**Student Engagement in Innovative Ways – FFCS Report**

A TECHNICAL PROJECT REPORT

*submitted by*

**Ananth Bharathwaj T A**

**21BCE1079**

**MICROSOFT INNOVATIONS CLUB – VIT CHENNAI**

**** 

# **Acknowledgments:**

I would like to express my sincere gratitude to my training officers, M. Braveen and Anusha K, for their guidance and support throughout the process of completing this thesis/dissertation. They were always available to answer my questions, provide me with feedback, and help me to stay on track. I would also like to sincerely thank the club leadership, AI-ML department leadership for giving me this opportunity to learn. I am deeply grateful for their mentorship and encouragement. I would also like to thank Google Colabratory, Stackoverflow, Geekeforgeeks and other such online resources that helped me learn more skills in this domain.

# **Abstract:**

This report summarizes the work I have done on four AI/ML tasks assigned to me as part of the Microsoft Innovations Club. The tasks were:

1. An overview of Pandas
2. An overview of Machine Learning using Scikit-learn and a group project to apply ML models on a Parkinson’s Dataset
3. Credit Score Classification Problem
4. Image Classification Problem

I have demonstrated a strong understanding of the fundamentals of Pandas, Scikit-learn, and machine learning. I have also applied my skills to solve real-world problems, such as predicting Parkinson's disease and classifying credit scores and images.

In particular, I developed a machine learning pipeline for building a neural network model for a classification task using TensorFlow and scikit-learn. I also trained and evaluated three different Convolutional Neural Network (CNN) models on the Fashion MNIST dataset, and visualized the training and validation performance of these models.

I am proud of the work I have done, and I believe that it is a valuable contribution to the Microsoft Innovations Club and the field of artificial intelligence. I hope that my work will help to inspire and educate other students about the potential of AI and machine learning.

Contents

[**Acknowledgment:** 1](#_Toc148716058)

[**Abstract:** 1](#_Toc148716059)

[**Introduction:** 3](#_Toc148716060)

[**Overview:** 3](#_Toc148716061)

[**Process:** 3](#_Toc148716062)

[**Work Assigned:** 3](#_Toc148716063)

[**Work Description:** 3](#_Toc148716064)

[**Task 1** 3](#_Toc148716065)

[**Task 2** 4](#_Toc148716066)

[**Task 2A** 4](#_Toc148716067)

[**Task 2B** 4](#_Toc148716068)

[**Task 3** 5](#_Toc148716069)

[**Task 4** 5](#_Toc148716070)

[**Conclusion:** 5](#_Toc148716071)

# **Introduction:**

This report summarizes the work done by the AI/ML team of the Microsoft Innovations Club. The team was assigned four tasks and five seminars/workshops. The tasks were:

* An overview of Pandas
* An overview of Machine Learning using Scikit-learn and a group project to apply ML models on a Parkinson’s Dataset
* Credit Score Classification Problem
* Image Classification Problem

# **Overview:**

Under the AI-ML team, the group was assigned 4 tasks and 5 seminars/workshops. The seminars included a web development workshop, Microsoft Student Ambassador introduction, and others.

# **Process:**

Each task was supplied with problem statements, context and resources for further learning and implementation. Each problem was then studied, researched and deployed on a Google Colab project.

# **Work Assigned:**

There were 4 tasks assigned for the AI-ML team.

1. An overview of Pandas
2. A. An overview of Machine Learning using Scikit-learn  
   B. A group project to apply ML models on a Parkinson’s Dataset
3. Credit Score Classification Problem
4. Image Classification Problem

# **Work Description:**

## **Task 1**

A pandas practice sheet was given, as a tutorial to the workings of the library. The content is as below:

1. Importing Libraries: The notebook begins by importing essential Python libraries, such as `pandas` for data manipulation.

2. Loading Data: The dataset, presumably related to car sales, is loaded into a DataFrame.

3. Data Exploration:

- The `head()` function is used to display the first few rows of the DataFrame.

