TABLE OF CONTENTS

CHAPTER – 1

INTRODUCTION

1.1Background:

Given the modern world's exponential development of data, data science has become a critical area that helps businesses make data-driven decisions and derive meaningful insights. Python is a popular programming language for data science because of its adaptability and ease of use. It is renowned for its simplicity and strong libraries.

The mission of the Government of India's National Small Industries Corporation Limited (NSIC) is to assist and develop small-scale industries throughout the nation. NSIC, which is situated in Sector B-24 of the Guindy Industrial Estate in Ekkaduthangal, Chennai 600032, is essential in helping small businesses become more competitive by facilitating technology upgrades, offering training, and providing a range of services.

Gaining practical experience in Python programming and data science during an internship at NSIC was a unique opportunity, especially when it came to aiding small-scale enterprises. This report details the experiences, learned skills, and projects worked on during the internship.

1.2Objectives:

- ➢ To acquire proficiency in Python programming:
- ➢ To understand data science processes:
- ➢ To implement machine learning algorithms:
- ➢ To work on real-life data science projects:
- ➢ To enhance professional skills:
- ➢ To understand the application of data science in small industries:

CHAPTER – 2

ORGANIZATION PROFILE

2.1 Overview of NSIC:

एन एस आई सी
N S I C
National Small Industries Corporation Ltd. (NSIC)

It was Founded in 1955 to encourage and assist the expansion of MSMEs in India, the National Small Industries Corporation Limited (NSIC) is a government agency under the Ministry of Micro, Small, and Medium Enterprises (MSME). The goal of NSIC is to improve MSMEs' sustainability and competitiveness by offering integrated support services in marketing, technology, finance, and other domains. Its goal is to promote sustainable and equitable economic growth.

NSIC provides a range of services, such as technological support through incubation centers and technical training, credit help through schemes and facilitation with financial institutions, and marketing support through consortia and online portals. It also offers infrastructural support through NSIC Business Parks, international consulting, and export promotion assistance.

NSIC has a main office in New Delhi and runs a network of branch and regional offices throughout India. The Chennai office plays a crucial role in assisting MSMEs in the southern area by offering training, technical assistance, and advisory services to improve their growth and competitiveness. It is situated at Sector B-24, Guindy Industrial Estate, Ekkaduthangal, Chennai - 600032.

CHAPTER – 3

INTERNSHIP OVERVIEW

3.1 Training Program and Schedule:

In NSIC, our training program started on the 29th of December, 2023 to the 29th of January, 2024. The Daily Schedule consists of a 3 hours session from morning 9:30 AM to 12:30 PM, in which the first 15 days of the program began with learning the basics of Python and further enrolling ourselves into Data related operations such as cleaning the data, surpassing the data, methods, and libraries for appropriate data.

Then the program was overtaken by Machine Learning basics and fundamental algorithms, which paved the path for testing and creation of various projects and ideas.

SCHEDULE:

Day 1 – Day 15 = Python Basics

Day 16 – Day 25 = Data Processing and Machine Learning

Day 26 – Day 30 = Capstone Project

CHAPTER – 4

PYTHON BASICS

4.1 Basics of Python:

Python is a high-level, interpreted programming language known for its simplicity and versatility. It is widely used in various fields such as web development, automation, data analysis, artificial intelligence, and scientific computing. During my internship at National Small Industries Corporation Limited (NSIC), I gained a solid foundation in Python programming, which is crucial for data science applications.

Python was created by Guido van Rossum and first released in 1991. Its design philosophy emphasizes code readability and simplicity, making it an ideal language for beginners and professionals alike. Python's syntax is clear and concise, which allows developers to write less code compared to other programming languages like Java or C++.

Key Features of Python

Easy to Read and Write: Python’s syntax is designed to be intuitive and its code resembles pseudo-code. This simplicity reduces the learning curve and enhances productivity.

Interpreted Language: Python code is executed line by line, which makes debugging easier and development faster.

Dynamically Typed: Python handles type checking at runtime, allowing more flexibility in coding.

Extensive Standard Library: Python comes with a rich standard library that supports many common programming tasks, such as file I/O, system calls, and web development.

Cross-Platform Compatibility: Python runs on various platforms, including Windows, macOS, and Linux.

