325982

INTERNSHIP AS A MACHINE LEARNING TRAINEE

A PROJECT REPORT

Submitted by

Dhanani Prit J

190420116011

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

Department of Information Technology

Sarvajanik College of Engineering & Technology, Surat

Gujarat Technological University, Ahmedabad

May, 2023

325982

Sarvajanik College of Engineering & Technology

Dr. R.K. Desai Marg, Opp. Mission Hospital, Athwalines, Surat – 395001, Gujarat, India

CERTIFICATE

This is to certify that the project report submitted along with the entitled **"Internship as a Machine Learning Trainee"** has been carried out by **Dhanani Prit** under my guidance in partial fulfillment for the degree of Bachelor of Engineering in Information Technology, 8th Semester of Gujarat Technological University, Ahmadabad during the academic year 2022-23.

Prof. (Dr.) Mita Parikh	Prof. (Dr.) Mita Parikh
Internal Mentor	Head of the Department

325982

Sarvajanik College of Engineering & Technology

Dr. R.K. Desai Marg, Opp. Mission Hospital, Athwalines, Surat – 395001, Gujarat, India

DECLARATION

I hereby declare that the Internship report submitted along with the Internship entitled **"TRAINING IN MACHINE LEARNING WITH PYTHONQ"** submitted in partial fulfillment for the degree of Bachelor of Engineering in Information Technology to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me under at TOPS Technologies under the supervision of **Prof.(Dr.) Mita Parikh** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student

Prit Dhanani

Sign of Student

_____

325982

325982

TOPS Technologies
Training - Outsourcing - Placement services

INTERNSHIP CERTIFICATE

Date : 02/05/2023

TO WHOM IT MAY CONCERN

This is to certify that **Prit Jerambhai Dhanani**, a student of Sarvajanik College of Engineering and Technology has successfully completed his internship in the field of Machine Learning from 06/02/2023 to 28/04/2023 under the guidance of Raj Yadav.

During his internship, he has demonstrated his skills with self-motivation to learn new skills. His performance exceeded our expectations and he was able to complete the target on time with the help of our mentor.

During the period of his internship program with us, he had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish him every success in his life and career.

Raj Yadav
(Project Head)

Signature

TOPS TECHNOLOGIES PRIVATE LIMITED

For, TOPS Technologies

9th Floor Samedh Complex CG Road Ahmedabad 079 30612162 | www.tops-int.com

325982

ACKNOWLEDGEMENT

I would like to express my sincerest gratitude to **Prof. (Dr.) Mita Parikh** who has guided me for the internship. Without your support and assistance, this achievement would not have been possible.

First and foremost, I would like to thank my supervisor for their guidance, patience, and encouragement throughout this entire process. Their insights and expertise were invaluable in shaping this work.

I would also like to extend my thanks to my colleagues and friends who provided me with invaluable feedback, support, and motivation. Your constructive criticism and words of encouragement were instrumental in helping me push through the difficult moments.

To recapitulate, I once again thank the faculties and members of **Sarvajanik College of Engineering and Technology** for their valuable support in completion of the project. Thank you all for your contributions and support

Sincerely,

Prit Dhanani

ABSTRACT

During my machine learning internship, I worked on developing and implementing various supervised learning algorithms to solve real-world problems. I began by learning the fundamental concepts of machine learning, including data preprocessing, feature selection, model selection, and evaluation metrics.

Throughout the internship, I also learned about the importance of data visualization and how it can be used to interpret and communicate the results of machine learning models effectively. Additionally, I gained experience with various data analysis tools such as Pandas, NumPy, and Matplotlib.

Overall, this internship provided me with a comprehensive understanding of the entire machine learning workflow, from data preprocessing to model selection and evaluation. The hands-on experience and exposure to various tools and techniques will be invaluable as I continue to develop my skills as a machine learning practitioner.

List of Figures

List of Abbreviations

Py	Python
Np	NumPy
Plt	Matplotlib
KNN	K Nearest Neighbors
SVM	Support Vector Machine
EDA	Exploratory data analysis
ML	Machine Learning
DA	Data Analysis

Table of Contents

1. ABOUT THE COMPANY

TOPS Technologies is one of the rapidly growing company that provides Software Development with Testing & Support services along with other professional development productivity.

We have worked for and have provided services to the 6 of the top 10 Fortune 500 companies of the world. Our team has worked in the US for more than 10 years and has management degrees from the US.

TOPS Technologies was awarded the TOP 2008 Business in the US for the year 2008- 2009.

Started with a vision of bridging the gap between skills required and talent created by colleges, today TOPS Technologies is one of the largest IT Training and Finishing with expertise in Data Science, Machine Learning, ASP.Net, PHP, Java, iPhone, Android, Software Testing, Web Design. Today we have Software Development training centers in Ahmedabad, Vadodara, Rajkot, Surat, Navsari, Mehsana, Junagadh, Indore, Bhopal, Jabalpur, Noida, Kota, Jaipur, Nagpur, Dehradun.

At TOPS Technologies we don't like to be termed as just an Outsourcing company; we believe in becoming your solutions partner. Our approach of "You think it, we build it" gives us the perfect mantra for the work we do. Our implementations have benefited organizations in creating the best solutions along with major cost savings.

- **TOPS Technologies offers IT Outsourcing Services includes:**

- Web Design and Development Services
- Mobile Apps Development Services (Android, iPhone)
- eCommerce Development Service
- Data Science (ML/DA)

2. INTRODUCTION OF PYTHON AND LIBRARIES

- **PYTHON:**

  - Python is a high-level, interpreted, interactive and object-oriented scripting language.
  - Python is designed to be highly readable.
  - It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
  - When it comes to Machine Learning, the python is core language, The entire ML is based on this only programming language.

- **NUMPY:**

  - NumPy is a Python library used for working with arrays.
  - It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
  - NumPy was created in 2005 by Travis Oliphant.
  - It is an open-source library and can use it freely.
  - NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow to process.
  - NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

- **PANDAS:**

  - Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.

- It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python
- Additionally, it has the broader goal of becoming the most powerful and flexible open-source data.
- It is pillar of machine learning; without it we cannot think about doing tasks in our dataset.
- It provides built in functions and methods, by using it we achieve our tasks.

• MATPLOTLIB:

- Matplotlib is one of the most popular Python packages used for data visualization.
- It is a cross-platform library for making 2D plots from data in arrays.
- It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter
- Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python.

• SEABORN:

- Seaborn is a library mostly used for statistical plotting in Python.
- It is built on top of Matplotlib and provides beautiful default styles and color palettes to make statistical plots more attractive.
- Seaborn helps explore and understand the data.

- **SKLEARN:**

- Sklearn (Scikit-learn) is the most useful and robust library for machine learning in Python.
- It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.
- This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.
- It is pillar of all Machine learning models and all the techniques which we perform in ML tasks.

3. EXPLORATORY DATA ANALYSIS

3.1 INTRODUCTION TO EDA

- Exploratory Data Analysis (EDA) is one of the techniques used for extracting vital features and trends used by machine learning and deep learning models in Data Science. Thus, EDA has become an important milestone for anyone working in data science.
- The Data Science field is now very important in the business world as it provides many opportunities to make vital business decisions by analysing hugely gathered data. Understanding the data thoroughly needs its exploration from every aspect. The impactful features enable making meaningful and beneficial decisions; therefore, EDA occupies an invaluable place in Data science

3.2 OBJECTIVES OF EDA

- Identifying and removing data outliers
- Identifying trends in time and space
- Uncover patterns related to the target
- Creating hypotheses and testing them through experiments
- Identifying new sources of data

3.3 STEPS INVOLVED IN EXPLORATORY DATA ANALYSIS (EDA)

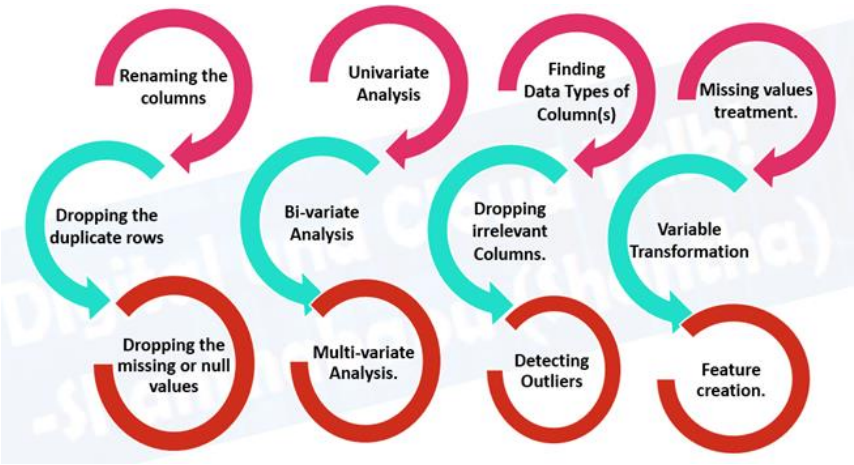

Figure 3-1 Flow of EDA

3.4 TYPES OF EDA

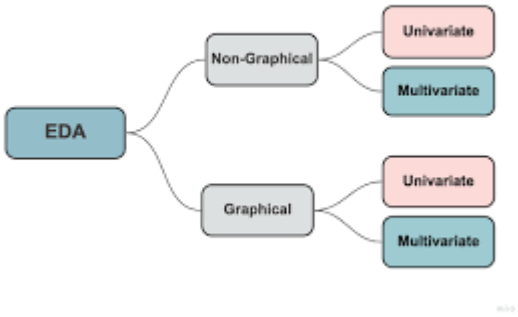

Figure 3-2 Types of EDA

1) **Univariate Analysis:** In univariate analysis, we analyze or deal with only one variable at a time. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it.