- Basic statistics of the dataset, such as count, mean, and standard deviation, are computed using the `describe()` function.

4. Data Cleaning:

- The column names are standardized by converting them to lowercase.

- The 'Price' and 'Odometer' columns are cleaned to remove non-numeric characters and convert the data to numeric types.

- Missing data is handled, including changing 'NaN' values and removing rows with missing values.

5. Adding Columns:

- A new column 'Seats' is added to the DataFrame.

6. Data Analysis:

- Further exploration and analysis of the data are likely performed, but the details are not provided in the document. The document ends with the addition of the 'Seats' column.

## **Task 2**

This task was split into two parts: 2a – individual and 2b - group

### **Task 2A**

This task is a tutorial to machine learning via the scikit-learn library of python. The notebook contains the following:

1. Introduction to Scikit-Learn: The notebook starts with an introduction to scikit-learn and its importance in machine learning.

2. Data Preparation:

- It covers loading and exploring a dataset, in this case, a heart disease dataset.

- Data preprocessing techniques like handling missing data, converting categorical variables to numeric form, and scaling the data are explained.

3. Model Building:

- A logistic regression model is built to predict heart disease presence.

- The dataset is split into training and testing sets.

- The model is trained on the training data, and evaluation metrics such as accuracy, precision, recall, F1-score, and ROC curve are discussed.

4. Hyperparameter Tuning:

- Hyperparameters of the logistic regression model are optimized using grid search and cross-validation.

5. Confusion Matrix and Classification Report:

- The concepts of a confusion matrix and a classification report are explained, showing how to evaluate the model's performance in more detail.

6. Cross-Validation:

- The notebook introduces the idea of cross-validation for robust evaluation of machine learning models.

- The scikit-learn `cross\_val\_score()` function is used to calculate cross-validated accuracy for the logistic regression model.

7. Cross-Validated Metrics:

- Cross-validated accuracy, precision, recall, and F1-score are calculated and discussed.

This notebook provides a practical demonstration of working with scikit-learn, including data preprocessing, model building, hyperparameter tuning, and evaluating model performance. It also emphasizes the importance of cross-validation for more robust model evaluation.

### **Task 2B**

This task was assigned to a team of 3: Iltimas Kabir, Shradda Pandey and Ananth Bharathwaj T A. The problem statement is as follows:

“The Parkinson dataset includes 195 biomedical records of people with 23 varied characteristics. The idea behind this project is to design an ML model that can differentiate between healthy people and those suffering from Parkinson’s disease. The model uses the XGboost (extreme gradient boosting) algorithm based on decision trees to make the separation.”

The submitted notebook was as described below:

1. Data Import and Preprocessing: The document begins with the importation of necessary libraries and the dataset. Data preprocessing steps, such as handling missing values, are carried out to prepare the dataset for analysis.

2. Exploratory Data Analysis (EDA): This section involves exploring the dataset to gain insights into its characteristics. Descriptive statistics, data visualization, and feature engineering are performed.

3. Machine Learning Models:

- Logistic Regression: The author trains a logistic regression model to predict a binary outcome. The accuracy and confusion matrix of the model on the test data are presented.

- XGBoost: An XGBoost classifier is used, and its performance metrics, including accuracy and classification report, are reported.

- Random Forest Classifier: A random forest classifier is trained and evaluated with accuracy, classification report, and confusion matrix.

- Support Vector Machine (SVM): An SVM model is used for classification, and its accuracy, classification report, and confusion matrix are presented.

4. Comparative Analysis of Models: A summary concludes the document, highlighting that, out of the three machine learning models (Logistic Regression, XGBoost, and Random Forest), Random Forest performed the best on this dataset. However, SVM performed the worst in terms of classification accuracy.

