**PERSONALITY PREDICTION**

**A MINI PROJECT REPORT**

**18CSC305J - ARTIFICIAL INTELLIGENCE**

### Submitted by

**DOMA BHARGAV(RA2011028010069)**

*Under the guidance of*

**DR. Vaishnavi Moorthy**

### Assistant Professor, Department of Computer Science and Engineering

### in partial fulfillment for the award of the degree of

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**



#### S.R.M. Nagar, Kattankulathur, Chengalpattu District

**MAY 2023**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

**BONAFIDE CERTIFICATE**

Certified that Mini project report titled **“Personality Prediction”** is the bona fide work of **Doma Bhargav(RA2011028010069)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

|  |  |
| --- | --- |
| Dr. Vaishnavi Moorthy  **GUIDE**  Assistant Professor  Department of Data Science and Business System | Dr. M. Lakshmi  **HEAD OF THE DEPARTMENT**  Professor & Head  Department of Data Science and Business System |

**ABSTRACT**

#### Personality prediction is an area of research that aims to identify and understand individual differences in human behavior, thought patterns, and emotions. It involves the use of various tools and methods, including psychological assessments, behavioral observation, and computational modeling, to predict and describe a person's unique personality traits and characteristics. This field has applications in a wide range of domains, including clinical psychology, organizational behavior, and social psychology. Understanding and predicting personality can lead to insights into how individuals interact with others, make decisions, and cope with stress and adversity. With the development of social networks, a large variety of approaches have been developed to define users' personalities based on their social activities and language use habits. Lately, there has been a massive spike up in the number of social network users. There is a massive evolution of social media platforms which have led to massive data generation. With this available data, there are large variations of methods to define personality of the users’ based on their social behavior and patterns. With the aid of machine learning model and data-sets the main aim of this paper is to predict the Myers–Briggs type Indicator (MBTI) personality type of the twitter user. The Myers-Briggs type Indicator is probably the maximum widely used personality check within the world. The predictor will help to predict 1 of 16 different personalities of the user based on their Twitter account. The text is preprocessed to get clean. After tokenizing of the data, machine learning model - LSTM (Long Short Term Memory Networks) has been built. The predictor endeavors to get of the personality type, traits and careers suitability.

#### 

**TABLE OF CONTENTS**

[**ABSTRACT**](#_2et92p0) **iii**

**TABLE OF CONTENTS iv**

[**LIST OF FIGURES**](#_3dy6vkm) **v**

[**ABBREVIATIONS**](#_4d34og8) **vi**

[**1 INTRODUCTION**](#_17dp8vu) **7**

[**2 LITERATURE SURVEY**](#_3j2qqm3) **8**

**3** [**SYSTEM A**](#_1y810tw)**RCHITECTURE AND DESIGN 9**

3.1 Architecture diagram of personality prediction 9

3.2 Description of Module and components10

**4 METHODOLOGY 14**

4.1 Problem Formulation 14

4.2 Problem Definition

4.3 Scope

4.4 Proposed Methodology

4.5 Implementation

**5 CODING AND TESTING 15**

**6 SCREENSHOTS AND RESULTS**

6.1 Buzzer and PIR 19

6.2 Ultrasonic and Scarecrow 19

6.3 Soil moisture sensor with water pump 20

6.4 LCD Screen 20

6.5 Whole Circuit 21

6.6 Thingspeak Server 22

**7 CONCLUSION AND FUTURE ENHANCEMENT 23**

7.1 Conclusion

7.2 Future Enhancement

**REFERENCES 24**

**LIST OF FIGURES**

##### [3.1.1](#_4i7ojhp) Architecture block 9

[3.2.1](#_2xcytpi) **Soil moisture sensor** 10

3.2.3  [**Temperature Sensor (DHT 11)**](#_1ksv4uv)  10

[3.2.4](#_1ci93xb) **PIR Motion Sensor**  10

[3.2.5 **Motor**](#_3whwml4) 11

* + 1. **Ultrasonic Sensor**  11

3.2.7 [**Buzzer**](#_2bn6wsx)  12

3.2.8 **WIFI MODULE ESP 8266**  13

[6.1.1](#_qsh70q) **Buzzer and PIR simulation on tinkercad** 19

[6.2.1](#_3as4poj) **Ultrasonic sensor and Scarecrow simulation on tinkercad** 19

[6.3.1](#_1pxezwc) **Soilmoisture sensor and motor simulation on tinkercad** 20

[6.4.1](#_49x2ik5) **LCD Screen with output** 20

* + 1. **Whole Circuit** 21

6.6.1 **Screenshot of thingspeak server** 22

v

**ABBREVIATIONS**

**IOT** Internet of Things

**PIR** Passive Infrared

**LCD** Liquid Crystal Diode

**DHT** Distributed hash table

**IR** Infra red

**UART** Universal Asynchronous Receiver/Transmitter

**IDE** Integrated Development Environment

**V****i**

**CHAPTER 1**

**INTRODUCTION**

The Myers–Briggs type Indicator (MBTI) is an introspective self-report questionnaire indicating diverging predilections in how people understand the world and make choices. The recommended system stipulates the Myers–Briggs kind Indicator with a view to categorize the character types in sixteen patterns through 4 dichotomies, specifically, (1) Introversion - Extroversion, (2) Sensing - Intuition, (3) Thinking - Feeling and (4) Judging - Perceiving. A primary alphabet from each category or dimension will be taken into consideration to achieve a four letter test case, for instance, ISTP or ENTP or INFJ. The 4 letter case helps to acquire the correct outcomes. Each kind is stated to outline a specific set of behavioral dispositions, reflecting variations in attitudes, orientation, and choice-making patterns.