2) **Bi-Variate Analysis:** This type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship between the two variables.
3) **Multivariate Analysis**: When the data involves three or more variables, it is categorized under multivariate.

- Depending on the type of analysis we can also subcategorize EDA into two parts.

1. **Non-graphical Analysis** – In non-graphical analysis, we analyze data using statistical tools like mean, median or mode or skewness.

2. **Graphical Analysis** – In graphical analysis, we use visualizations charts to visualize trends and patterns in the data.

3.5 EDA PROJECT (US _ACCIDENT)

- Display The Dataset

```
df.iloc[:,1:20]
```

	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	End_Lat	End_Lng	Distance(mi)	Description	Number	Street	Side	City
0	3	2016-02-08 00:37:08	2016-02-08 06:37:08	40.108910	-83.092860	40.112060	-83.031870	3.230	Between Sawmill Rd/Exit 20 and OH-315/Olentang...	NaN	Outerbelt E	R	Dublin
1	2	2016-02-08 05:56:20	2016-02-08 11:56:20	39.865420	-84.062800	39.865010	-84.048730	0.747	At OH-4/OH-235/Exit 41 - Accident.	NaN	I-70 E	R	Dayton
2	2	2016-02-08 06:15:39	2016-02-08 12:15:39	39.102660	-84.524680	39.102090	-84.523960	0.055	At I-71/US-50/Exit 1 - Accident.	NaN	I-75 S	R	Cincinnati
3	2	2016-02-08 06:51:45	2016-02-08 12:51:45	41.062130	-81.537840	41.062170	-81.535470	0.123	At Dart Ave/Exit 21 - Accident.	NaN	I-77 N	R	Akron
4	3	2016-02-08 07:53:43	2016-02-08 13:53:43	39.172393	-84.492792	39.170476	-84.501798	0.500	At Mitchell Ave/Exit 6 - Accident.	NaN	I-75 S	R	Cincinnati
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2845337	2	2019-08-23 18:03:25	2019-08-23 18:32:01	34.002480	-117.379360	33.998880	-117.370940	0.543	At Market St - Accident.	NaN	Pomona Fwy E	R	Riverside

Figure 3-3 Display The data

- Visualize the missing percentage of data for each feature

```
In [8]: missing_percentages[missing_percentages != 0].plot(kind='barh')
Out[8]: <AxesSubplot:>
```

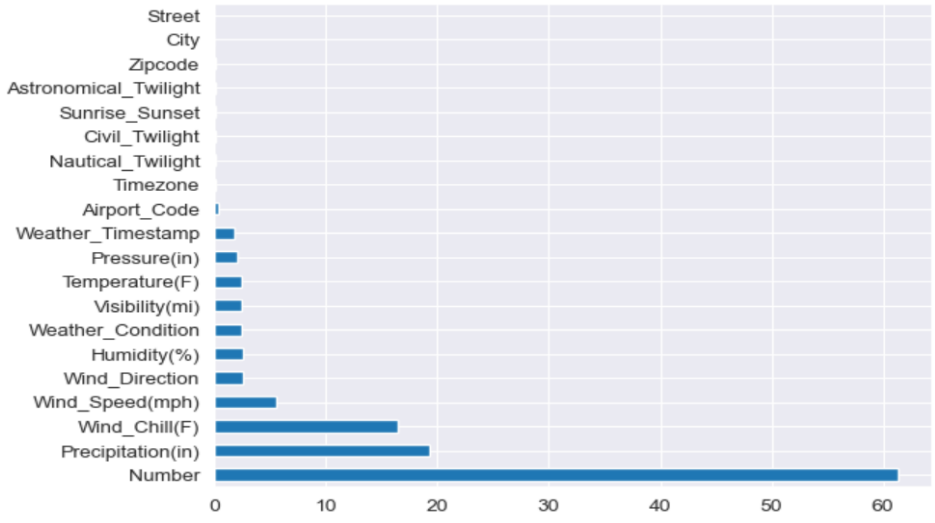

Figure 3-4 Visualizing missing values

- Here, Number, Precipitation, Wind_Chill has most missing values so we can drop these columns
- Display the columns

```
In [10]: df.columns

Out[10]: Index(['ID', 'Severity', 'Start_Time', 'End_Time', 'Start_Lat', 'Start_Lng',
                'End_Lat', 'End_Lng', 'Distance(mi)', 'Description', 'Number', 'Street',
                'Side', 'City', 'County', 'State', 'Zipcode', 'Country', 'Timezone',
                'Airport_Code', 'Weather_Timestamp', 'Temperature(F)', 'Wind_Chill(F)',
                'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Direction',
                'Wind_Speed(mph)', 'Precipitation(in)', 'Weather_Condition', 'Amenity',
                'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway',
                'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal',
                'Turning_Loop', 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilight',
                'Astronomical_Twilight'],
               dtype='object')
```

Figure 3-5 Data features

- Filtering the cities with most accidents, the bottom are the cities with 1 accident which is not possible so we can remove that are.

```
In [13]: cities_by_accident = df.City.value_counts()
         cities_by_accident

Out[13]: Miami                              106966
         Los Angeles                         68956
         Orlando                             54691
         Dallas                              41979
         Houston                             39448
                                              ...
         Ridgedale                               1
         Sekiu                                   1
         Wooldridge                              1
         Bullock                                 1
         American Fork-Pleasant Grove            1
         Name: City, Length: 11681, dtype: int64
```

Figure 3-6 Cities by Accident

Visualize the top 20 cities by accident with count of accident

```
cities_by_accident[:20].plot(kind='barh')
```

```
<AxesSubplot:>
```

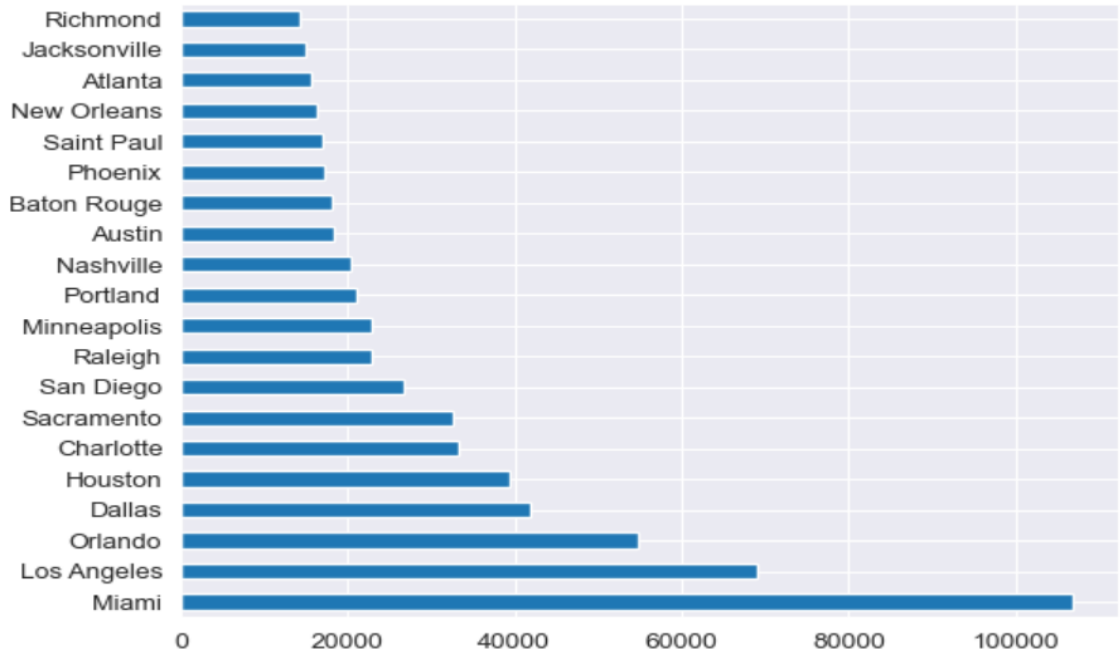

Figure 3-7 Visualize cities with accident count

- From this graph we can interpret that most accident held on Miami, LA, Orlando.