Python Basics

Variables and Data Types

Python supports several data types including integers, floats, strings, lists, tuples, dictionaries, and sets. Variables in Python are dynamically typed, meaning you don’t need to declare their type explicitly.

```
# Examples of variable assignments

integer_var = 10

float_var = 10.5
```

```python
string_var = "Hello, NSIC!"
list_var = [1, 2, 3, 4, 5]
tuple_var = (1, 2, 3, 4, 5)
dict_var = {'name': 'Python', 'version': 3.8}
set_var = {1, 2, 3, 4, 5}
```

Control Structures

Python provides several control structures to manage the flow of the program, such as conditional statements and loops.

Conditional Statements:

```python
x = 10
if x > 5:
    print("x is greater than 5")
elif x == 5:
    print("x is equal to 5")
else:
    print("x is less than 5")
```

Loops:

```python
# For loop
for i in range(5):
    print(i)
```

 # While loop

```python
count = 0
while count < 5:
    print(count)
    count += 1
```

Functions

Functions in Python are defined using the def keyword and can have parameters and return values.

```
def greet(name):
    return f"Hello, {name}!"
print(greet("NSIC"))
```

Modules and Packages

Python's modularity allows you to organize your code into modules and packages. A module is a file containing Python code, and a package is a collection of modules.

```
# Importing a module
import math
print(math.sqrt(16))
# Importing specific functions from a module
from math import pi, e
print(pi, e)
```

File Handling

Python provides built-in functions to read from and write to files.

```
# Writing to a file
with open('example.txt', 'w') as file:
    file.write("Hello, NSIC!")
# Reading from a file
with open('example.txt', 'r') as file:
    content = file.read()
    print(content)
```

4.2 Libraries and Frameworks

During my internship, I also learned to use various Python libraries and frameworks that are essential for data science, such as:

NumPy: For numerical operations and handling arrays.

Pandas: For data manipulation and analysis.

Matplotlib, Seaborn: For data visualization.

Scikit-learn: For machine learning algorithms and models.

CHAPTER – 5

DATA SCIENCE ESSENTIALS

5.1 Anaconda Navigator:

Anaconda Navigator is a desktop GUI that comes with the Anaconda distribution, designed for scientific computing, data science, and machine learning. It simplifies package management and deployment, making it easy to manage environments, install packages, and launch essential applications like Jupyter Notebook, Spyder, RStudio, and Visual Studio Code. Users can create, manage, and switch between environments effortlessly, ensuring isolated project dependencies and minimizing conflicts. It streamlines the installation, updating, and removal of packages from repositories like Anaconda Cloud and PyPI. To use Anaconda Navigator, install the Anaconda distribution, launch the Navigator from the start menu or applications folder, and manage environments through the "Environments" tab. This tool enhances productivity and simplifies workflows, making it indispensable for data scientists and researchers.

5.2 Python 3.9:

As of this writing, the most recent significant release of Python is Python 3.9, made available on October 5, 2020. It enhances readability, maintainability, and efficiency with new features and optimizations. During my internship training in data science at National Small Industries Corporation Limited (NSIC), I worked with Python 3.9, which provided a strong foundation for data science projects.

Python 3.9 introduces key features like dictionary merge and update operators (| and |=), which simplify merging and updating dictionaries. New string methods (str.removeprefix and str.removesuffix) facilitate easier string manipulation. Improvements to type hinting, including typing. Annotated and generic types with built-in collections, enhance readability and maintainability in large codebases.

Performance improvements in Python 3.9 include optimized garbage collection, speeding up cycles, and reducing memory overhead for objects with __slots__. Standard library modules like math, statistics, and zone info have been enhanced. The asyncio module has also seen improvements, aiding scalable and efficient asynchronous code for I/O-bound tasks.

These enhancements made developing data science projects easier during my course. New dictionary operators and enhanced type hinting improved code clarity and maintenance. Standard library and asynchronous programming improvements provided powerful tools for complex data processing tasks.

Python 3.9 represents a significant advancement with new features, performance enhancements, and improved maintainability. My experience with Python 3.9 at NSIC has equipped me to leverage these developments in future projects, ensuring effective and efficient data science solutions.