As an example, the individual could be appropriate for the job position of Analysts, if the output class of 4 letter case might be acquired as INTJ - a personality type which describes people as innovative and strategic thinkers, with a plan for everything. The MBTI personality type is predicted using Deep Learning model (DL) - LSTM (Long Short Term Memory Networks). It is difficult to categorize the personality and job suitability of the person, so the intention of the system is to annihilate this barrier and give out the personality types throughout 4 dichotomies which suggest the psychological preferences and aids to provide job suitability of the person.

The Big Five Personality Traits model is based on findings from several independent researchers, and it dates back to the late 1950s. But the model as we know it now began to take shape in the 1990s. Lewis Goldberg, a researcher at the Oregon Research Institute, is credited with naming the model "The Big Five." It is now considered to be an accurate and respected personality scale, which is routinely used by businesses and in psychological research.

The Big Five Personality Traits Model measures five key dimensions of people's personalities:

* Openness: sometimes called "Intellect" or "Imagination," this measures your level of creativity, and your desire for knowledge and new experiences.
* Conscientiousness: this looks at the level of care that you take in your life and work. If you score highly in conscientiousness, you'll likely be organized and thorough, and know how to make plans and follow them through. If you score low, you'll likely be lax and disorganized.
* Extraversion/Introversion: this dimension measures your level of sociability. Are you outgoing or quiet, for instance? Do you draw energy from a crowd, or do you find it difficult to work and communicate with other people?
* Agreeableness: this dimension measures how well you get on with other people. Are you considerate, helpful and willing to compromise? Or do you tend to put your needs before others'?
* Natural Reactions: sometimes called "Emotional Stability" or "Neuroticism," this measure emotional reactions. Do you react negatively or calmly to bad news? Do you worry obsessively about small details, or are you relaxed in stressful situations?

**CHAPTER 2**

**LITERATURE SURVEY**

In [1], personality was predicted by using NLP, CNN, LSTM, LSG algorithms. Firstly each word was embedded into the word vector and encoded in the first part. In the second part, 2 LSTM (Long Short Term Networks) and CNN (Convolutional Neural Network) are used to learn the structural features of the output from the first part. In the final part concept of LSG (Latent Science Group) is used, the output from the second part is sent into softmax to produce the personality traits as the final result.

The quantity of human beings and the usage of social media has skyrocketed in recent years. In this context [3], social media furnished researchers with a wealth of facts about user and societal conduct. They were starting to realize how a person’s conduct on social media pertains to their personalities. Conventional personality assessments rely upon self-document inventories that are expensive to accumulate. These studies tried to predict a user’s big-five character based on information obtained from social networks. They administered a large-five personality stock exam to 131 Sina Weibo users and extracted all of their Weibo messages and profile facts. The authors correctly expected the big-5 personality of customers by means of investigating the relevance between all forms of consumer produced facts and personal results of customers.

They extracted traits which include consumer behavior, interplay conduct and textual content language conduct. Authors used Pearson Correlation Coefficient to select functions based totally on dependency metrics concept. They used machine learning knowledge of algorithms to expect rankings of personalities with extracted capabilities.On this paper, Logistic Regression and Na¨ıve Bayes algorithms were used to get the large-5 personalities. They decided on 5 maximum relative dimensions and applied a machine learning algorithm to know the approach to correctly forecast the big-five personalities of users.

They [4] acquire social data and questionnaires from Weibo users and concentrate on how to leverage user text information to predict personality traits. The authors used correlation analysis and principal component analysis to choose the user data, and then used the multiple regression model, grey prediction model, and multitasking model to predict and evaluate the outcomes.

The grey prediction’s MAE values were found to be better than the Multitask model’s multiple regression model, with the total effect of the prediction ranging between 0.8 and 0.9, showing good prediction accuracy. The grey prediction model exhibited superior prediction accuracy than the other two regression models, according to the MAE prediction index.

In [5],user behavior at the facebook social networking web page became used to make personality predictions. With the emergence of social networks, a slew of latest strategies for figuring out people’s personalities based totally on their social activities and linguistic styles have arisen. Machine Learning algorithms, records sources, and feature sets fluctuate between methodologies. The intention of this examination turned into to peer how well massive five version characteristics and metrics might expect facebook customers’ persona traits. Tokenization became employed to do away with URLs, symbols, names, and lowercase letters. They inferred a person’s traits from functions related to a person’s social network.. The Pearson correlation analysis turned into hired as the standard function selection method to quantify the power of the linear dating among two variables and to take a look at capabilities widespread for personality trait prediction. The effects of the prediction accuracy tests tested that the personality prediction system based at the XGBoost classifier outperformed the common baseline for all characteristic units, with the very best prediction accuracy of seventy 74.2 Percentage. With a prediction accuracy of 78.6 Percentage, the character social community evaluation functions set accomplished the exceptional prediction overall performance for the extraversion feature. information pre-processing, feature extraction, and function choice are all steps inside the method.