- Visualize the dataset with percentage of accident with hours.
- From this bar graph we can say that most accident done on 10 a.m.to 4 p.m. period of time.

```
In [92]: sns.distplot(df.Start_Time.dt.hour, bins=24, kde=False, norm_hist=True)
Out[92]: <Axes: xlabel='Start_Time'>
```

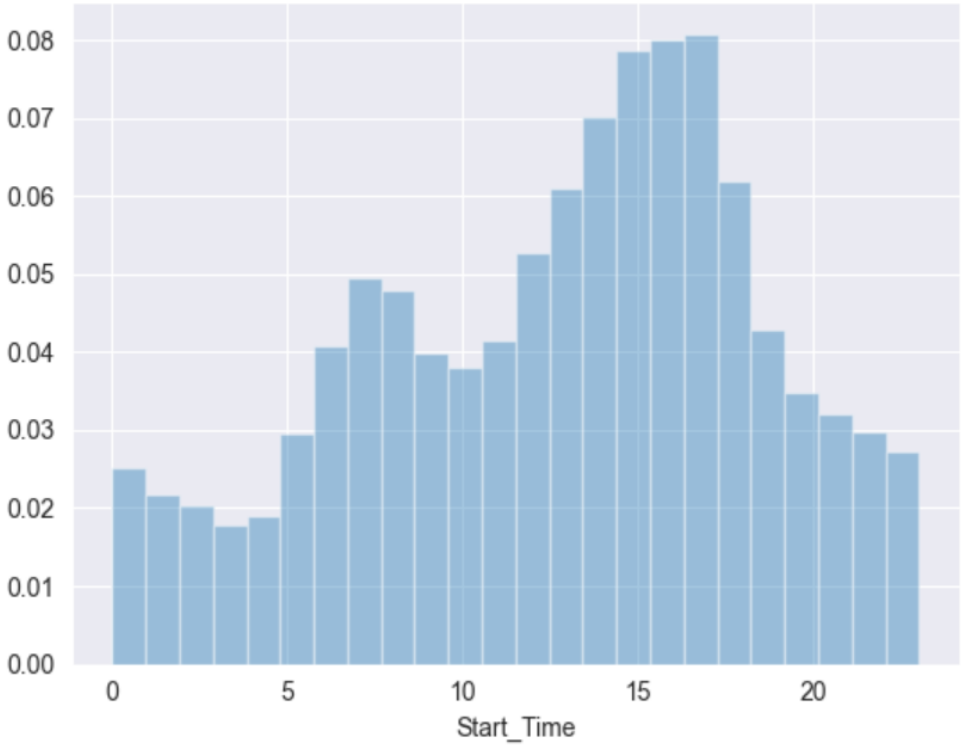

Figure 3-8 Visualize the accident on a day hour

- Visualize the dataset with percentage of accident with week.

```
In [25]: sns.distplot(df.Start_Time.dt.dayofweek, bins=7, kde=False, norm_hist=True)
Out[25]: <AxesSubplot:xlabel='Start_Time'>
```

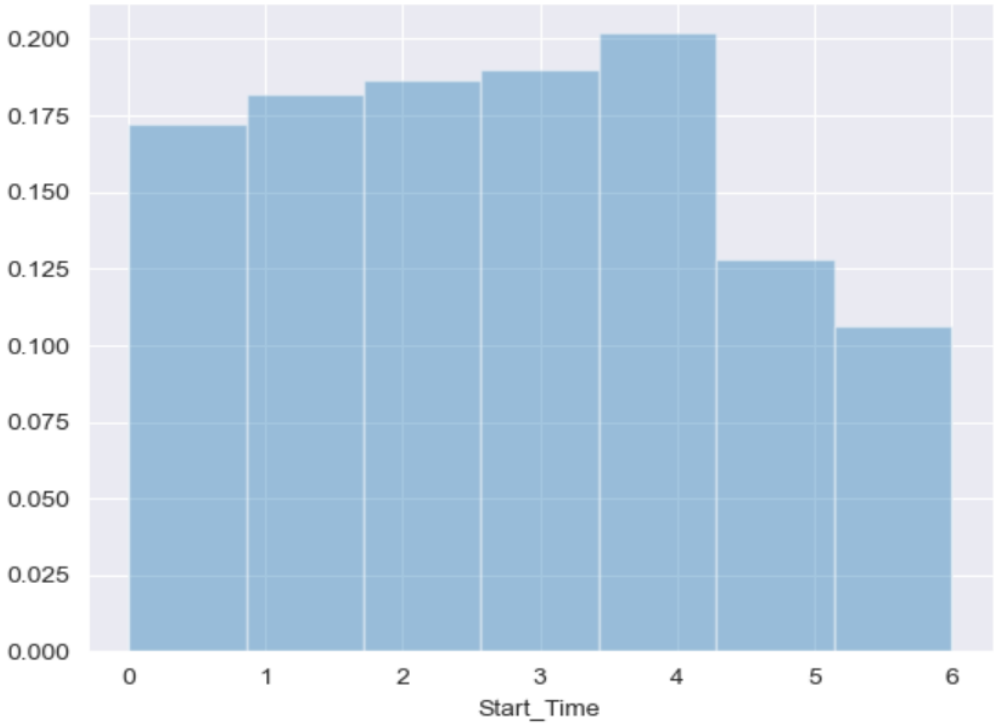

Figure 3-9 Visualize the accident on a week

- Plotting coordinates with the help of scatterplot.

```
In [32]: sns.scatterplot(x=sample_df.Start_Lng, y=sample_df.Start_Lat, size=0.001)
Out[32]: <AxesSubplot:xlabel='Start_Lng', ylabel='Start_Lat'>
```

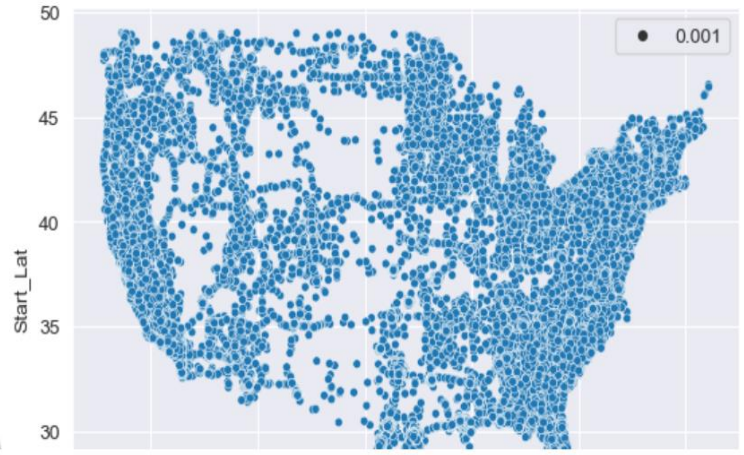

Figure 3-10 Visualize the coordinate on scatter plot

- Outcome of EDA Project
  - Less than 5% of cities have more than 1000 yearly accidents.
  - Over 1200 cities have reported just one accident (need to investigate)
  - most accident held on Miami, LA, Orlando.
  - most accident done on 10 a.m.to 4 p.m. period of time.
  - most accident done on the Monday to Friday.

4. INTRODUCTION TO MACHINE LEARNING

4.1 WHAT IS MACHINE LEARNING

Machine learning is an application of AI that enables systems to learn and improve from experience without being explicitly programmed. Machine learning focuses on developing computer programs that can access data and use it to learn for themselves.\

With the ever-increasing amounts of data becoming available there is good reason to believe that smart data analysis will become even more pervasive as a necessary ingredient for technological progress.

4.2 HOW DOES MACHINE LEARNING WORK?

Similar to how the human brain gains knowledge and understanding, machine learning relies on input, such as training data or knowledge graphs, to understand entities, domains and the connections between them. With entities defined, deep learning can begin.

