Chapter 1

INTRODUCTIONABOUTTHECOMPANY

- Karunadu Technologies is a leading technology company specializing in software development, data analysis, and machine learning solutions.
- A leading Skills and Talent Development company that is building a manpower pool for global industry requirements.
- Empower individual with knowledge, skills and competencies that assist to escalate as integrated individuals with a sense of commitment and dedication towards the society.

1.1 Profile

Fig 1.1 Company Logo

- The company offers broad range of customized software applications powered by concrete technology and industry expertise. It also offers end to end embedded solutions and services. They deal with broad range of product development along with customized features ensuring at most customer satisfaction and also empower individual with knowledge, skills and competencies that assist them to escalate as integrated individuals with a sense of commitment and dedication.
- **Vision:** To Empower Unskilled Individual with knowledge, skills and technical competencies in the field of Information Technology and Embedded engineering which assist them to escalate as integrated individuals contributing to company's and Nation's growth.
- **Mission:** Create, design and deliver business solutions with high value and innovation by leveraging technology expertise and innovative business models to

address long-term business objectives. Focus on creating sustainable value growth through innovative solutions and unique partnerships.

1.2:Overview of the Organization

Objectives

- To develop software and Embedded solutions and services focussing on quality standards and customer values.
- Offer end to end embedded solutions which ensure the best customer satisfaction.
- To build Skilled and Talented manpower pool for global industry requirements.
  To develop software and embedded products which are globally recognized.
- To become a global leader in Offering Scalable and cost-effective Software solutions and services across various domains like E-commerce, Banking, Finance, Healthcare and much more.
- To generate employment for skilled and highly talented youth of our Country INDIA.

Website	**http://www.karunadutechnologies.com**
Industry	Embedded Software Products
Headquarters	Bengaluru, karnataka
Type	Privately Held
Founded	2017

Specialties

Software Development, WebDesigning, DigitalMarketing, SKill Development, Machine Learning, Artificial Intelligence, Data Analytics, DataScience, Generative AI, Internet Of Things, and Full stack Web Application Development.

Location

#17, ATK complex, 4th Floor, Acharya College Main Road, Beside Karur Vysya Bank, Guttebasaveshwaranagar, Chikkabanvara, Bengaluru, Karnataka- 560090

Chapter 2

TASK PERFORMED

- As a part of internship program, we were provided with different tasks to perform which helped us to improve our technical and professional skills.
- Engage in a variety of tasks that enhance the understanding of programming, data manipulation, data handling, model development, and deployment.
- Project to develop a machine learning model with promising accuracy for diabetes prediction using logistic regression.

2.1: Learning Experience

- Gained hands-on experience with the machine learning development process.
- Learned about data collection, preprocessing, and feature engineering.
- Explored various machine learning algorithms for classification tasks.
- Developed and evaluated a model for predictions.
- During my internship at Karunadu Technologies, I had the opportunity to immerse myself in a dynamic and innovative environment that significantly enhanced my technical and professional skills.
- The internship at Karunadu Technologies was a transformative experience that provided me with a deeper understanding of machine learning, data science, and professional work ethics.

2.2: Knowledge Acquired

- Basics of python and features of python.
- Gained practical experience in the machine learning lifecycle: data collection, pre-processing, feature selection, model training, and evaluation.
- Deepened understanding of supervised learning algorithms, particularly logistic regression used in this project.
- Learned data visualization techniques for effective communication of project findings.
- Performance evaluation metrics (accuracy, precision, recall)

2.2.1: Introduction to Python

- Python is a high-level, interpreted programming language known for its readability, simplicity, and versatility.
- Python is a general purpose high level programming language.
- Python is a popular programming language.
- It was created by Guido van Rossum, and released in 1991. **2.2.2: Features of Python**

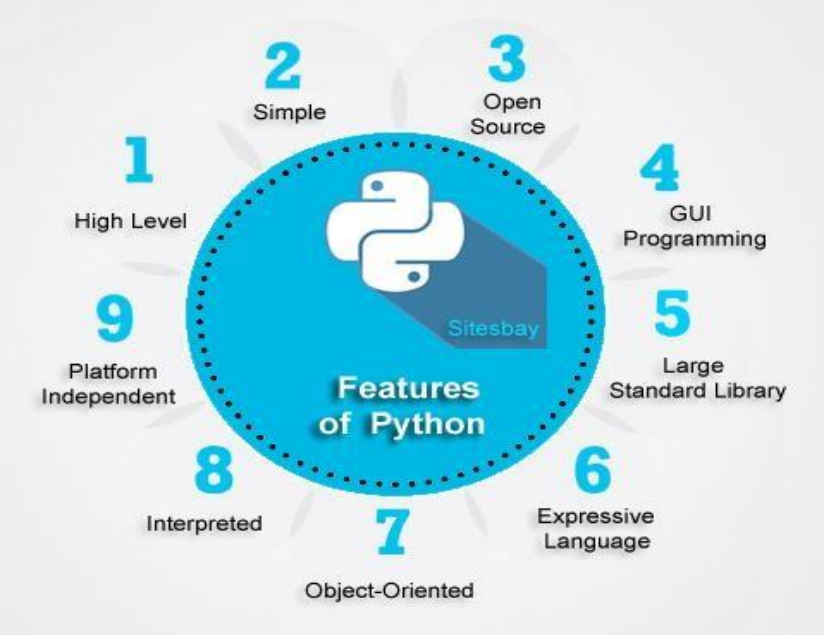

Fig 2.1 Features of Python

- **High-Level Language:**
  - ✧ Easy to read and write.
  - ✧ Abstracts away most of the complex details of the computer.

