considered normal. If a value lies more than one and a half times the length of the box from either end, the value is deemed to be an outlier.

9) Explain Central Tendency functions.

**Mean** : It is the central value of a discrete set of numbers.

**Median** : It is calculated by sorting the data and taking the central value. If the |X| is odd, the median is the central value. If the |X| is even, the median is the mean of the 2 middle values.

**Mode** : Is the value in a distribution that occurs the most.

**Conclusion:**

Data collection, data preparation, handling various data types was studied and exploratory data analysis was performed.