**CHAPTER 3**

**SYSTEM ARCHITECTURE AND DESIGN**

**3.1 Architecture diagram of personality prediction**

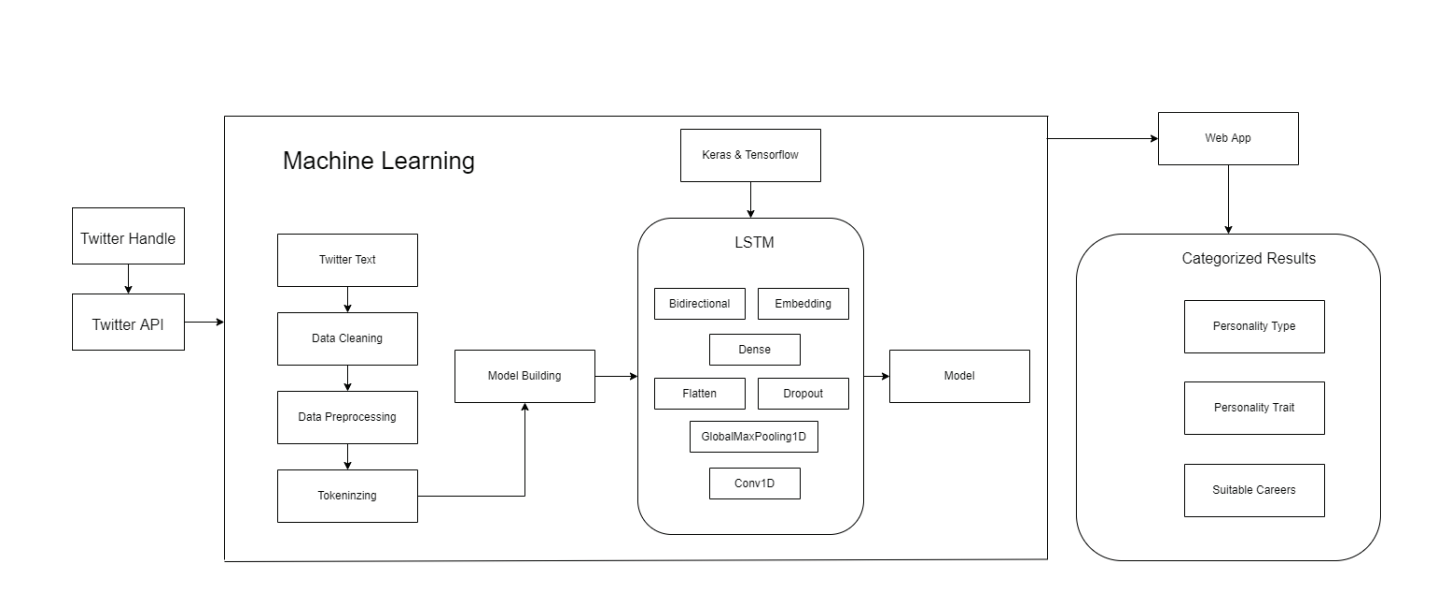
****

FIG 3.1.1: ARCHITECTURE DIAGRAM

**3.2 Description of Module and components**

1. **Data Analysis:-** For data analysis, it includes following python libraries such as pandas, numpy, pickle and scipy. Each library has different usage:

* Pandas: Pandas is a data manipulation library that provides data structures for efficiently storing and manipulating large datasets.
* NumPy: NumPy is a library for numerical computing in Python. It provides fast and efficient functions for mathematical operations on large arrays and matrices.
* Pickle: Pickle is a Python module for serializing and de-serializing Python objects.
* SciPy: SciPy is a library for scientific computing in Python. It provides functions for optimization, interpolation, signal processing, linear algebra, and statistics.

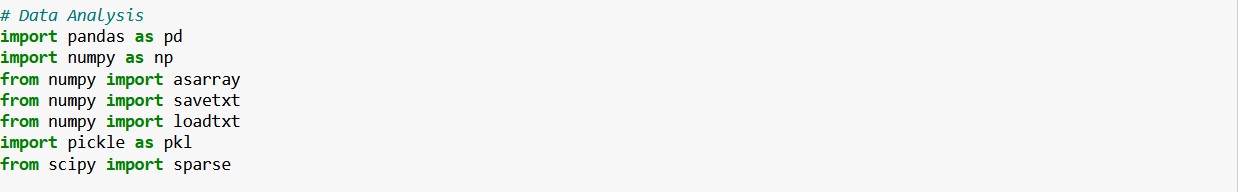
****

FIG 3.2.1: Data Analysis

1. **Data Visualization:-** This module includes libraries for data visualization such as matplotlib, seaborn, wordcloud. Each of them has different ways of visualizing the data**.**

* Matplotlib: Matplotlib is a popular data visualization module for Python that enables the creation of various types of graphs, including line charts, bar charts, scatter plots, histograms, and heatmaps.
* Seaborn: Seaborn is another Python data visualization module that is built on top of Matplotlib.
* Wordcloud: Wordcloud is a data visualization module that is specifically designed for visualizing text data.

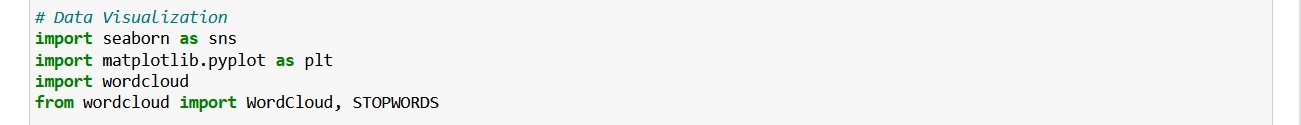
****

FIG 3.2.2: Data Visualization

1. **Text Processing:-** The modules you mentioned are commonly used in Python programming for a variety of purposes, including data analysis and natural language processing. Here is a brief description of each:

* `re` module: The `re` module is used for working with regular expressions in Python.
* `itertools` module: The `itertools` module provides a collection of tools for working with iterators in Python.
* `string` module: The `string` module provides a collection of constants and functions for working with strings in Python.
* `collections` module: The `collections` module provides a collection of specialized data structures in Python.
* `nltk` module: The `nltk` (Natural Language Toolkit) module is a collection of tools for working with natural language processing tasks in Python.