- **Interpreted:**
  - ✧ Executes code line-by-line.
  - ✧ No need for compilation, which makes debugging easier.
- **Dynamic Typing:**
  - ✧ Variable types are determined at runtime.
  - ✧ Flexible, but can lead to runtime errors if not managed properly.
- **Syntax and Readability:**
  - ✧ Uses indentation to define code blocks.
  - ✧ Emphasizes readability and simplicity.
- **Object-Oriented Programming (OOP)**:
  - ✧ Supports classes and objects.
  - ✧ Allows for encapsulation, inheritance, and polymorphism.
- **Modules and Libraries**:
  - ✧ Extensive standard library.
  - ✧ Large ecosystem of third-party libraries (e.g., NumPy, pandas, TensorFlow).
- **File I/O:**
  - ✧ Easy to read from and write to files using built-in functions.
- **Data Types:**
  - ✧ **Numbers:** Integers, floats, and complex numbers.
  - ✧ **Strings:** Text data enclosed in quotes.
  - ✧ **Lists:** Ordered, mutable collections.
  - ✧ **Tuples:** Ordered, immutable collections.
  - ✧ **Dictionaries:** Key-value pairs, unordered.
  - ✧ **Sets:** Unordered collections of unique items.

2.2.3: Applications of Python

- ◆ Web Development
- ◆ Data Science and Analytics
- ◆ Machine Learning and AI ◆ Automation and Scripting
- ◆ Scientific Computing
- ◆ Game Development
- ◆ Network Programming

- GUI Development
- Cybersecurity

2.2.4:Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) and computer science that focuses on the using data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy.

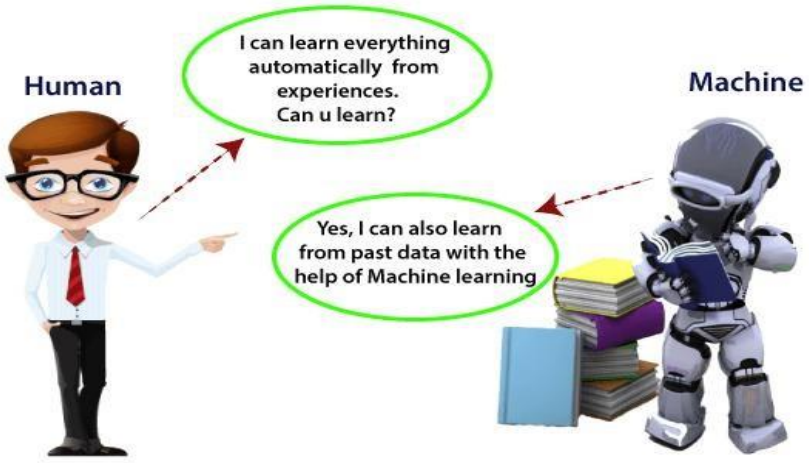

Fig 2.2 Machine Learning

2.2.5: Types of Machine Learning

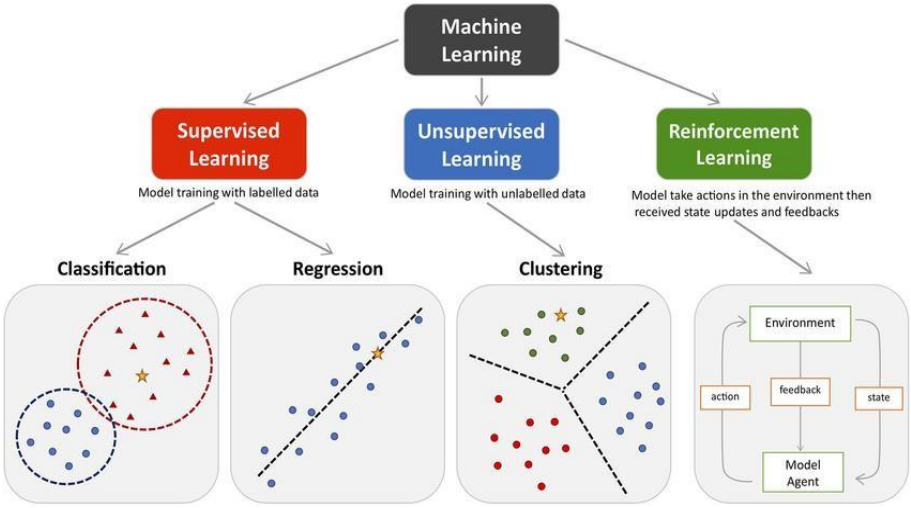

Fig 2.3 Types of Machine Learning

- **Supervised machine learning**

  Supervised learning, also known as supervised machine learning, is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately.

- **Unsupervised machine learning**

  Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets (subsets called clusters). These algorithms discover hidden patterns or data groupings without the need for human intervention.

- **Reinforcement machine learning**

  Reinforcement machine learning is a machine learning model that is similar to supervised learning, but the algorithm isn't trained using sample data. This model learns as it goes by using trial and error. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem.