5.3 Jupyter Notebook:

Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text, making it an indispensable tool in data science, machine learning, and scientific research. During my internship training in Python with Data Science at National Small Industries Corporation Limited (NSIC), Jupyter Notebook facilitated the exploration, analysis, and presentation of data efficiently. This interactive computing environment supports real-time code execution, which is crucial for testing and iterating on code with immediate visibility of results. It integrates rich media such as text, images, videos, and interactive plots, enhancing the documentation and presentation of the analysis process. While primarily associated with Python, Jupyter Notebook supports over 40 programming languages through various kernels, offering flexibility in coding. It also excels in data visualization by integrating with libraries like Matplotlib, Seaborn, and Plotly, allowing for the creation of high-quality, interactive visualizations. Additionally, Jupyter Notebooks are easily shareable and publishable in multiple formats, promoting collaboration and making it straightforward to distribute and present work. By combining code, data, and narrative text in a single document, Jupyter Notebooks ensure reproducibility, which is crucial for scientific research and data-driven projects. The use of Jupyter Notebook during my internship significantly enhanced the efficiency and clarity of my data science tasks, making it a vital component of my learning and development.

CHAPTER – 6

DATA HANDLING

6.1 Collection:

During my internship training in Python with Data Science at National Small Industries Corporation Limited (NSIC), data collection was a fundamental aspect of the data analysis process. Data collection involves systematically gathering and measuring information on variables of interest to answer research questions, test hypotheses, and evaluate outcomes. Effective data collection is critical for building accurate models, performing robust analyses, and deriving meaningful insights. There are two primary sources of data: primary data, which is collected directly through methods such as surveys, interviews, observations, and experiments, and secondary data, which is obtained from existing sources like databases, government publications, industry reports, and scientific journals. For primary data, surveys and questionnaires are useful for gathering large amounts of information on opinions and behaviors, while interviews provide in-depth insights through direct engagement. Observational methods capture behaviors and interactions in natural settings, and experimental methods allow for the manipulation of variables to establish cause-and-effect relationships. Secondary data collection leverages existing data, saving time and resources but requiring careful evaluation for relevance and reliability. During my internship, I employed various data collection tools and techniques to ensure the accuracy and completeness of the data, which was essential for subsequent data analysis and decision-making processes.

6.2 Cleaning:

Data cleaning, also known as data cleansing or data scrubbing, is a crucial step in the data analysis process that involves detecting and correcting errors or inconsistencies in data to improve its quality. During my internship training in Python with Data Science at National Small Industries Corporation Limited (NSIC), I learned that clean, accurate data is essential for building reliable models and deriving meaningful insights. Real-world data is often messy, containing inaccuracies, duplicates, missing values, and inconsistencies, which can significantly affect the performance of data analysis and machine learning models. Data cleaning ensures that the data is consistent, accurate, and ready for analysis, leading to better decision-making and more accurate predictions. The process involves several key steps: handling missing values by either removing records, imputing values, or using predictive modeling; removing duplicate entries to ensure that each data point is unique; and correcting inconsistencies in data formats, such as standardizing date formats or ensuring consistent capitalization. Through my internship, I gained hands-on experience with various data cleaning techniques and tools, which are essential for preparing high-quality data for analysis and achieving reliable results in data science projects.

6.3 Visualization:

Data visualization is a critical component of data science that involves representing data visually through charts, graphs, and plots. Throughout my internship training in Python with Data Science at National Small Industries Corporation Limited (NSIC), I recognized the significance of data visualization in effectively conveying insights from complex datasets. By presenting data visually, through techniques like bar charts, line plots, scatter plots, and heatmaps, stakeholders can quickly grasp trends, patterns, and relationships within the data. This visual representation facilitates better understanding and communication of key findings, enabling informed decision-making. Moreover, data visualization techniques allow for the identification of outliers, correlations, and anomalies that may not be apparent from raw data alone, enhancing the accuracy and depth of data analysis. Leveraging various visualization tools and libraries in Python, such as Matplotlib, Seaborn, and Plotly, I gained practical experience in creating informative and compelling visualizations to support data-driven insights and recommendations. Overall, data visualization serves as a powerful tool for extracting meaningful insights from data and driving business success in today's data-driven world.