The machine learning process begins with observations or data, such as examples, direct experience or instruction. It looks for patterns in data so it can later make inferences based on the examples provided. The primary aim of ML is to allow computers to learn autonomously without human intervention or assistance and adjust actions accordingly.

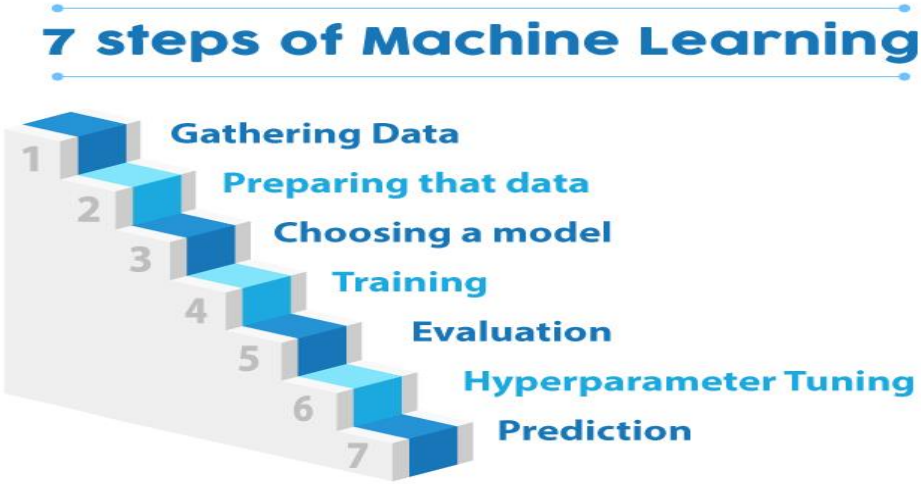

Figure 4-1 process of ml

4.3 IMPORTANCE OF MACHINE LEARNING.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

Applications:

- Voice assistants
- Google News
- Recommendation systems
- Face Recognition source
- Auto-completion
- Stock market prediction
- Character recognition
- AlphaGo
- Self-driving cars
- Drug discovery

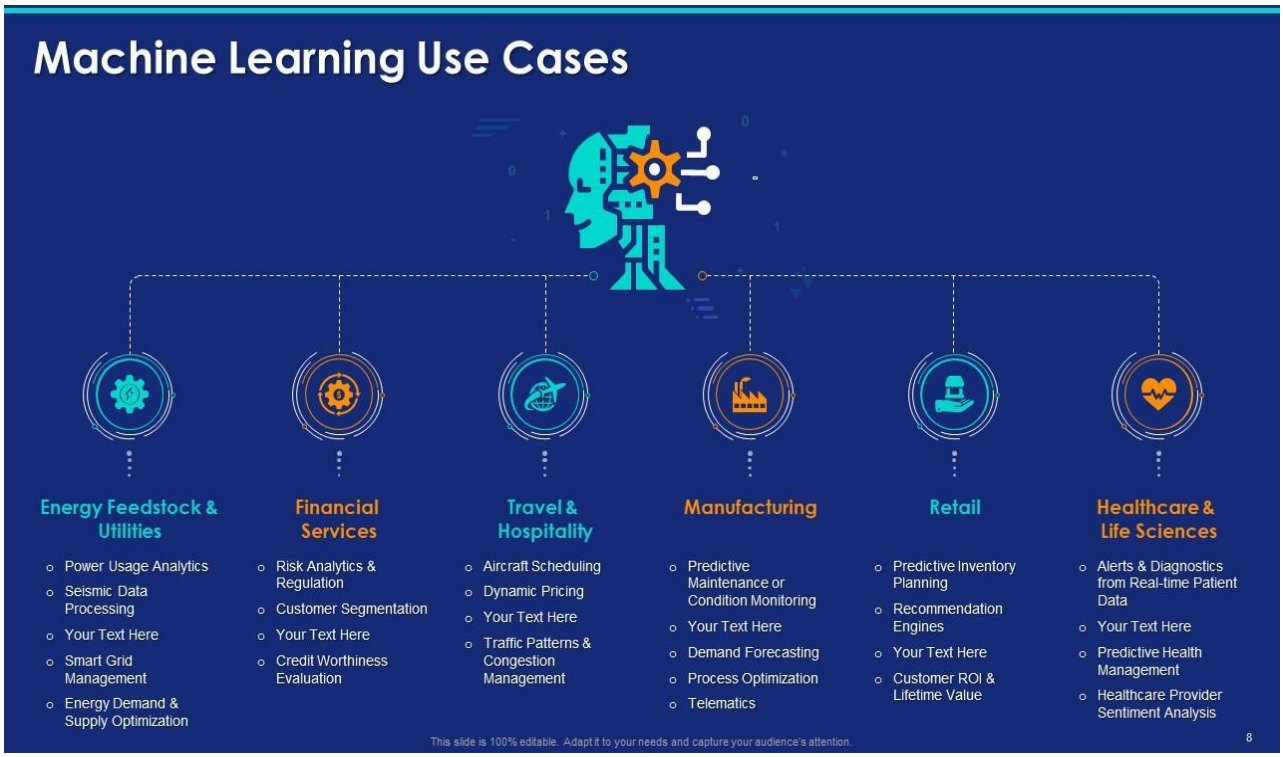

Figure 4-2 use case of machine learning

4.4 RELATIONSHIP BETWEEN AI AND ML

To sum things up, AI solves tasks that require human intelligence while ML is a subset of artificial intelligence that solves specific tasks by learning from data and making predictions.

This means that all machine learning is AI, but not all AI is machine learning.

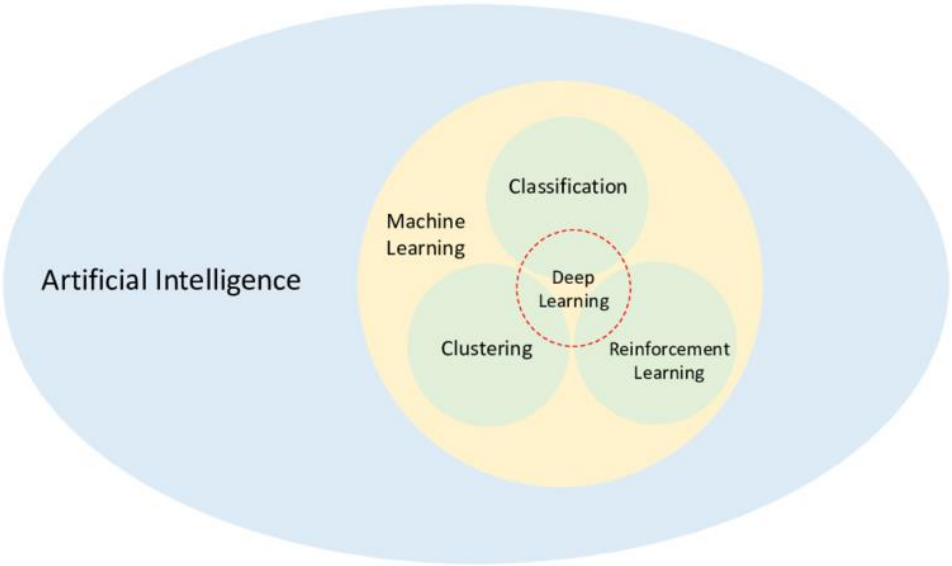

Figure 4-3 Field of Artificial intelligence

4.5MACHINE LEARNING ALGORITHM

- Traditional Programming Vs Machine Learning

➢ Machine learning can also adapt to changes in the data set, whereas raditional methods can become less accurate over time. As the data set changes, machine learning will adapt its predictions accordingly. This ensures that the predictions are always accurate and up-to-date.