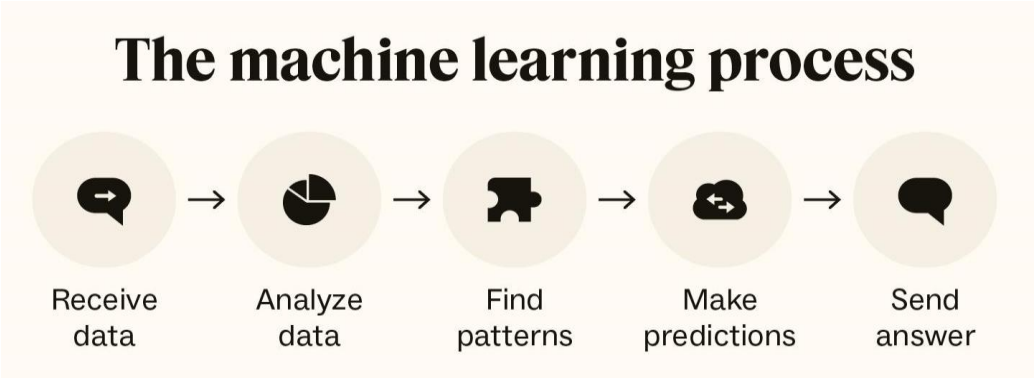

Fig 2.4 Machine Learning Process

2.2.6: Applications of Machine Learning

- Recommendation systems
- Social media connections
- Image recognition
- Natural language processing (NLP)
- Virtual personal assistants
- Stock market predictions
- Credit card fraud detection

2.2.7: Software Tools Used

- **Anaconda:** Free and open-source distribution of the Python programming language for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda.
- Anaconda distribution comes with more than 1,500 packages as well as the conda package and virtual environment manager.There are many applications available by default in navigator; among them is the Jupyter notebook.
- **Jupyter Notebook:** open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. Here are some key features and uses of Jupyter Notebook.
- **Machine Learning Libraries:** Depending on the project's needs, you might have used libraries like:
  - **Scikit-learn:** A popular library for classical machine learning algorithms, including logistic regression.
  - **TensorFlow or PyTorch:** These are powerful libraries for building and training deep learning models. While less common for simpler projects like diabetes prediction with logistic regression, they might be used for more complex healthcare applications.
- **Data Analysis Libraries**: Tools like Pandas (for data manipulation) and NumPy (for numerical computations) are essential for working with datasets.
- **Data Visualization Libraries:** Libraries like Matplotlib or Seaborn might have been used to create charts and graphs to explore the data and visualize model performance.
- **PyQt5:** PyQt5 is a library used for developing graphical user interfaces (GUIs) for Python applications.

2.3: Skills Learned

- **Technical:** Proficient in Python, building GUIs using PyQt5
- **Machine learning libraries**: Scikit-learn

- **Data Analytics and Visualization:** Pandas, NumPy, Matplotlib
  - Improved ability to analyze complex datasets and extract meaningful insights.
- **Project management:** Setting goals, meeting deadlines, and effective communication
- **Soft Skills:** Enhanced communication, teamwork, and problem-solving abilities.
- **Problem-Solving and Critical Thinking:**
  - Enhanced ability to break down complex problems into manageable tasks.
  - Improved logical thinking and problem-solving skills in the context of data analysis and machine learning.
- **Collaboration and Communication:**
  - Gained experience in working collaboratively on a project.
  - Developed skills in communicating technical findings effectively to both technical and non-technical audiences.

Chapter 3
CHALLENGING TASK PERFORMED

- **Data pre-processing:** Handling missing values and inconsistencies in the dataset.
- **Model Interpretation:** Analyzed feature importance and model coefficients to understand the influence of different features.
- **Statistical Analysis:** Performed descriptive statistics to understand data distribution and central tendencies.
- **Accuracy:** To measure the performance of the trained model.
- Experimented with various machine learning algorithms, including:
  - Logistic Regression
  - Linear Regression
  - Multiple Linear Regression
  - KNN Algorithm

3.1: Machine Learning Algorithms

Machine learning algorithms can be broadly categorized into three types: supervised learning, unsupervised learning, and reinforcement learning.

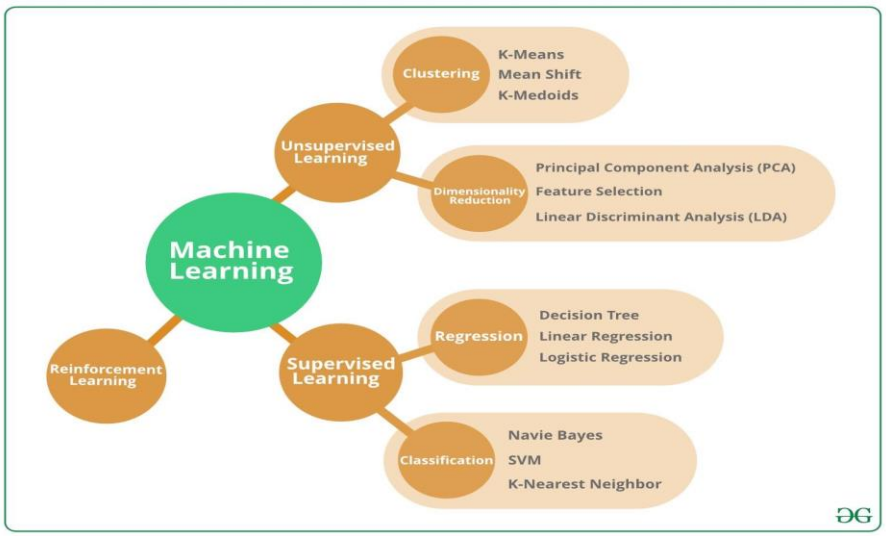

Fig 3.1 Machine Learning Algorithms

- These algorithms form the foundation of many machine learning applications, each with its strengths and suitable use cases.

3.1.1: Linear Regression

Linear regression is one of the simplest and most widely used algorithms in machine learning for predicting a continuous target variable based on one or more input features. It's used to model the relationship between a dependent variable (target) and one or more independent variables (features).