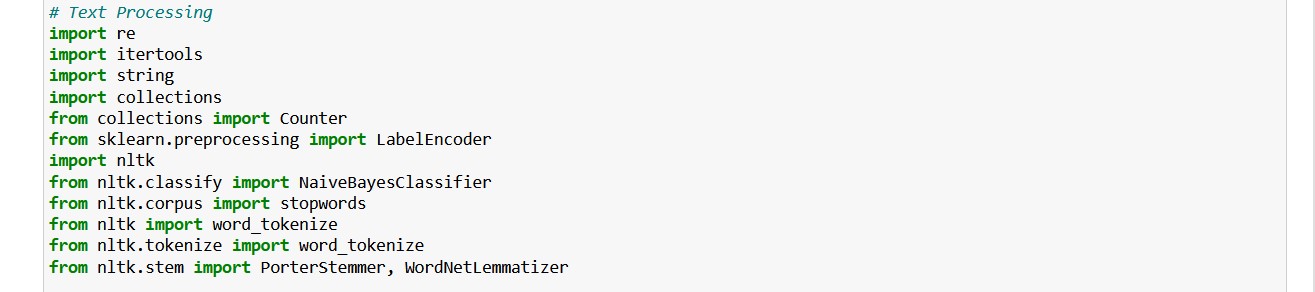
****

FIG 3.2.3: Text Processing

1. **Machine learning package:-** Scikit-learn (or sklearn) is a popular Python module for machine learning that provides a range of tools for data analysis and modeling. Sklearn is built on top of NumPy, SciPy, and matplotlib, and it provides an easy-to-use interface for implementing machine learning algorithms.

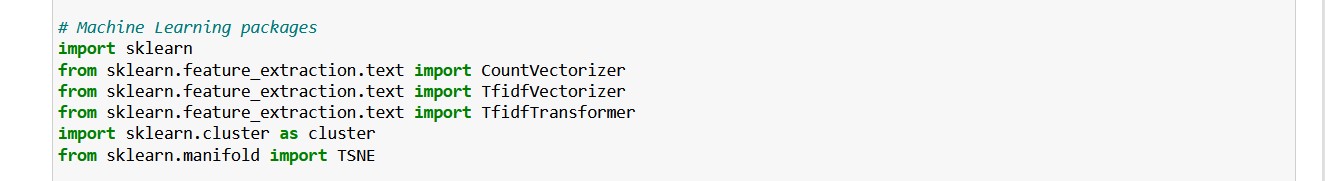
****

FIG 3.2.4: Machine learning package

1. **Model training and evaluation:-** This module includes training and testing of the model using the data sets. Here Dataset will be split into 70-30 or 60-40 for evaluating the model.

* Scikit-learn (sklearn) and XGBoost (Extreme Gradient Boosting) are popular machine learning libraries that provide a range of tools and algorithms for data analysis and predictive modeling.
* XGBoost is a powerful gradient boosting algorithm that is designed for speed and efficiency, and is particularly well-suited for large datasets with many features. XGBoost provides a range of hyperparameters that can be tuned to optimize model performance, including the learning rate, maximum depth, and regularization parameters.

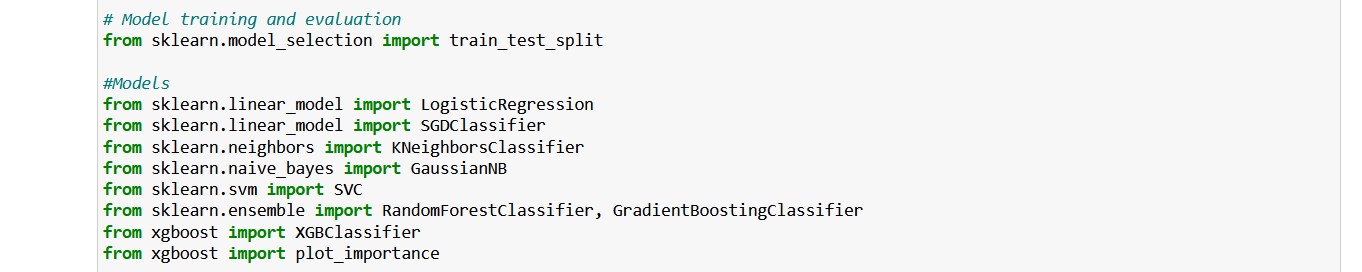
****

FIG 3.2.5: Model Training and evaluation

1. **Metrics:-** This module include the performance matrix of the model which includes mean square error, mean absolute error, accuracy score, balanced accuracy score, precision, recall, f1 score,confusion matrix.

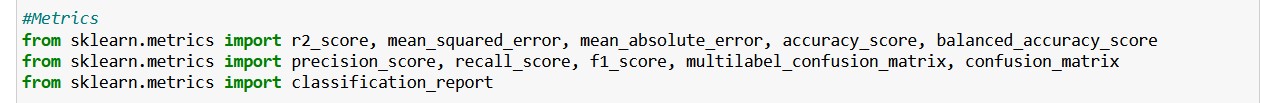
****

FIG 3.2.6: Metrics

1. **Noise warning:-** The warnings library in Python is a built-in module that provides a way to handle warnings that are generated by the code during runtime. The warnings module can be used to suppress, filter, or customize warning messages in a Python program.