CHAPTER – 7

MACHINE LEARNING

Machine learning is a subset of artificial intelligence (AI) that enables computers to learn from data and make predictions or decisions without being explicitly programmed. During my internship training in Python with Data Science at National Small Industries Corporation Limited (NSIC), I gained hands-on experience in applying machine learning algorithms to analyze and extract insights from datasets. Machine learning has vast applications across various industries, including finance, healthcare, marketing, and manufacturing, offering opportunities to automate processes, optimize operations, and drive innovation.

Key Concepts of Machine Learning

Supervised Learning: Supervised learning involves training a model on labeled data, where each example is paired with the correct output. The goal is to learn a mapping from inputs to outputs, enabling the model to make predictions on new, unseen data. Common supervised learning algorithms include linear regression, logistic regression, decision trees, and support vector machines.

Unsupervised Learning: Unsupervised learning deals with unlabeled data, where the algorithm must discover patterns and structures on its own. Clustering and dimensionality reduction are common tasks in unsupervised learning. K-means clustering, hierarchical clustering, and principal component analysis (PCA) are popular unsupervised learning techniques.

Deep Learning: Deep learning is a subfield of machine learning that focuses on neural networks with multiple layers. Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are capable of learning intricate patterns from large amounts of data. They excel in tasks such as image recognition, natural language processing, and speech recognition.

Steps in the Machine Learning Workflow

Data Preprocessing: This step involves cleaning and preparing the data for analysis, including handling missing values, encoding categorical variables, and scaling numerical features.

Feature Engineering: Feature engineering aims to create new features or transform existing ones to improve the performance of machine learning models. Techniques include feature scaling, dimensionality reduction, and creating interaction terms.

Model Selection and Training: In this step, different machine learning algorithms are evaluated and trained on the dataset. Hyperparameter tuning and cross-validation are used to select the best-performing model.

Model Evaluation: The trained model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score. This step assesses the model's ability to generalize to new, unseen data.

Deployment and Monitoring: Once the model is trained and evaluated, it can be deployed into production for making predictions on new data. Continuous monitoring and updating of the model are essential to ensure its performance remains optimal over time.

Machine Learning Libraries

Machine learning libraries are essential tools for implementing various algorithms, models, and techniques in Python. During my internship training on Python with Data Science at National Small Industries Corporation Limited (NSIC), I had the opportunity to work with several prominent machine learning libraries that facilitate the development and deployment of machine learning solutions.

1. Scikit-learn (sklearn)

Scikit-learn is a widely used machine learning library that provides simple and efficient tools for data mining and data analysis. It features a wide range of supervised and unsupervised learning algorithms, including classification, regression, clustering, dimensionality reduction, and model selection. Scikit-learn is known for its user-friendly interface, extensive documentation, and versatility, making it an ideal choice for beginners and experienced practitioners alike.

2. TensorFlow

TensorFlow is an open-source machine learning framework developed by Google Brain for building and training deep learning models. It offers a flexible architecture that allows developers to deploy machine learning models across a variety of platforms, from desktops to mobile devices to cloud servers. TensorFlow supports both high-level APIs, such as Keras for easy model building, and low-level APIs for more advanced customization and optimization.

3. PyTorch

PyTorch is another popular open-source machine learning library that is widely used for building deep learning models. Developed by Facebook's AI Research lab, PyTorch offers dynamic computation graphs, which enable more flexibility and ease of use compared to static computation graphs. PyTorch is known for its intuitive interface, seamless integration with Python, and strong community support.

4. Keras

Keras is a high-level neural networks API written in Python and capable of running on top of TensorFlow, Microsoft Cognitive Toolkit (CNTK), or Theano. It is designed for fast experimentation with deep neural networks and provides a simple and consistent interface for building and training models. Keras allows for easy prototyping of deep learning architectures and supports both convolutional and recurrent neural networks.

5. XGBoost

XGBoost is an optimized, distributed gradient-boosting library designed for speed and performance. It is widely used in machine learning competitions and is known for its accuracy and efficiency in handling large datasets. XGBoost implements parallelization and cache optimization techniques to speed up training and achieve better predictive performance.