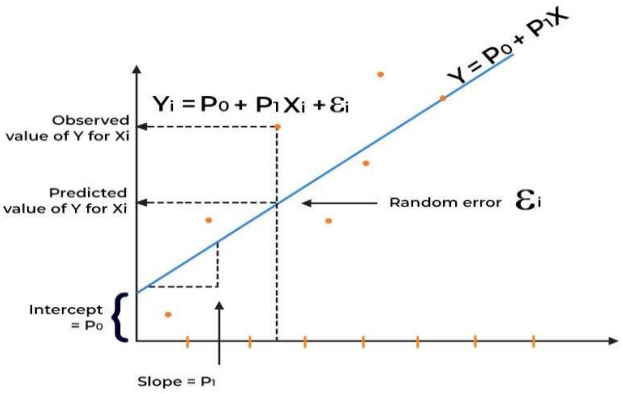

Fig 3.2 Linear Regression

This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:

$y=\beta 0+\beta 1X$

where:

- Y is the dependent variable
- X is the independent variable
- β0 is the intercept
- β1 is the slope

```
def area(self):
    import pandas as pd
    import matplotlib.pyplot as plt
    import sklearn
    from sklearn import linear_model
    path = "C:\\Users\\SHREYAS K\\OneDrive\\Desktop\\ML INTERN\\Data\\Data\\homeprices.csv"
    data = pd.read_csv(path)
    print(data)
    print(data.shape)
    inputdata = data.drop('price','columns')
    output = data.drop('area','columns')
    print(inputdata)
    print(output)
    plt.xlabel('Area')
    plt.ylabel('Price')
    plt.scatter(inputdata,output)
    model = linear_model.LinearRegression() #train
    model.fit(inputdata,output) #training
    result = model.predict([[self.txt1.text()]])
    self.lblmsg.setText(str(result))
```

Fig 3.3 Linear Regression Implementation

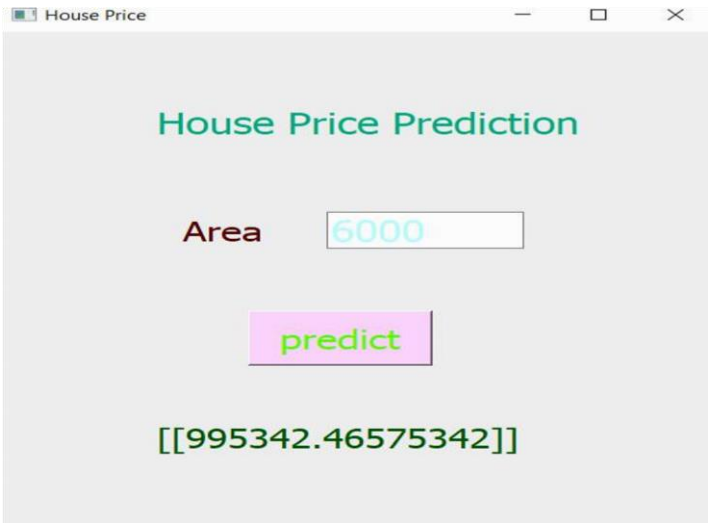

Fig 3.4 Linear Output

3.1.2: Multiple Linear Regression

This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is:

$y=\beta 0+\beta 1X+\beta 2X+\ldots\ldots\beta nX$

where:

- Y is the dependent variable
- X1, X2, …, Xp are the independent variables
- β0 is the intercept
- β1, β2, …, βn are the slopes

```
def profit(self):
    import pandas as pd
    import sklearn
    from sklearn import linear_model

    path = "C:\\Users\\SHREYAS K\\OneDrive\\Desktop\\ML INTERN\\Data\\Data\\50_Startups.csv"

    data = pd.read_csv(path)

    print(data)

    inputs = data.drop('Profit','columns')
    print(inputs)
    outputs = data.drop(['R&D Spend','Administration','Marketing Spend'],'columns')
    print(outputs)
    model = linear_model.LinearRegression()
    model.fit(inputs,outputs)
    result = model.predict([[self.txtrdspend.text(),self.txtadm.text(),self.txtmarketingspend.text()]])
    self.txtprofit.setText(str(result))
```

Fig 3.5 Multiple Linear Regression Implementation

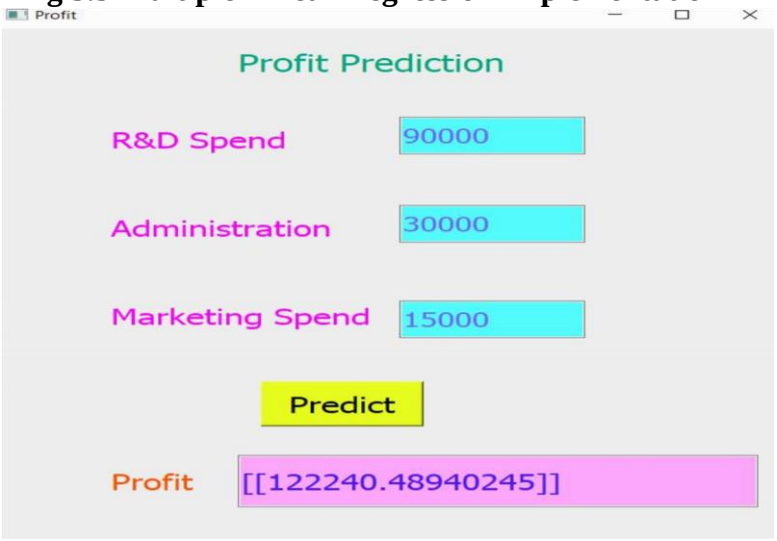

Fig 3.6 Multiple Linear Output

3.1.3: Logistic Regression

Logistic regression is a supervised machine learning algorithm used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors.

Logistic regression is used for binary classification where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.