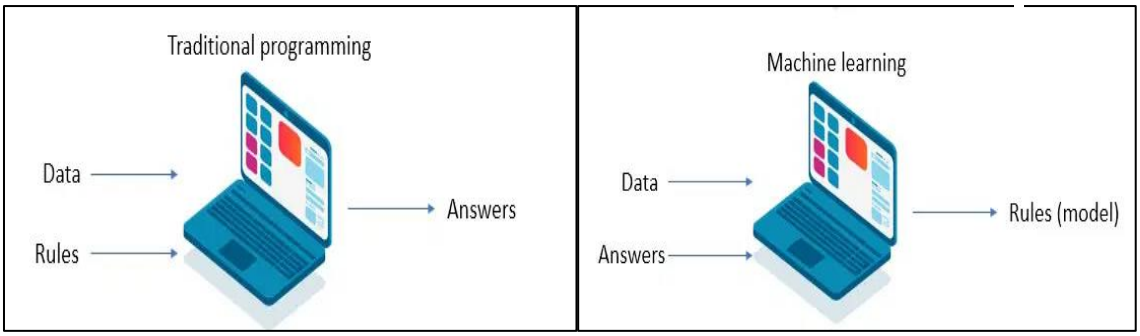

Figure 4-4 Traditional Vs Machine learning

- Machine Learning Techniques

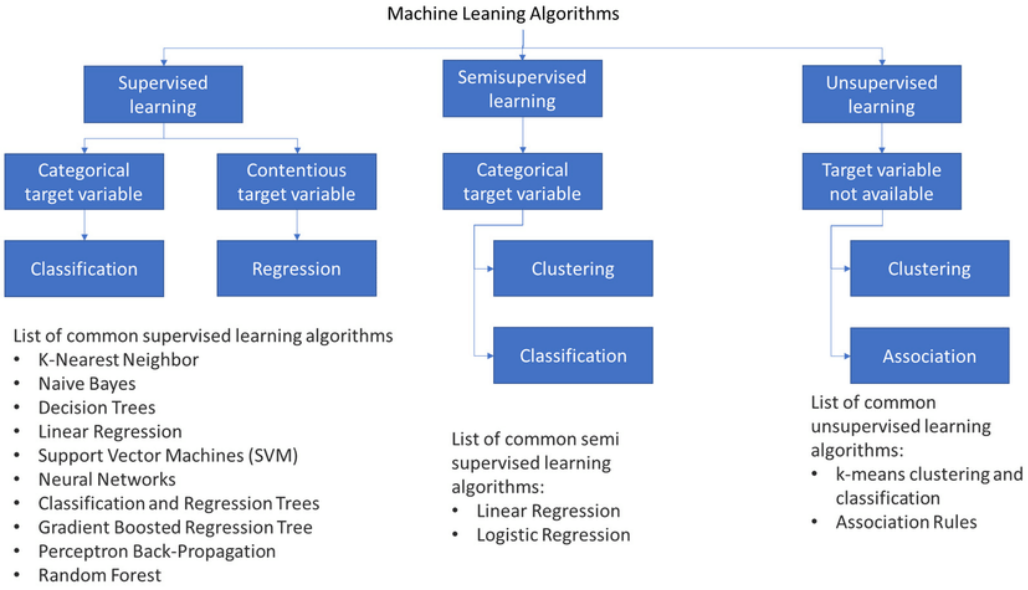

Figure 4-5 Machine Learning Algorithm Classification

4.6 TYPES OF MACHINE LEARNING

4..1 SUPERVISED LEARNING

- What is supervised learning?

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples

Training a model from input data and its corresponding targets to predict targets for new examples.

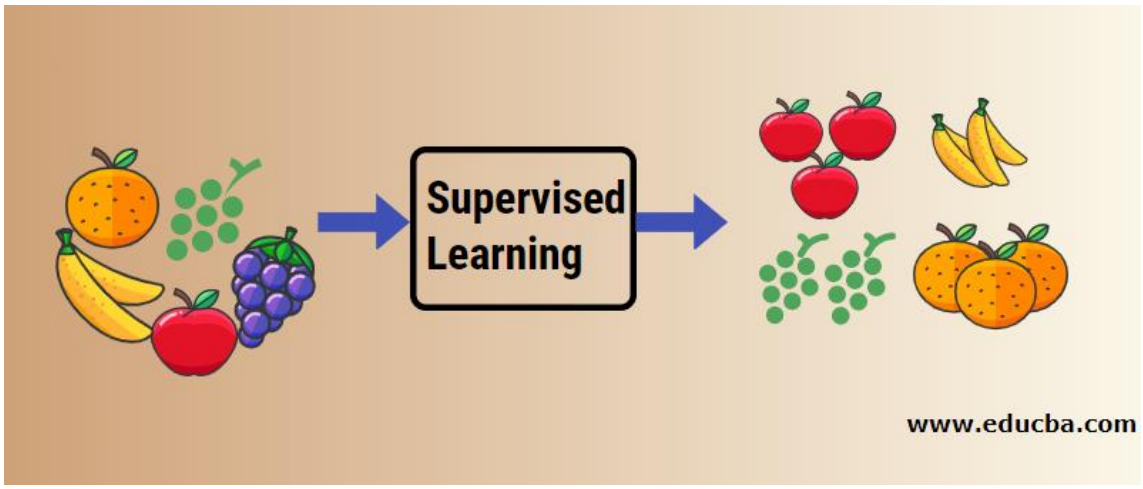

Figure 4-6 Example of supervised ml

4.6.2 UNSUPERVISED LEARNING

- What is Unsupervised learning?

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance

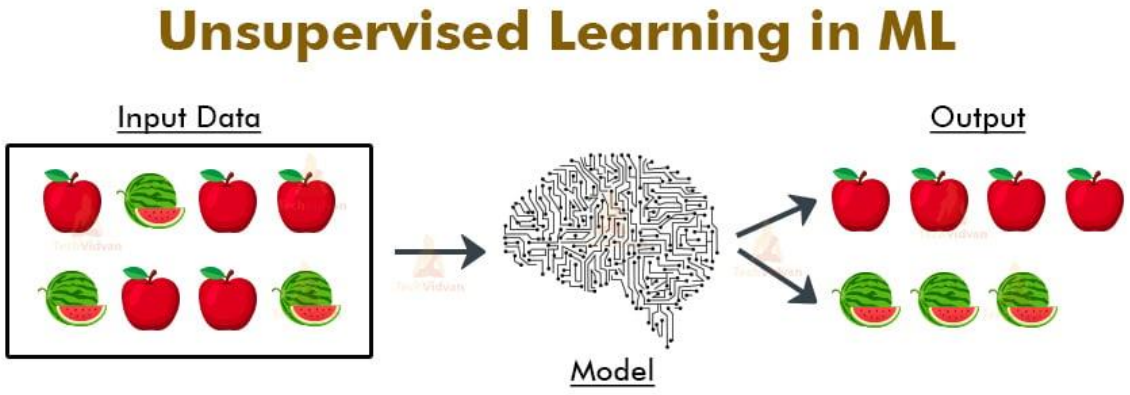

Figure 4-7 Example of unsupervised learning

4.6.3 SEMI- SUPERVISED LEARNING

- What is semi-supervised learning?

  It is combination of supervised and unsupervised learning

Semi-supervised learning is a class of machine learning tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data).

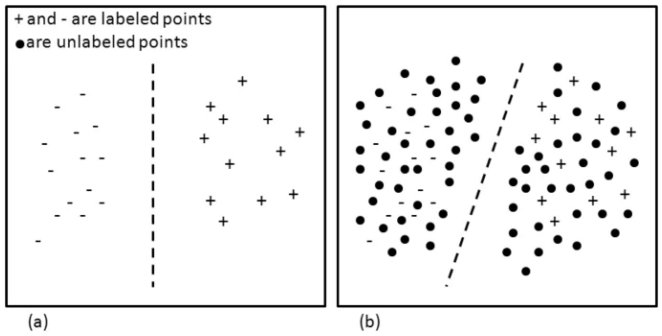

Figure 4.8 semi-supervised learning

For (Figure 4.8)
(a) The decision boundary in presence of labeled data points only, and
(b) the decision boundary in presence of both labeled and unlabeled data.

Semi-supervised learning tries to increase the generalization of classification performance by placing the decision boundary through the sparse regions in presence of both labeled and unlabeled data points.

4.6.4 REINFORCEMENT LEARNING

Reinforcement Learning is a type of Machine Learning that allows the learning system to observe the environment and learn the ideal behavior based on trying to maximize some notion of cumulative reward.