## **Task 3**

This code appears to be a machine learning pipeline for building a neural network model for a classification task using TensorFlow and scikit-learn. Here's a summary of the code:

1. Import necessary libraries including Pandas, NumPy, TensorFlow, scikit-learn, and Matplotlib.

2. Mount Google Drive to access a dataset located in a Google Drive directory.

3. Load a dataset from a CSV file using Pandas. It issues a DtypeWarning for column 26 with mixed data types.

4. Encode categorical columns using LabelEncoder.

5. Handle outliers in numeric columns using the IQR (Interquartile Range) method to replace values that fall below or above a certain threshold.

6. Scale the numeric columns using Min-Max scaling.

7. Merge numeric and categorical columns into X and assign the target variable 'Credit\_Score' to y.

8. Use one-hot encoding to encode the second column (presumably the 'Name' column) in the dataset.

9. Create a random subset of the data (20000 samples) and split it into training and validation sets using `train\_test\_split`.

10. Handle missing values using SimpleImputer with the strategy of filling missing values with the mean of a random sample.

11. Apply TruncatedSVD (Singular Value Decomposition) to reduce the dimensionality of the input data, reducing it to 1000 components.

12. Define a neural network model using Keras, consisting of multiple dense layers with ReLU activation functions and L2 regularization.

13. Compile the model using sparse categorical cross-entropy loss and the Adam optimizer with a learning rate of 0.0001.

14. Train the model on the training data for 20 epochs with a batch size of 64, tracking accuracy and validation metrics.

15. The training history is printed, showing loss and accuracy for each epoch.

16. Additionally, it generates a visualization of the model architecture and saves it to a file named "model.png" using `plot\_model`.

The model training results show that it starts with relatively low accuracy but improves over the epochs. The model is also subject to overfitting as the training accuracy is significantly higher than the validation accuracy.

You can further analyze and improve this model by fine-tuning hyperparameters, handling class imbalance if necessary, and possibly exploring different neural network architectures to achieve better results.

## **Task 4**

The provided task is training and evaluating three different Convolutional Neural Network (CNN) models on the Fashion MNIST dataset, and then visualizing the training and validation performance of these models. Here's a summary of the code:

1. Import TensorFlow and relevant modules, such as Sequential, Dense, Conv2D, MaxPooling2D, and matplotlib.pyplot.
2. Load the Fashion MNIST dataset using fashion\_mnist.load\_data(). This dataset contains grayscale images of clothing items.
3. Normalize the pixel values of both the training and test images to a range between 0 and 1 by dividing by 255.0.
4. Define three different CNN models (model1, model2, and model3) with increasing complexity:
5. model1 consists of one convolutional layer with 32 filters, a max-pooling layer, and two fully connected layers. This model is relatively simple.
6. model2 extends model1 by adding another convolutional layer with 64 filters and max-pooling.
7. model3 further extends model2 by adding a third convolutional layer with 128 filters and an additional fully connected layer.
8. Compile each model using the Adam optimizer and sparse categorical cross-entropy as the loss function.
9. Train each model on the training data (images and labels) for 10 epochs while validating on the test data. The training history is stored in history1, history2, and history3 for each respective model.
10. Define a function plot\_history to plot and visualize the training and validation loss and accuracy for each model.
11. Call plot\_history for each model to create separate plots showing loss and accuracy for each of the three models.

# **Conclusion:**

In this report, I have summarized the work I have done on the four tasks assigned to me by the AI/ML team of the Microsoft Innovations Club. I have demonstrated a strong understanding of the fundamentals of Pandas, Scikit-learn, and machine learning. I have also applied my skills to solve real-world problems, such as predicting Parkinson's disease and classifying credit scores and images.

I am proud of the work I have done, and I believe that it is a valuable contribution to the Microsoft Innovations Club and the field of artificial intelligence. I hope that my work will help to inspire and educate other students about the potential of AI and machine learning.

I would like to thank the AI/ML team for their guidance and support. I have learned a great deal from them, and I am grateful for the opportunity to have worked on these challenging and rewarding tasks.