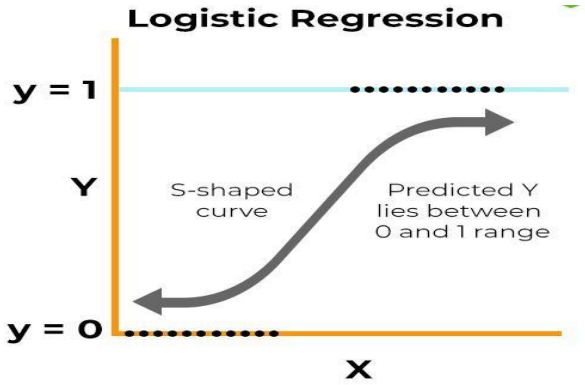

Fig 3.7 Logistic Regression

- · **Interpretability:** Logistic regression provides a clear understanding of how each feature influences the model's predictions. This is crucial in healthcare settings where understanding the rationale behind the model's decisions is important.
- · **Efficiency:** Logistic regression is computationally efficient, making it suitable for working with datasets of moderate size, which might have been the case for your internship project.

```
In [5]: import pandas as pd
path="C:\\Users\\Pooja\\Desktop\\syllabus\\sce\\2023_ISE_Project\\2023_ISE_Project\\Logistic\\Diabetes Prediction\\diabetes.csv"
data=pd.read_csv(path)
print(data)
print(data.info())
print(data.shape)
print(data.isnull().sum())
data=data.dropna()
print(data)
inputs=data.drop('Outcome','columns')
inputs
output=data.drop(["Pregnancies","Glucose","BloodPressure","SkinThickness","Insulin","BMI","DiabetesPedigreeFunction","Age"],axis=
output
import sklearn
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(inputs,output)
res=model.predict([[6,148,72,35,0,33.6,0.627,50]])
print("OUTCOME:",res)
acc=model.score(inputs,output)
print("ACCURACY:",acc*100)
```

Fig 3.8 Logistic Regression Implementation

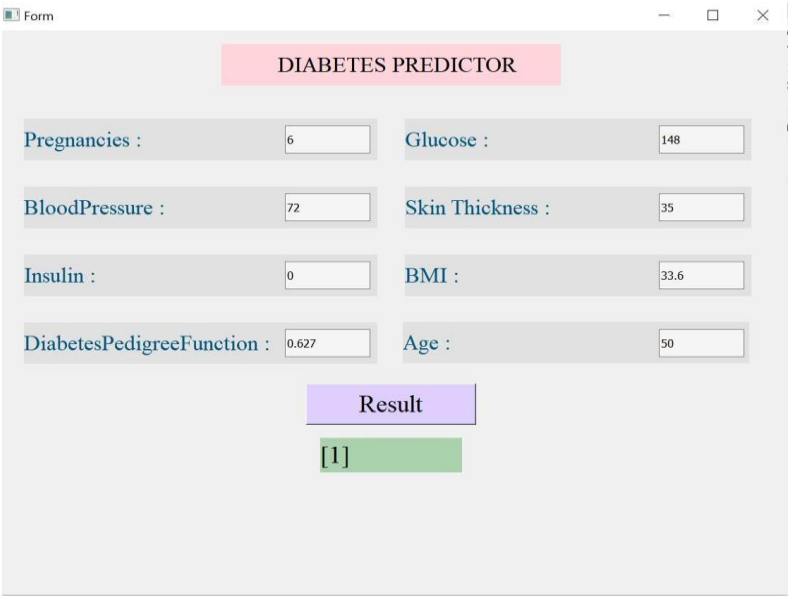

Fig 3.9 Logistic Output

3.1.4: KNN Regression

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

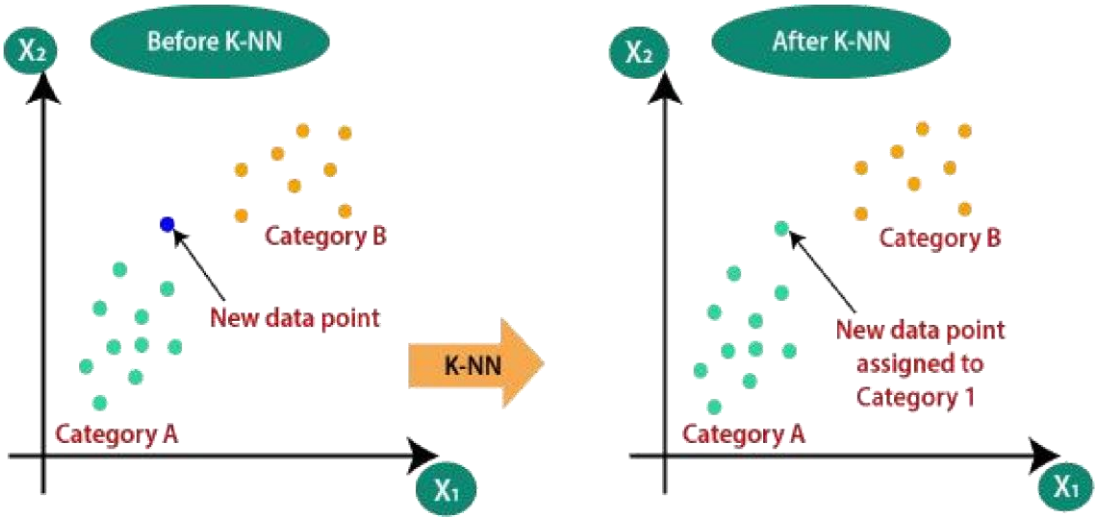

Fig 3.10 KNN

- **Store the Training Data:** The entire training dataset is used as a reference.
- **New Data Point:** When a new data point arrives for prediction (e.g., a new patient's data), the algorithm identifies its K nearest neighbors within the training data based on a chosen distance metric (e.g., Euclidean distance).
- **Majority Vote:** The class label (diabetic or non-diabetic in our case) that is most frequent among these K nearest neighbors is assigned as the predicted class for the new data point.

```
import pandas as pd
path="C:\\Users\\Pooja\\Desktop\\syllabus\\sce\\2023_ISE_Project\\2023_ISE_Project\\Logistic\\Magzine\\Kid.csv"
data=pd.read_csv(path)
print(data)
print(data.info())
print(data.isnull().sum())
print(data["Buy"].unique())
inputs=data.drop(["Obs No.", "Buy"],axis=1)
output=data.Buy
print(inputs)
print(output)
import sklearn
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(n_neighbors=13)
model.fit(inputs,output)
res=model.predict([[75000,1,1,1,1,0,0,15,1,0,1,1,1,1,1,0]])
print(res)
acc=model.score(inputs,output)
print(acc*100)
```