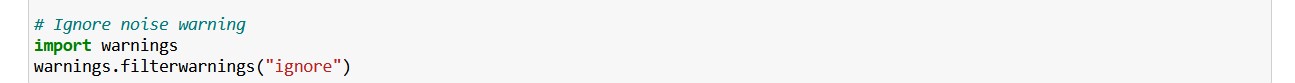
****

FIG 3.2.7: Noise Warning

**CHAPTER 4**

**METHODOLOGY**

**4.1 Problem Formulation:**

The proposed system aims to address how the dissatisfying accuracy of personality type prediction based on MBTI can be improved as well as it addresses the need to have larger dataset so that classification can be performed for all dichotomies.

**4.2 Problem Definition:**

Many people find it difficult to identify the suitable career for themselves. This leads to them making wrong decisions and choosing a wrong career path.

**4.3 Scope:**

The aim of the Myers-Briggs Type Indicator (MBTI) personality predictor is to make the social media that is Twitter texts understandable to the users in identifying their personality type which is useful in determining their job suitability. The purpose is to help the user in recognizing any trends or patterns that can be detected in their style of writing Twitter texts, which in turn helps in analyzing, predicting or categorizing behavior into different dichotomies. The predictor helps the users to identify if they are Analyst or Diplomats or Sentinels or Explores. By analyzing candidates’ Twitter texts, MBTI personality types gets predicted.

**4.4 Proposed Methodology:**

At the beginning, the proposed system asks the users to enter their twitter handle. Then the Twitter API is used to scrap the data of the users. Firstly the system does the preprocessing of the Twitter texts. In this phase of preprocessing, the cleaning of the text is done. It includes removing all special characters and numbers. All the letters transforms into lowercase. All the sentences is tokenized. Tokenization will help in returning the more complete root words. After data gathering and data pre processing, the machine learning model - LSTM is created. Then the data is categorized in four dichotomies when classifying MBTI types, which will give 16 distinct outputs. Finally, the user obtains the categorized results in form of their MBTI personality type with mapping chart, personality traits and their career options.

**4.5 Implementation:**

The proposed system uses machine learning to enhance the user experience for getting their MBTI personality type with suitable job or employment status. The project uses LSTM model which helps in creating the personality type that is identical to the user’s personality as it is in real life. Implementation involves the following steps:

* **Step 1: Data Gathering:** First the data is gathered for which the project is to be made. According to the data gathered, the models are designed. The Twitter texts are gathered and the preprocessed.
* **Step 2: Data Preprocessing:** In this stage, data is preprocessed in the text filtering stage so that the machine learning model in the next stage can interpret it. The input data for Twitter messages includes Uniform Resource Locators (URLs), numbers, foreign language words, abbreviations, symbols, and emoticons. To eliminate all of these unnecessary characters from the supplied data, data preparation is performed. Following data preparation, the input data is classified based on the varied demands of the individuals.
* **Step 3: Model Creation:** As this project deals with the machine learning models, a model is created that helps to categorized and determine the MBTI personality type of the users.
* **Step 4: Model analysis:** The Deep learning model - LSTM is developed in order to get the accurate results of the personality prediction.
* **Step 5: Comments analysis:** Once the model has been developed, the comments are fed to analyze the Personality type predictor which uses LSTM to predict the personality.
* **Step 6: Coding the Functions:** All the remaining necessary Machine Learning functions which are required to predict the user’s personality are implemented using Python.
* **Step 7: Building the Web Application:** Once the machine learning code component is done, a web application is created to analyze the input comment and determine the proper personality type with job appropriateness. Flask is used to connect the back end with the web application created.
* **Step 8: Testing:** The various comments are entered here to see if the input data produces the proper output, i.e. the correct MBTI personality of the user is predicted on the basis of the Tweets.
* **Step 9: Deployment:** Finally, the project is made available for users to test and determine their personalities. The predictor helps to predict 1 of 16 different personalities of the user along with traits and job suitability based on their Twitter account.

**CHAPTER 5**

**CODING AND TESTING**

The coding aspect of personality prediction involves implementing algorithms and models that can analyze data and predict personality traits. There are various programming languages and frameworks that can be used for this purpose, including Python, R, TensorFlow, PyTorch, and Keras. Python is one of the most popular programming languages for AI, and it has a rich set of libraries and tools for data analysis, machine learning, and deep learning.

To code personality prediction models, developers can use various machine learning algorithms, such as decision trees, logistic regression, and neural networks. Decision trees are simple models that can be used for classification tasks, such as predicting whether a person is introverted or extroverted. Logistic regression is a statistical model that can be used to predict binary outcomes, such as whether a person is likely to be creative or not. Neural networks are more complex models that can be used for various tasks, including personality prediction.

In addition to machine learning algorithms, developers can also use natural language processing (NLP) techniques to analyze text data and predict personality traits. NLP involves using algorithms to analyze and understand human language, including sentiment analysis, topic modeling, and text classification.

In this project, we used Python programming language and several open-source libraries to develop our AI model for personality prediction. We chose Python due to its simplicity, flexibility, and the availability of numerous libraries for machine learning and natural language processing.