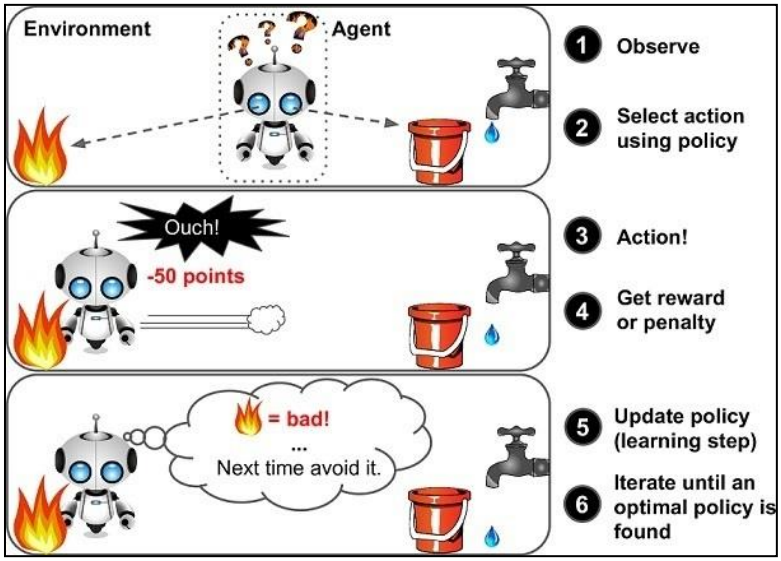

Figure 4-9 example of reinforcement learning

5 DATA PRE-PROCESSING

5.1WHAT IS DATA PRE-PROCESSING

Data pre-processing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data pre-processing is to improve the quality of the data and to make it more suitable for the specific data mining task.

5.2 WHY IS DATA PREPROCESSING IMPORTANT?

It improves accuracy and reliability. Preprocessing data removes missing or inconsistent data values resulting from human or computer error, which can improve the accuracy and quality of a dataset, making it more reliable.

It makes data consistent. When collecting data, it's possible to have data duplicates, and discarding them during preprocessing can ensure the data values for analysis are consistent, which helps produce accurate results.

It increases the data's algorithm readability. Preprocessing enhances the data's quality and makes it easier for machine learning algorithms to read, use, and interpret it.

5.2COMMON PREPROCESSING TECHNIQUES

1 Imputation

- Data imputation is a method for retaining the majority of the dataset's data and information by substituting missing data with a different value. These methods are employed because it would be impractical to remove data from a dataset each time. Additionally, doing so would substantially reduce the dataset's size, raising questions about bias and impairing analysis.

Data imputation Techniques

- Next or Previous Value
- Average or Linear Interpolation
- Fixed Value
- K Nearest Neighbors
- Maximum or Minimum Value
- Missing Value Prediction
- Most Frequent Value
- Practical Imputation using scikit learn – tool

```
# Preprocessing and pipeline
from sklearn.impute import SimpleImputer
```

```
In [31]: imputer = SimpleImputer(strategy="median")
         imputer.fit(X_train)
         X_train_imp = imputer.transform(X_train)
         X_test_imp = imputer.transform(X_test)
```

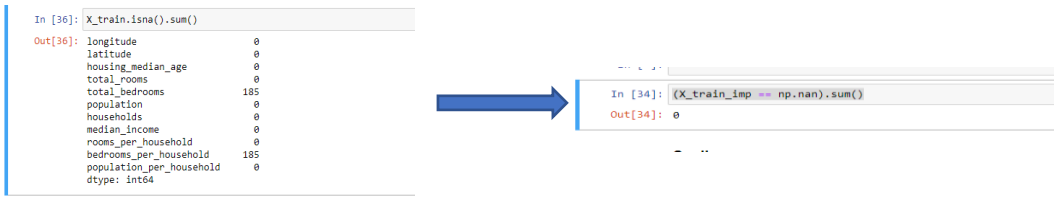

Figure 0-1 Imputation

2 Feature Scaling

- Feature scaling is an important step in the data transformation stage of data preparation process.
- Feature Scaling is a method used in Machine Learning for standardization of independent variables of data features

Techniques of Feature Scaling

1 Standardization

Standardization is a popular feature scaling method, which gives data the property of a standard normal distribution (also known as Gaussian distribution).

All features are standardized on the normal distribution (a mathematical model). The mean of each feature is centered at zero, and the feature column has a standard deviation of one.

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation } (x)}$$

1 Normalization

In most cases, normalization refers to rescaling of data features between 0 and 1, which is a special case of Min-Max scaling.

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

In the given equation, subtract the min value for each feature from each feature instance and divide by the spread between max and min.

In effect, it measures the relative percentage of distance of each instance from the min value for that feature.

2 Encoding

Encoding categorical data is a process of converting categorical data into integer format so that the data with converted categorical values can be provided to the models to give and improve the predictions.

Encoding categorical data is one of such tasks which is considered crucial. As we know, most of the data in real life come with categorical string values and most of the

machine learning models work with integer values only and some with other different values which can be understandable for the model.

All models basically perform mathematical operations which can be performed using different tools and techniques. But the harsh truth is that mathematics is totally dependent on numbers. So in short we can say most of the models require numbers as the data, not strings or not anything else and these numbers can be float or integer.

\

6 PROJECT

6.1 CONTEXT

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes.

6.2 INFORMATION ABOUT DATASET

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases.

The dataset has several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The features are:

1. Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
2. BloodPressure: Diastolic blood pressure (mm Hg)
3. SkinThickness: Triceps skin fold thickness (mm)
4. Insulin: 2-Hour serum insulin (mu U/ml)
5. BMI: Body mass index (weight in kg/(height in m)^2)
6. DiabetesPedigreeFunction: Diabetes pedigree function
7. Age: Age (years)
8. Outcome: Class variable (0 or 1)

- **Features in detail:**

1. Diastolic blood pressure (mm Hg):

- Diastolic blood pressure refers to the pressure in your blood vessels when your heart is resting between beats. It is the second and lower number in a blood pressure reading, written as the bottom number.

- Blood pressure is measured in millimetres of mercury (mm Hg) and is typically recorded as two numbers: systolic pressure (the top number) and diastolic pressure (the bottom number).

- A normal diastolic blood pressure reading is typically around 80 mm Hg or lower. A reading of 90 mm Hg or higher is generally considered high and may indicate hypertension (high blood pressure), which can increase the risk of cardiovascular disease, stroke, and other health problems.

2. Triceps skin fold thickness (mm):

- Triceps skinfold thickness (TST) is a measurement of the amount of subcutaneous fat (fat located just beneath the skin) in the triceps area.
- The measurement is taken by pinching a fold of skin and subcutaneous fat using a calliper, usually at the midpoint between the shoulder and elbow on the back of the upper arm. The thickness of the skinfold is then measured in millimetres.

3. 2-Hour serum insulin (mu U/ml):

- A 2-hour serum insulin test measures the amount of insulin present in the blood two hours after a person consumes glucose. This test is often used to diagnose insulin resistance and diabetes.

4 Body mass index (weight in kg/(height in m)^2):

- Body mass index (BMI) is a measure of body fat based on your weight in kilograms divided by the square of your height in meters. It is a commonly used indicator of whether a person has a healthy body weight.

5 Diabetes pedigree function:

- The Diabetes Pedigree Function (DPF) is a mathematical formula used to predict the risk of developing diabetes based on a family history of the disease.

- The formula takes into account the age at which family members were diagnosed with diabetes and the number of affected relatives in the family.

6.3 STEPS TO UNDERSTAND, PRE-PROCESS, LEARN AND TRAIN THE DATA(EDA)

Import the needed libraries

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
        import seaborn as sns
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler # for scaling
        import pickle # for Pickling
        from sklearn.linear_model import LogisticRegression

        from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
        from sklearn.model_selection import GridSearchCV # for hyper parameter tuning
        from sklearn import metrics
        import warnings
        warnings.filterwarnings('ignore')
```

Figure 6-2 Library

1. Understanding the datasets by looking at the data.

```
In [2]: df = pd.read_csv('diabetes.csv')
```

```
In [3]: df
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Figure 6-3 View dataset

Using pandas, read the data with read_csv function and print in form of DataFrame.