Fig 3.11 KNN Implementation

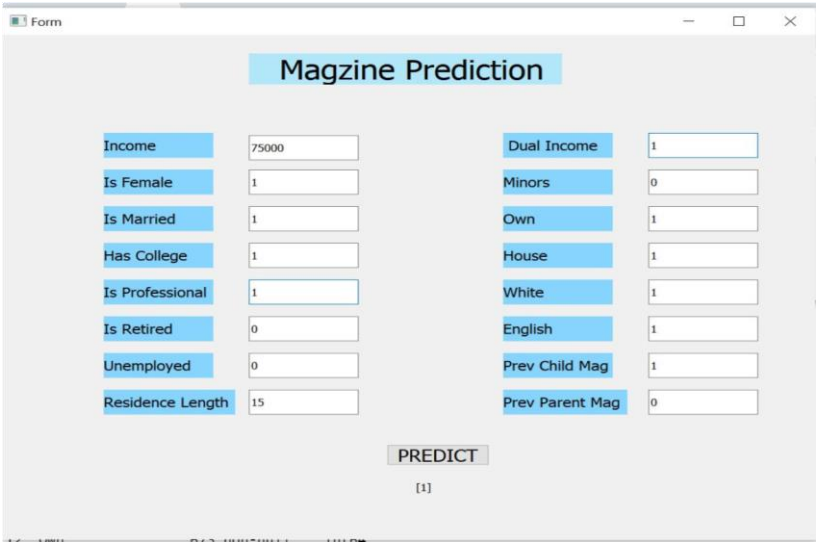

Fig 3.12 KNN Output

Chapter 4

PREDICT WHETHER THE CUSTOMER HAS SUBSCRIBED TO A TERM DEPOSIT OR NOT USING SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm primarily used for classification and regression tasks. SVM aims to find the optimal hyperplane that best separates different classes in the feature space.

- **Classification:** For classification tasks, SVM constructs a hyperplane or a set of hyperplanes in a high-dimensional space. The goal is to maximize the margin between the different classes, as a larger margin generally leads to a lower generalization error of the classifier. New data points are classified based on which side of the hyperplane they fall.

- **Regression:** For regression (Support Vector Regression or SVR), SVM uses the same principles to find a function that deviates from the true targets by no more than a specified epsilon, while keeping the function as flat as possible.

4.1: Problem Statement

4.1.1: This project aims to develop a predictive model using the Support Vector Machine (SVM) algorithm to determine whether a banking customer is likely to subscribe to a term deposit plan, thereby optimizing marketing strategies for telecallers by targeting individuals with a higher propensity to subscribe. The model utilizes a dataset comprising 16 input features, including demographic information like age, job, marital status, and education, alongside behavioral data such as balance, housing and personal loans, contact details, and past campaign interactions (day, month, duration, campaign, pdays, previous, poutcome), with the 'y' feature indicating the customer's subscription decision as the binary target.

Dataset:

You have a dataset containing the following attributes for a sample of individuals:

- **Age** (numeric): Age of the customer.
- **Job** (categorical): Type of job of the customer.
- **Marital** (categorical): Marital status of the customer.
- **Education** (categorical): The education level of the customer.
- **Default** (binary): Indicates if the customer has credit in default (yes/no).
- **Balance** (numeric): Average yearly balance (in Euros).
- **Housing** (binary): Indicates if the customer has a housing loan (yes/no).
- **Loan** (binary): Indicates if the customer has a personal loan (yes/no).
- **Contact** (categorical): Contact communication type.
- **Day** (numeric): Last contact day of the month.
- **Month** (categorical): Last contact month of the year.
- **Duration** (numeric): Last contact duration, in seconds.
- **Campaign** (numeric): Number of contacts performed during this campaign and for this client.
- **Pdays** (numeric): Number of days that passed after the client was last contacted from a previous campaign (-1 means the client was not previously contacted).
- **Previous** (numeric): Number of contacts performed before this campaign and for this client.
- **Poutcome** (categorical): Outcome of the previous marketing campaign.