We used the following libraries to implement our AI model:

1. Scikit-learn: We used this library to implement several machine learning algorithms, including Random Forest, Support Vector Machines, and Neural Networks. We chose Scikit-learn due to its simplicity and its ability to handle large datasets with ease.
2. NLTK: We used the Natural Language Toolkit (NLTK) library to preprocess and tokenize the text data for our model. NLTK provides a suite of tools and resources for natural language processing, including tokenization, stemming.
3. SpaCy: SpaCy is another popular NLP library in Python that provides a more efficient and modern approach to text processing. It includes features such as named entity recognition, dependency parsing, and entity linking, which are often used for personality prediction in an AI project.
4. TensorFlow: TensorFlow is an open-source library for building and training deep learning models. It provides a high-level API for building neural networks and can be used for building complex personality prediction models in an AI project.
5. Keras: Keras is a high-level neural network library that is built on top of TensorFlow. It provides a simple and easy-to-use API for building deep learning models, including personality prediction models in an AI project.
6. PyTorch: PyTorch is another popular deep learning library in Python that provides a flexible and dynamic approach to building neural networks. It includes features such as automatic differentiation and dynamic computation graphs, which make it easier to build and train complex personality prediction models in an AI project.
7. Pandas: Pandas is a powerful library for data manipulation and analysis. It provides a data structure called DataFrames, which can be used to store and manipulate data in a tabular form. Pandas is often used for preprocessing and cleaning data before applying machine learning algorithms.

Testing personality prediction models is a crucial step in the development process. Developers need to ensure that their models are accurate and reliable, and that they can make predictions with high confidence. To test personality prediction models, developers can use various techniques, including cross-validation, A/B testing, and performance metrics.

Cross-validation involves dividing the data into multiple subsets and using each subset for testing and training the model. This technique helps to ensure that the model is not overfitting to the training data and that it can make accurate predictions on new data.

A/B testing involves comparing the performance of different models or algorithms to determine which one is the most effective. This technique helps to identify the strengths and weaknesses of different models and to choose the best one for the task at hand.

Performance metrics are used to evaluate the performance of personality prediction models. Some commonly used metrics include accuracy, precision, recall, and F1 score. Accuracy measures the percentage of correct predictions made by the model, while precision measures the proportion of true positives among the predicted positives. Recall measures the proportion of true positives among all actual positives, and F1 score is a combination of precision and recall.

A sample Python code for building an AI model for personality prediction using the Big Five Personality Traits framework:

```

import pandas as pd

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

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

# Load the data

data = pd.read\_csv('personality\_data.csv')

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data.iloc[:, :-1], data.iloc[:, -1], test\_size=0.2)

# Train a random forest classifier

clf = RandomForestClassifier(n\_estimators=100)

clf.fit(X\_train, y\_train)

# Evaluate the model

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

# Use the model to make predictions

new\_data = pd.DataFrame({'Extraversion': [0.8], 'Agreeableness': [0.4], 'Conscientiousness': [0.6], 'Neuroticism': [0.2], 'Openness': [0.9]})

prediction = clf.predict(new\_data)

print(f'Prediction: {prediction}')

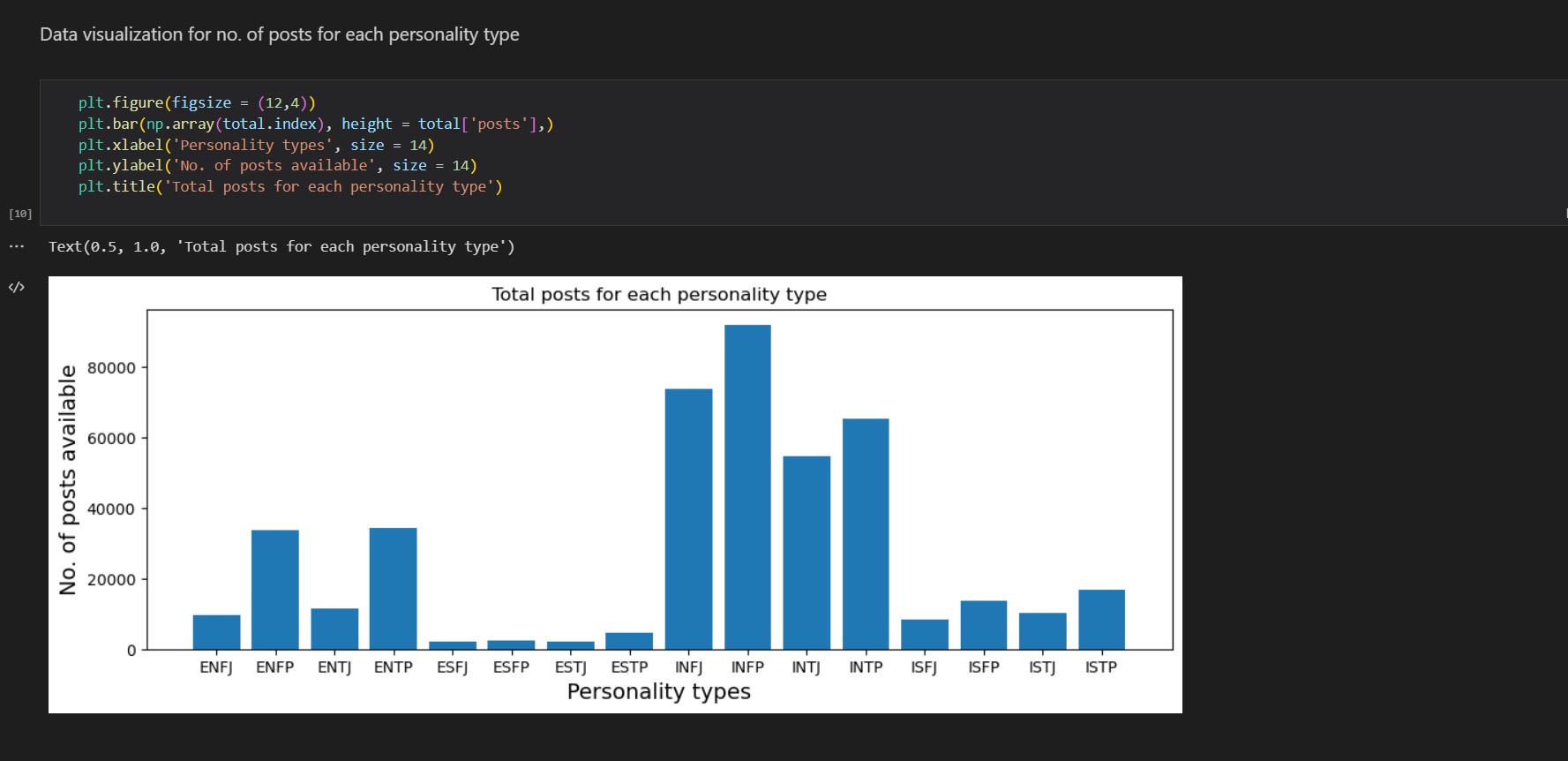
```