CHAPTER – 8

PROJECTS

8.1 Cardiovascular and Distribution:

During my internship training in Python with Data Science at National Small Industries Corporation Limited (NSIC), I had the opportunity to work on an impactful project focused on analyzing cardiovascular data and its distribution. This project aimed to leverage data science techniques to gain insights into cardiovascular health patterns, identify risk factors, and predict the likelihood of cardiovascular diseases (CVD).

Project Overview

The Cardiovascular and Distribution project involved the collection, cleaning, analysis, and visualization of a comprehensive dataset containing various health-related attributes of individuals. The primary objective was to explore the distribution of cardiovascular conditions within the population and to develop predictive models to assess the risk of cardiovascular diseases.

Data Collection and Preprocessing

The dataset used for this project was sourced from a public health database, which included information such as age, gender, blood pressure, cholesterol levels, body mass index (BMI), smoking status, and other relevant health indicators. The data collection phase involved downloading and integrating multiple datasets to ensure a comprehensive analysis.

Data preprocessing was a critical step in preparing the dataset for analysis. This included handling missing values, removing duplicates, and normalizing the data. Additionally, we performed feature engineering to create new variables that could enhance the predictive power of our models. For instance, we derived new features such as age group categories and BMI ranges.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to understand the underlying patterns and distributions within the data. We used various visualization techniques to examine the relationships between different variables and the occurrence of cardiovascular diseases. Key findings from the EDA included:

Age and Cardiovascular Risk: The analysis revealed a positive correlation between age and the likelihood of developing cardiovascular conditions. Older individuals showed higher incidences of CVD.

Cholesterol Levels: Elevated cholesterol levels were significantly associated with increased cardiovascular risk.

Smoking and Blood Pressure: Smokers and individuals with high blood pressure exhibited a higher probability of cardiovascular issues.

Predictive Modeling

The next phase involved building predictive models to estimate the risk of cardiovascular diseases. We employed several machine learning algorithms, including logistic regression, decision trees, random forests, and gradient boosting. The models were trained and validated using cross-validation techniques to ensure robustness and generalizability.

Model Evaluation

The performance of the predictive models was evaluated using various metrics such as accuracy, precision, recall, F1-score, and the area under the Receiver Operating Characteristic (ROC) curve. Among the models, the gradient boosting algorithm achieved the highest accuracy and provided the best balance between sensitivity and specificity.

Visualization and Insights

To effectively communicate the findings, we created a series of visualizations that highlighted key insights from the analysis. These included:

Distribution Plots: Visualizing the distribution of cardiovascular conditions across different age groups and cholesterol levels.

Heatmaps: Illustrating the correlation between various health indicators and cardiovascular risk.

ROC Curves: Comparing the performance of different predictive models.

OUTPUT FORMS:

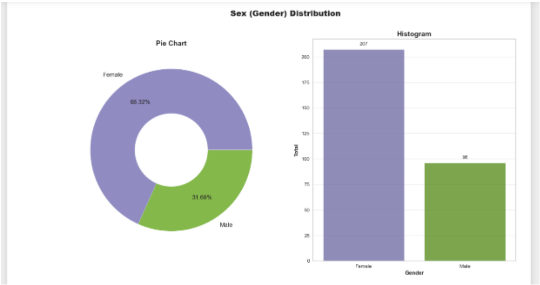

FIGURE 8.1.1

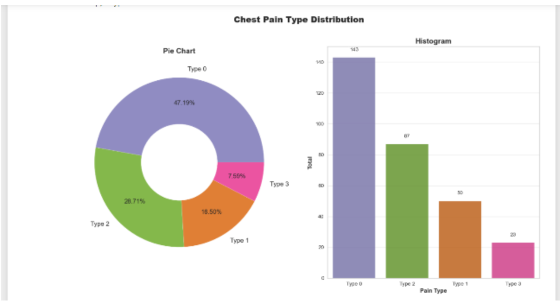

FIGURE 8.1.2

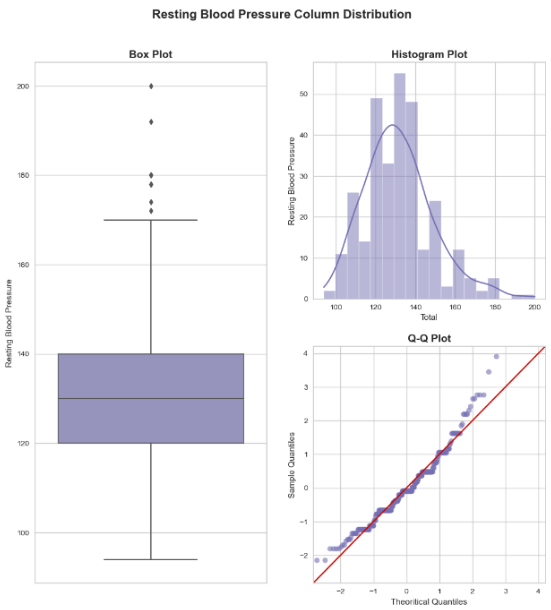

FIGURE 8.1.3

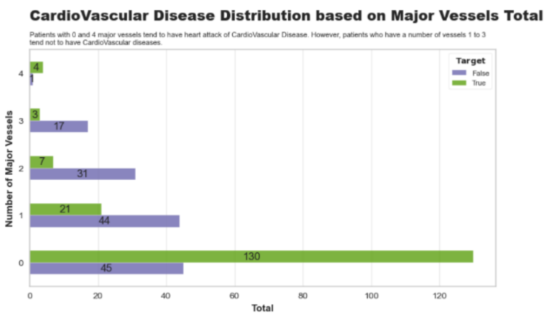

FIGURE 8.1.4

8.2 Heart Problem Prediction:

During my internship training in Python with Data Science at National Small Industries Corporation Limited (NSIC), I had the opportunity to work on a significant project focused on predicting heart problems. The Heart Problem Prediction project aimed to utilize data science techniques and machine learning algorithms to predict the likelihood of individuals developing heart-related issues based on various health indicators.

Project Overview

The objective of the Heart Problem Prediction project was to develop a predictive model that could accurately identify individuals at high risk of heart problems. This involved collecting, preprocessing, and analyzing a comprehensive dataset containing numerous health-related attributes. By leveraging machine learning, the project sought to provide actionable insights that could aid in the early diagnosis and prevention of heart diseases.

Data Collection and Preprocessing

The dataset for this project was obtained from a publicly available health database, encompassing a wide range of variables such as age, sex, blood pressure, cholesterol levels, blood sugar levels, electrocardiographic results, maximum heart rate, exercise-induced angina, and other relevant medical conditions. The initial step involved thorough data cleaning to address missing values, remove duplicates, and ensure consistency.

Feature engineering played a crucial role in this phase. New features were derived to enhance the predictive power of the model. For instance, the ratio of cholesterol to HDL (high-density lipoprotein) and the difference between resting and maximum heart rate were calculated to provide deeper insights into heart health.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to understand the relationships and patterns within the data. Various visualization techniques were employed to identify correlations and trends among the variables. Key findings from the EDA included:

Age and Heart Problems: Older individuals exhibited a higher prevalence of heart problems.

Cholesterol Levels: Elevated cholesterol levels were strongly associated with increased heart disease risk.

Exercise-Induced Angina: The presence of exercise-induced angina was a significant indicator of heart problems.

Predictive Modeling

Several machine learning algorithms were employed to build the predictive model, including logistic regression, decision trees, random forests, support vector machines (SVM), and gradient boosting. The dataset was split into training and testing sets to evaluate the performance of each model. Cross-validation techniques were used to ensure the robustness and generalizability of the models.

Model Evaluation

The models were evaluated using various performance metrics such as accuracy, precision, recall, F1-score, and the area under the Receiver Operating Characteristic (ROC) curve. Among the tested algorithms, the random forest model achieved the highest accuracy and demonstrated a strong balance between precision and recall, making it the most effective model for predicting heart problems.

Visualization and Insights

To effectively communicate the results, a series of visualizations were created, including:

Confusion Matrix: Displaying the performance of the predictive models in terms of true positives, true negatives, false positives, and false negatives.

ROC Curve: Illustrating the trade-off between sensitivity and specificity across different thresholds for each model.

Feature Importance Plot: Highlighting the most significant features contributing to the prediction of heart problems.