age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y_bool
51	admin.	married	tertiary	no	148	no	no	cellular	15	apr	263	2	-1	0	unknown	0
55	blue-colla	married	secondary	no	7160	yes	no	unknown	4	may	315	1	-1	0	unknown	0
65	blue-colla	divorced	tertiary	no	2197	yes	no	cellular	12	may	102	2	-1	0	unknown	0
25	admin.	married	secondary	no	6658	yes	no	cellular	16	feb	197	1	-1	0	unknown	1
36	services	married	secondary	no	1761	yes	no	cellular	19	sep	177	1	-1	0	success	0
52	self-empl	single	tertiary	no	1887	yes	yes	unknown	19	jun	56	1	-1	0	unknown	1
22	admin.	single	secondary	no	2456	yes	yes	cellular	16	may	539	4	-1	0	unknown	1
31	technician	married	tertiary	no	87	no	no	cellular	11	apr	272	2	-1	0	unknown	1
25	blue-colla	married	secondary	no	3899	yes	yes	unknown	5	may	401	2	-1	0	unknown	0
30	technician	married	tertiary	no	228	yes	no	unknown	13	jun	107	1	-1	0	unknown	0
52	technician	single	tertiary	no	801	yes	yes	unknown	18	nov	552	10	-1	0	failure	0
57	blue-colla	single	secondary	no	510	yes	no	cellular	22	apr	64	7	-1	0	unknown	0
29	technician	single	secondary	no	209	yes	no	cellular	1	apr	212	2	214	0	failure	0
22	managem	single	tertiary	yes	1970	no	no	unknown	29	jul	18	1	-1	0	unknown	1
42	unknown	married	secondary	no	1197	yes	no	cellular	3	nov	334	1	-1	0	failure	0
39	services	married	tertiary	no	423	no	no	cellular	31	jul	107	1	-1	0	unknown	0
52	technician	divorced	tertiary	no	2666	no	yes	cellular	1	jul	154	2	-1	0	unknown	1
36	admin.	single	primary	no	2877	yes	no	unknown	14	may	46	1	-1	0	unknown	1
34	technician	married	secondary	yes	223	no	no	cellular	17	jul	188	1	-1	0	unknown	1
28	admin.	divorced	secondary	no	1288	no	no	cellular	10	nov	165	3	-1	0	unknown	0
34	admin.	married	secondary	no	760	no	no	cellular	17	feb	965	2	9	0	failure	0
60	services	married	secondary	no	671	no	no	cellular	10	mar	142	3	-1	0	unknown	1

Fig 4.1 Dataset

4.2: Algorithm

4.2.1: Working of Support Vector Machine(SVM)

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm that constructs a hyperplane or a set of hyperplanes in a high-dimensional space. The goal is to maximize the margin between the different classes, which helps in classifying new data points based on which side of the hyperplane they fall

Programming Steps:

Build an SVM model to predict the 'y' (term deposit subscription) variable based on the other features in the dataset. Evaluate the model's performance using appropriate metrics (e.g., accuracy, precision, recall, F1-score).

Steps to Approach:

1. **Data Preprocessing:** Handle missing values, encode categorical variables, and scale numerical features, which is crucial for SVM.
2. **Feature Selection:** Choose relevant features based on their importance in predicting term deposit subscription behavior.
3. **Model Training:** Implement the SVM algorithm, selecting an appropriate kernel function and tuning hyperparameters like C and gamma for optimal performance.
4. **Model Evaluation:** Assess the model's performance using cross-validation and metrics such as accuracy, precision, recall, and F1-score.

4.3: Implementation

4.3.1: Code

```
import pandas as pd
path="C:/Users/Admin/Downloads/2025_projects/2024_projects/19_SubscriberPrediction/train.csv"
data=pd.read_csv(path)
print(data)
print(data.info())
print(data.isnull().sum())
```

```
       age          job   marital  education default  balance housing loan  \
0       51       admin.   married   tertiary      no      148      no   no
1       55  blue-collar   married  secondary      no     7160     yes   no
2       65  blue-collar  divorced   tertiary      no     2197     yes   no
3       25       admin.   married  secondary      no     6658     yes   no
4       36     services   married  secondary      no     1761     yes   no
...    ...          ...       ...        ...     ...      ...     ...  ...
20995   26   technician   married  secondary      no      289      no   no
20996   34   management   married   tertiary      no      122     yes   no
20997   32  blue-collar   married   tertiary     yes      521      no  yes
20998   40   technician   married  secondary      no     2092      no   no
20999   48   technician    single    primary      no      509      no   no

         contact  day month  duration  campaign  pdays  previous poutcome  \
0       cellular   15   apr       263         2     -1         0  unknown
1        unknown    4   may       315         1     -1         0  unknown
2       cellular   12   may       102         2     -1         0  unknown
3       cellular   16   feb       197         1     -1         0  unknown
4       cellular   19   sep       177         1     -1         0  success
```

```
import pandas as pd
path="C:/Users/Admin/Downloads/2025_projects/2024_projects/19_SubscriberPrediction/train.csv"
data=pd.read_csv(path)
print(data)
print(data.info())
print(data.isnull().sum())
print(data["y_bool"].unique())
```

Fig 4.2 Code

```
from sklearn.preprocessing import LabelEncoder
le_job=LabelEncoder()
data["job"]=le_job.fit_transform(data["job"])
le_marital=LabelEncoder()
data["marital"]=le_marital.fit_transform(data["marital"])
le_education=LabelEncoder()
data["education"]=le_education.fit_transform(data["education"])
le_default=LabelEncoder()
data["default"]=le_default.fit_transform(data["default"])
le_housing=LabelEncoder()
data["housing"]=le_housing.fit_transform(data["housing"])
le_loan=LabelEncoder()
data["loan"]=le_loan.fit_transform(data["loan"])
le_contact=LabelEncoder()
data["contact"]=le_contact.fit_transform(data["contact"])
le_month=LabelEncoder()
data["month"]=le_month.fit_transform(data["month"])
le_poutcome=LabelEncoder()
data["poutcome"]=le_poutcome.fit_transform(data["poutcome"])
le_y_bool=LabelEncoder()
data["y_bool"]=le_y_bool.fit_transform(data["y_bool"])

print(data)
inputs=data.drop(["y_bool"],axis=1)
output=data.y_bool
print(input)
```

Fig 4.3 Sklearn code

```
from sklearn.svm import SVC
model=SVC()
model.fit(inputs,output)
res=model.predict([[51,0,1,2,0,148,0,0,0,15,0,263,2,-1,0,3]])
print(res)

acc=model.score(inputs,output)
print(acc*100)
```

```
C:\Users\Admin\anaconda3\ANAC\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
```

```
[0]
73.0904761904762
```