2. Try to understand the data's features.

- Performing non graphical EDA on dataset.

```
df['Pregnancies'].value_counts()
```

```
Pregnancies
1      135
0      111
2      103
3       75
4       68
5       57
6       50
7       45
8       38
9       28
10      24
11      11
13      10
12       9
14       2
15       1
17       1
Name: count, dtype: int64
```

Figure 6-4 dataset value count

```
In [4]: df.nunique()
```

```
Out[4]: Pregnancies                 17
        Glucose                    136
        BloodPressure               47
        SkinThickness               51
        Insulin                    186
        BMI                        248
        DiabetesPedigreeFunction   517
        Age                         52
        Outcome                      2
        dtype: int64
```

Figure 6-5 dataset unique values

```
In [6]: df.isnull().sum()
Out[6]: Pregnancies                 0
        Glucose                     0
        BloodPressure               0
        SkinThickness               0
        Insulin                     0
        BMI                         0
        DiabetesPedigreeFunction    0
        Age                         0
        Outcome                     0
        dtype: int64
```

Figure 6-6 null values

```
In [8]: df.describe()
```

Out[8]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Figure 6-7 Getting knowing the dataset

3. Pre-process the datasets.

- Finding the Outliers in all the columns with the help of box plot.
- Start with Glucose feature.

```
In [10]: # Checking for outliers
         sns.boxplot(df['Glucose'])

Out[10]: <Axes: >
```

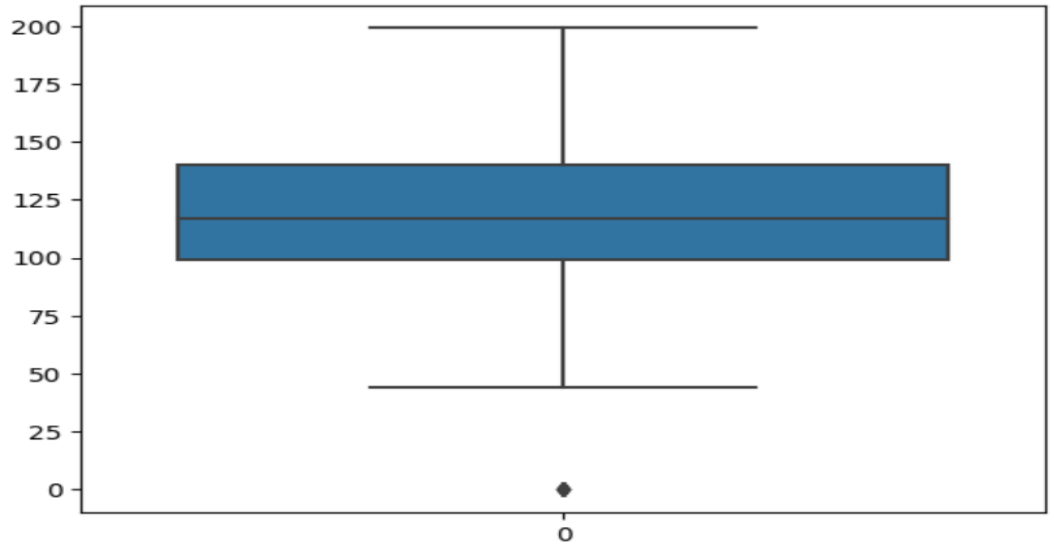

Figure 6-8 dataset boxplot 1

- Outlier is 0 in this feature, and try to remove it from dataset using replace method and the replaced value is mean of all the data which available in Glucose field.

```
In [11]: # doing median Imputaion
         df['Glucose'] = df['Glucose'].replace(to_replace=0,value=df['Glucose'].mean())

In [12]: mask = df['Glucose'] == 0
         df[mask]
         sns.boxplot(df['Glucose'])

         # now it is fine

Out[12]: <Axes: >
```

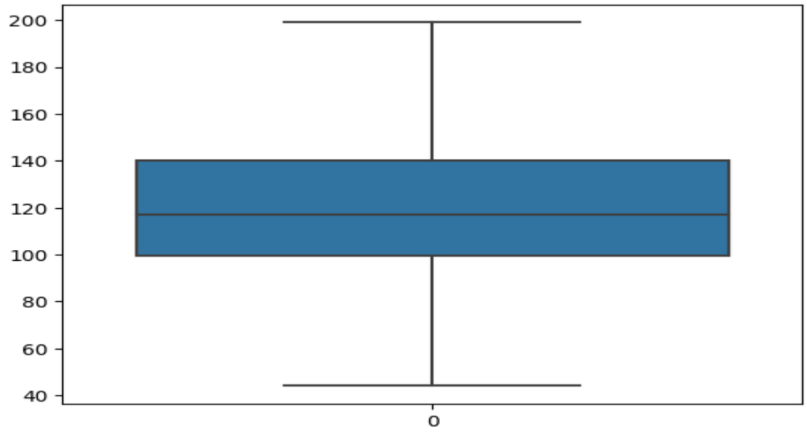

Figure 6-9 dataset boxplot 2

- With this way, all the feature will be cleaned

SkinThickness feature

```
In [21]: plt.subplots(1,3,figsize = (20,7))
         plt.subplot(1,3,1)
         plt.title('Checking for Outliers')
         sns.boxplot(df['BMI'])
         plt.subplot(1,3,2)
         plt.title('before replacing')
         sns.histplot(df['SkinThickness'],kde=True)
         plt.subplot(1,3,3)
         df['SkinThickness'] = df['SkinThickness'].replace(to_replace=0,value=df['SkinThickness'].median())
         plt.title('after replacing')
         sns.histplot(df['SkinThickness'],kde=True)

Out[21]: <Axes: title={'center': 'after replacing'}, xlabel='SkinThickness', ylabel='Count'>
```

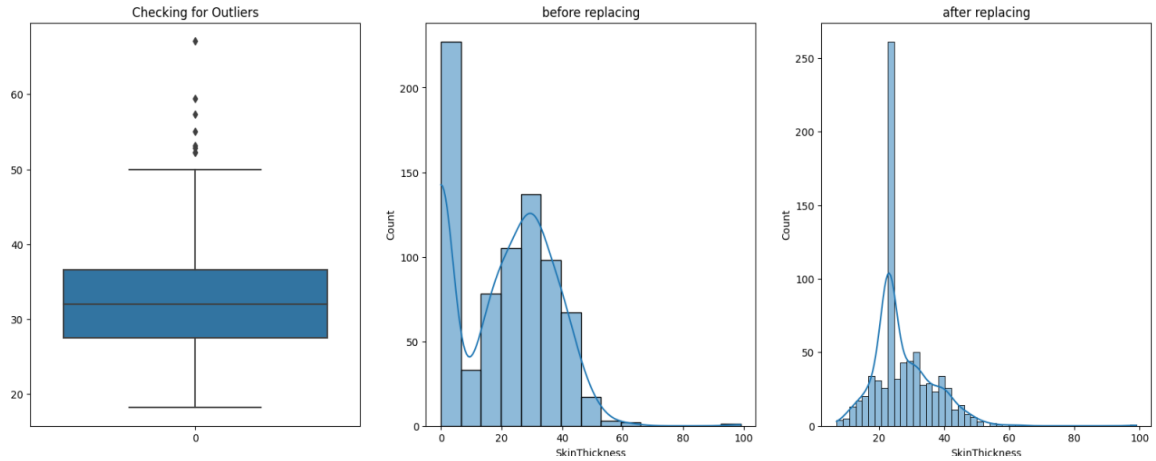

Figure 6-10 dataset boxplot and bar plot

Visualize the all features with heatmap, It shows that if there any feature is likelihood to another feature

```
In [29]: plt.figure(figsize=(8,5))
         # seaborn has an easy method to showcase heatmap
         p = sns.heatmap(df.corr(), annot=True,cmap ='RdYlGn')
```

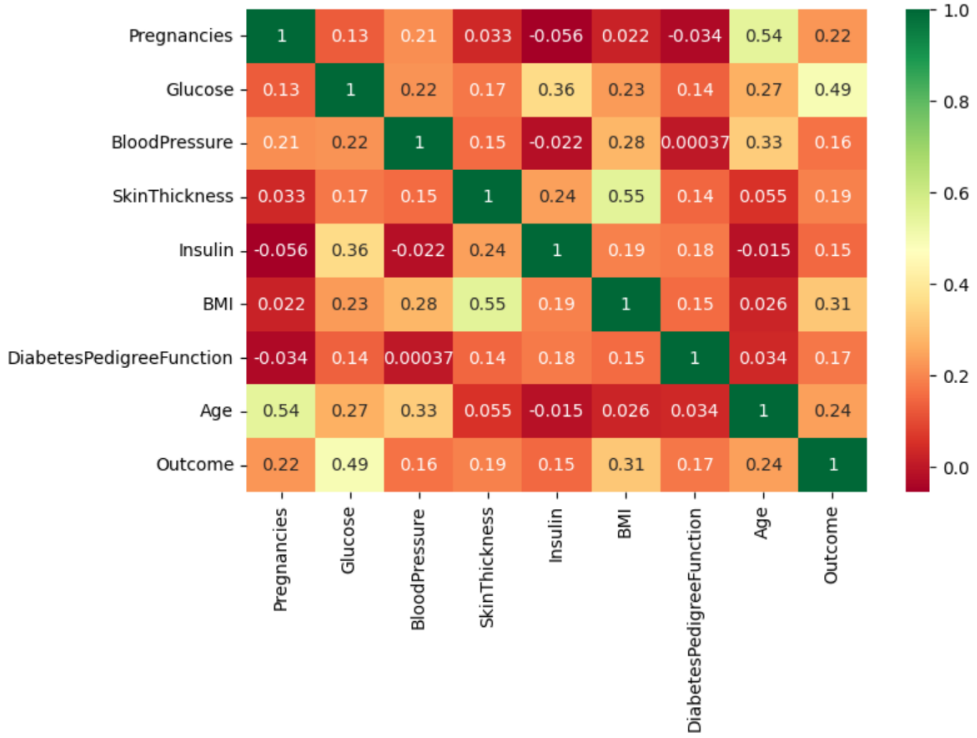

Figure 6-11 Dataset Heatmap

- Now, Scaling the data, the reason is some of the feature is in high numeric numbers compare to other features.