OUTPUT FORMS:

Out[12]: <AxesSubplot:xlabel='sex', ylabel='target'>

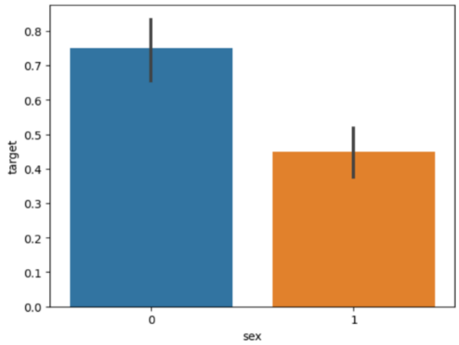

FIGURE 8.2.1

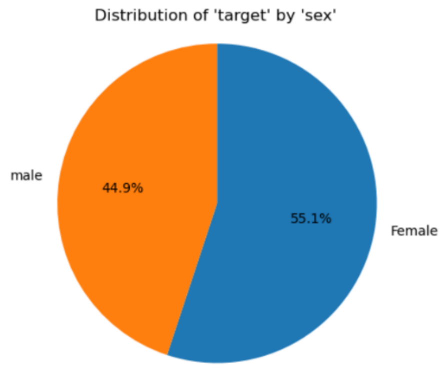

FIGURE 8.2.2

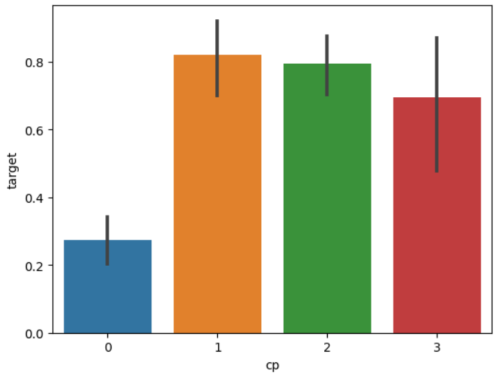

FIGURE 8.2.3

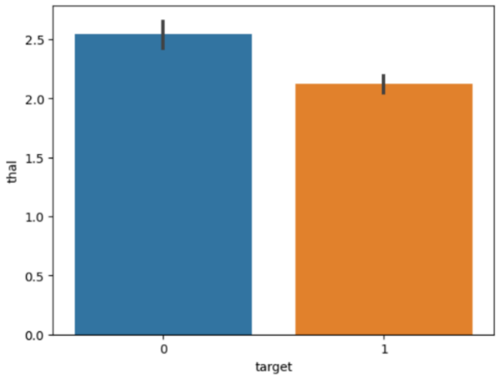

FIGURE 8.2.4

CHAPTER – 9

CONCLUSION

During my internship at National Small Industries Corporation Limited (NSIC), I gained a strong foundation in Python with Data Science, which has been a useful experience. By gaining practical experience with Python 3.9 and important libraries like TensorFlow, PyTorch, I was able to improve my machine learning and data analysis abilities.

I gained insight into data gathering, cleansing, analysis, and visualization by applying theoretical knowledge to real-world situations through projects like the Cardiovascular and Distribution project and the Heart Problem Prediction project. The workflow was streamlined by Anaconda Navigator's assistance and with jupyter notebook with effective package and environment management.

Overall, I feel that my experience at NSIC has greatly improved my data science capabilities and given me the knowledge and practical skills I will need for my future profession. I am appreciative of the chance to develop in such a fast-paced setting and am excited to use what I've learned to tackle data science problems in the real world.

CERTIFICATE

No. : A1050

राष्ट्रीय लघु उद्योग निगम लिमिटेड

एन एस आई सी

N S I C

THE NATIONAL SMALL INDUSTRIES CORPORATION LIMITED

(A Govt. of India Enterprise)

NSIC - TECHNICAL SERVICES CENTRE

Sector B-24, Guindy Industrial Estate, Ekkaduthangal, Chennai - 600 032.

CERTIFICATE

This is to certify that Ms. **SHERIN M.** D/o. Shri. **MARIYA JOSEPH S.** student of III year B.Tech.(IT), Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, has undergone **"Internship Training on PYTHON WITH DATA SCIENCE"** conducted at our centre for a period of one month from 29.12.2023 to 29.01.2024.

HEAD OF TRAINING

HEAD OF CENTRE