Fig 4.4 Outcome

4.3.2: Result

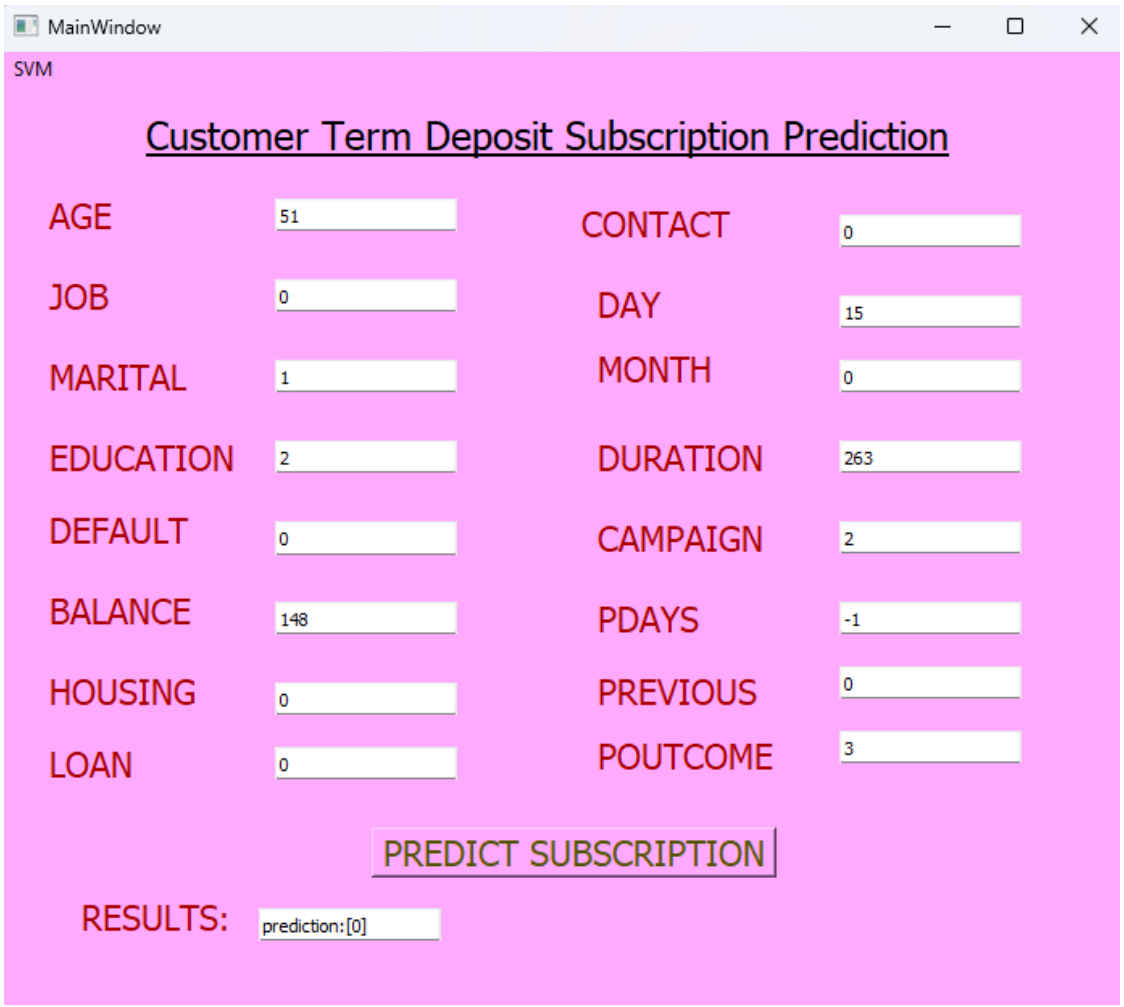

Fig 4.5 Output

REFLECTION

During my internship at Karunadu Technologies, Bangalore, I had the opportunity to work on a significant project focused on Prediction of whether the customer has subscribed to a term deposit or not using Support Vector Machine. This project involved several key stages: data preprocessing, model training, evaluation, and interpretation.This internship project was a transformative experience. It not only provided me with hands-on experience in machine learning but also deepened my understanding of how these technologies can be applied to solve real-world problems. The skills and knowledge I gained will be invaluable as I continue my journey in the field of data science and machine learning. Reflecting on this project, I feel more confident in my ability to tackle complex problems, apply machine learning techniques, and contribute meaningfully to future projects. I am grateful for the guidance and support from my mentors at Karunadu Technologies, which has been instrumental in my learning and growth

CONCLUSION

The internship provided a valuable opportunity to apply Python programming and Machine Learning theory to solve real-time, complex problems. It helped in identifying appropriate prediction models and suitable learning algorithms for future use, significantly enhancing programming skills and practical knowledge. In this project, predicting whether a customer has subscribed to a term deposit using the Support Vector Machine (SVM) algorithm presented both opportunities and challenges. SVM's ability to find an optimal hyperplane for classification makes it highly effective, especially in high-dimensional spaces or when clear separation boundaries exist between classes. Its robustness against overfitting and good generalization capabilities, particularly with carefully selected kernel functions and optimized hyperparameters, are significant advantages. However, SVM's performance can be limited by its computational intensity for very large datasets, sensitivity to the choice of kernel and regularization parameters (C, gamma), and the difficulty in interpreting the model's decision-making process compared to simpler algorithms.

BIBLIOGRAPHY

- https://www.geeksforgeeks.org/understanding-logistic-regression/
- https://www.geeksforgeeks.org/support vector machine/
- https://www.karunadutechnologies.com/
- https://www.geeksforgeeks.org/ml-linear-regression/

\