- Thus, Implement StandartdScaler() method for scaling.

Standard Scaling

```
In [57]: def standard_scaler(X_train,X_test):
             # scaling the data
             scaler = StandardScaler() # made instance
             X_train_scaled = scaler.fit_transform(X_train)
             X_test_scaled = scaler.transform(X_test)



             return X_train_scaled,X_test_scaled

In [58]: X_train_scaled,X_test_scaled=standard_scaler(X_train=X_train,X_test=X_test)

In [60]: X_train_scaled

Out[60]: array([[ 1.50755225, -1.09947934, -0.89942504, ..., -1.45563645,
                  -0.98325882, -0.04863985],
                [-0.82986389, -0.1331471 , -1.23618124, ...,  0.0927149 ,
                  -0.62493647, -0.88246592],
                [-1.12204091, -1.03283573,  0.61597784, ..., -0.03631437,
                   0.39884168, -0.5489355 ],
                ...,
                [ 0.04666716, -0.93287033, -0.64685789, ..., -1.14023155,
                  -0.96519215, -1.04923114],
                [ 2.09190629, -1.23276654,  0.11084355, ..., -0.36605587,
                  -0.5075031 ,  0.11812536],
                [ 0.33884418,  0.46664532,  0.78435594, ..., -0.09366072,
                   0.51627505,  2.953134  ]])
```

Figure 6-12 dataset scaling

- In our case no features are categorical so, that OneHotEncoding, Ordinal encoding cannot do.

4. Train the scaled model with Different models.

Logistic Regression

```
In [30]: classifier = LogisticRegression()
         classifier

Out[30]: LogisticRegression
         LogisticRegression()

In [31]: classifier.fit(X_train_scaled,y_train)
         classifier.score(X_train_scaled,y_train)

Out[31]: 0.7725694444444444

In [32]: accuracy_score(y_test,classifier.predict(X_test_scaled))

Out[32]: 0.796875

In [33]: y_pred_log = classifier.predict(X_test_scaled)

In [34]: print(confusion_matrix(y_test,y_pred_log))
         print(classification_report(y_test,y_pred_log))

[[117  13]
 [ 26  36]]
              precision    recall  f1-score   support

           0       0.82      0.90      0.86       130
           1       0.73      0.58      0.65        62

    accuracy                           0.80       192
   macro avg       0.78      0.74      0.75       192
weighted avg       0.79      0.80      0.79       192
```

Figure 6-13 Logistic Regression

- While using Logistic Regression the train score is 77.2% and test score is 79%
- Which are bearable numbers but bot great
- Thus, train another model.

Random Forest

```
In [35]: from sklearn.ensemble import RandomForestClassifier

         rfc = RandomForestClassifier(n_estimators=2000)
         rfc.fit(X_train, y_train)
```

```
Out[35]: RandomForestClassifier
         RandomForestClassifier(n_estimators=2000)
```

```
In [36]: rfc_train = rfc.predict(X_train)
         print("Accuracy_Score =", format(metrics.accuracy_score(y_train, rfc_train)))

         Accuracy_Score = 1.0
```

```
In [37]: #overfitting
```

```
In [38]: y_predictions = rfc.predict(X_test)
         print("Accuracy_Score =", format(metrics.accuracy_score(y_test, y_predictions)))

         Accuracy_Score = 0.7916666666666666
```

```
In [39]: print(classification_report(y_test,y_predictions))

                       precision    recall  f1-score   support

                    0       0.82      0.88      0.85       130
                    1       0.71      0.60      0.65        62

             accuracy                           0.79       192
            macro avg       0.77      0.74      0.75       192
         weighted avg       0.79      0.79      0.79       192
```

Figure 6-14 Random Forest Classification

- Using Random Forest Classifier, looks like training data is overfitted but it normal in this model.
- But the test score is impressive compare to Logistic Regressor.
- The f1 score is also similar but random forest gives more precision with recall.

Support Vector Machine (SVM)

In [40]:
```
from sklearn.svm import SVC

svc_model = SVC()
svc_model.fit(X_train_scaled, y_train)
svc_model.score(X_train_scaled, y_train)
```

Out[40]:
```
0.8246527777777778
```

In [41]:
```
svc_pred = svc_model.predict(X_test_scaled)
print("Accuracy Score =", format(metrics.accuracy_score(y_test, svc_pred)))
```

```
Accuracy Score = 0.7552083333333334
```

In [42]:
```
svc_pred
```

Out[42]:
```
array([1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

In [43]:
```
print(confusion_matrix(y_test, svc_pred))
print(classification_report(y_test,svc_pred))
```

```
[[115  15]
 [ 32  30]]
              precision    recall  f1-score   support

           0       0.78      0.88      0.83       130
           1       0.67      0.48      0.56        62

    accuracy                           0.76       192
   macro avg       0.72      0.68      0.70       192
weighted avg       0.74      0.76      0.74       192
```

Figure 6-15 SVM

- SVM is one of the linear classifiers.
- But test score is not so impressive here compare to random forest classifier.

K Nearest Neighbor(KNN)

```
In [56]: from sklearn.neighbors import KNeighborsClassifier
         knn_model = KNeighborsClassifier(n_neighbors=7)
         knn_model.fit(X_train_scaled,y_train)
         knn_model.score(X_train_scaled,y_train)

Out[56]: 0.7951388888888888

In [57]: knn_pred = knn_model.predict(X_test_scaled)
         print("Accuracy Score =", format(metrics.accuracy_score(y_test, knn_pred)))

         Accuracy Score = 0.7708333333333334

In [58]: print(confusion_matrix(y_test, knn_pred))
         print(classification_report(y_test,knn_pred))

         [[114  16]
          [ 28  34]]
                       precision    recall  f1-score   support

                    0       0.80      0.88      0.84       130
                    1       0.68      0.55      0.61        62

             accuracy                           0.77       192
            macro avg       0.74      0.71      0.72       192
         weighted avg       0.76      0.77      0.76       192
```

Figure 6-16 KNN

- K_Nearest_Neighbor is Classifier model.
- Uses K for choosing the number of neighbours if k Is higher the model will too complex
- and might end with curse of dimensionality.
- So, far Random Forest Classifier perform well against this dataset with default parameters.

7 CONCLUSION

7.1 OVERALL OUTCOME OF INTERNSHIP

In conclusion, completing an internship in machine learning is a valuable experience that can provide me with a solid foundation in this exciting field. Through my internship, I have gained hands-on experience with various machine learning techniques, tools, and technologies. I have also had the opportunity to work with a team of professionals and gain insight into the practical applications of machine learning in real-world scenarios.

The skills and knowledge I have acquired during your internship will serve me well in your future endeavors, whether I choose to pursue further education in machine learning or pursue a career in this field. Remember to continue practicing and refining my skills to stay up-to-date with the latest advancements in machine learning. With dedication and hard work, I can become a proficient machine learning practitioner and make significant contributions to this growing field.

REFERENCES

1. https://www.linkedin.com/pulse/machine-learning-vs-traditional-programming-kailash-rajeshmaharaj/?articleId=6406820979540054016
2. https://www.researchgate.net/figure/Overview-of-machine-learning-techniques_fig1_348764759
3. https://levelup.gitconnected.com/algorithm-test-engineering-part-2-machine-learning-1c539e9c7a88
4. https://onlinebme.com/overfitting-and-underfitting/
5. https://www.javatpoint.com/supervised-machine-learning
6. https://medium.com/analytics-vidhya/the-math-behind-logistic-regression-c2f04ca27bca
7. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/
8. https://scientistcafe.com/ids/regression-and-decision-tree-basic.html
9. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/
10. https://techvidvan.com/tutorials/unsupervised-learning/
11. https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c
12. https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c
13. https://www.javatpoint.com/reinforcement-learning
14. https://www.kaggle.com/datasets/amitabhajoy/US_Accidents(DES_21)_UPDATED