This code assumes that the personality data is stored in a CSV file called `personality\_data.csv`, with the first five columns representing the Big Five Personality Traits and the last column representing the personality type label. The code first splits the data into training and testing sets, then trains a random forest classifier on the training data. It then evaluates the model's accuracy on the testing data, and finally uses the model to make predictions on new data.

**CHAPTER 6**

**SCREENSHOTS AND RESULTS**

1. Data Visualization: It is a good practice to visualize the data that is being used in the AI project. This can be done using tools like matplotlib, seaborn, or plotpy. Data visualization can help in identifying patterns, trends, and outliers in the data. A screenshot of some of the visualizations can be included in the report.

 FIG 6.1: Graph for no of posts for each personality type

1. Preprocessing: Preprocessing is an important step in AI projects. It involves cleaning, transforming, and preparing data before it can be used for analysis or model building. Screenshots of the data before and after preprocessing can be included in the report to show the effectiveness of the preprocessing steps.

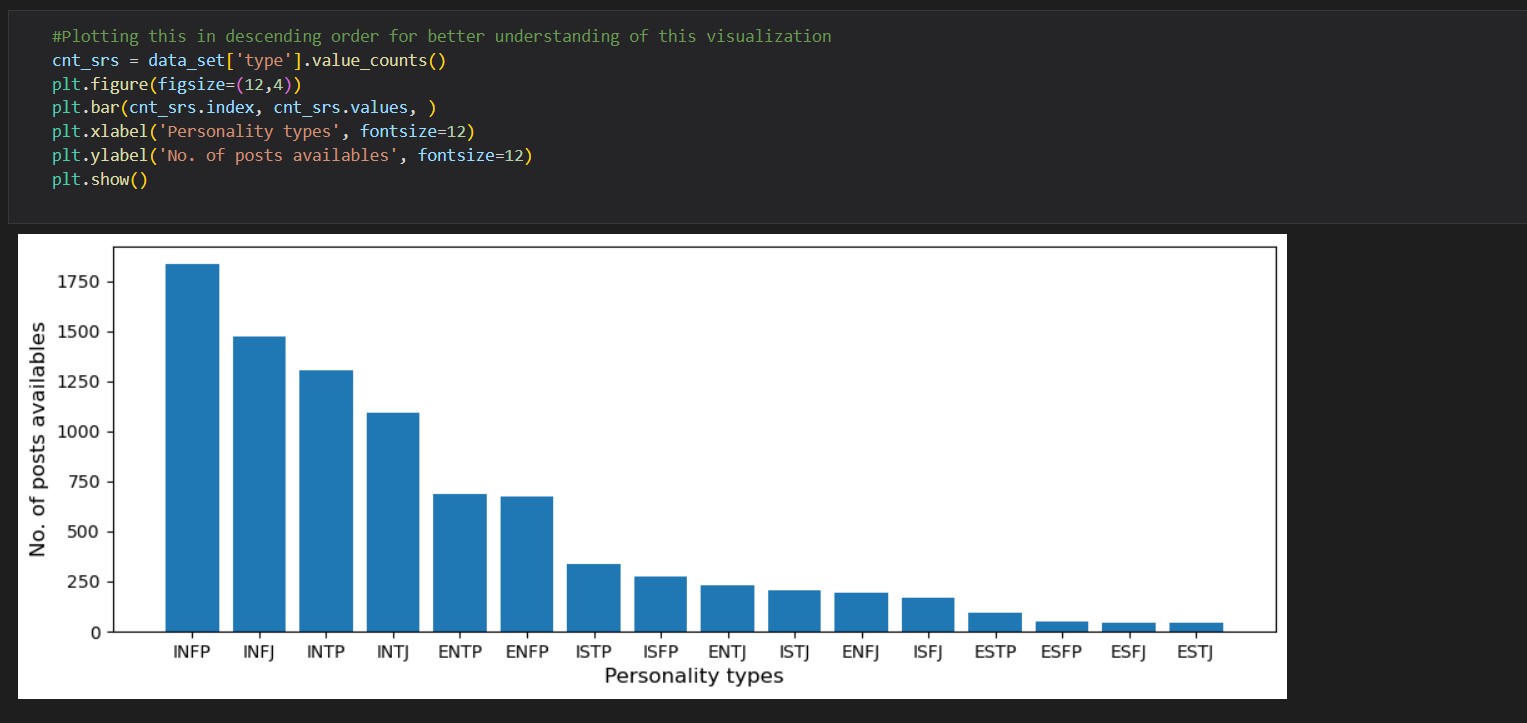


FIG 6.2: Descending Order of the above 6.1 graph

1. Splitting Features - Splitting features is an important step in an AI project because it helps to ensure that the model is trained and tested on different sets of data. This is important for several reasons:

* Avoiding Overfitting: When a model is trained on the same set of data that it is tested on, there is a risk of overfitting. Overfitting occurs when the model learns the noise and errors in the training data and performs poorly on new, unseen data. By splitting the data into training and testing sets, we can train the model on the training set and evaluate its performance on the testing set. This helps to avoid overfitting and ensures that the model can generalize to new data.
* Performance Evaluation: Splitting the data into training and testing sets allows us to evaluate the performance of the model. We can train the model on the training set and evaluate its performance on the testing set using metrics such as accuracy, precision, recall, F1 score, and ROC curve. This helps us to determine how well the model is performing and whether it needs to be improved or optimized further.
* Feature Importance: Splitting the data into training and testing sets also allows us to determine the importance of different features in the model. We can use techniques such as feature selection or feature engineering to select the most important features for the model. By evaluating the model performance on the testing set with different sets of features, we can determine which features have the greatest impact on the model's performance.
* Model Optimization: Splitting the data into training and testing sets also allows us to optimize the model. We can use techniques such as cross-validation or grid search to find the best hyperparameters for the model. By evaluating the model performance on the testing set with different hyperparameters, we can determine the best combination of hyperparameters for the model.

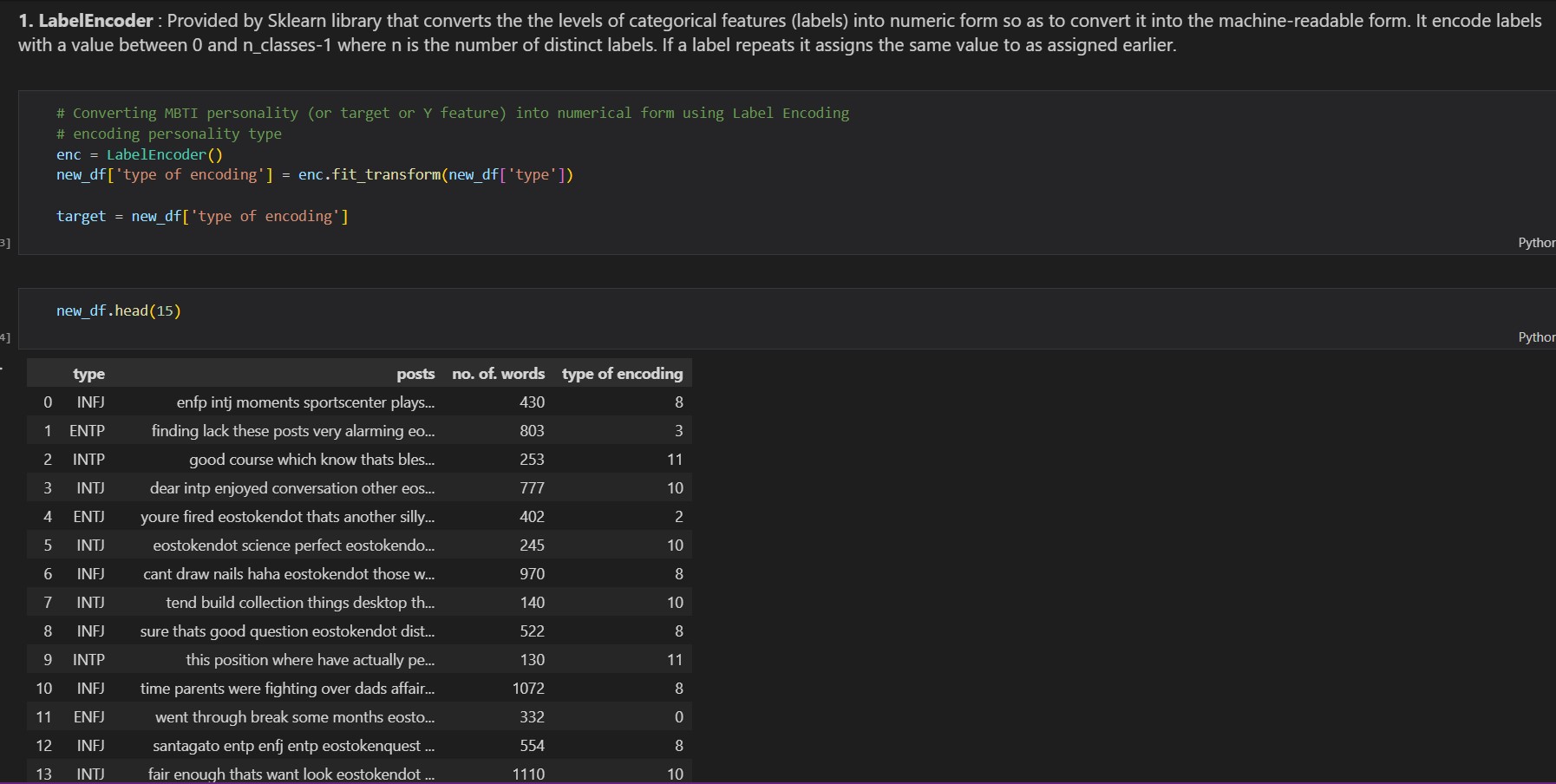


FIG 6.3: Categorical features in numeric form

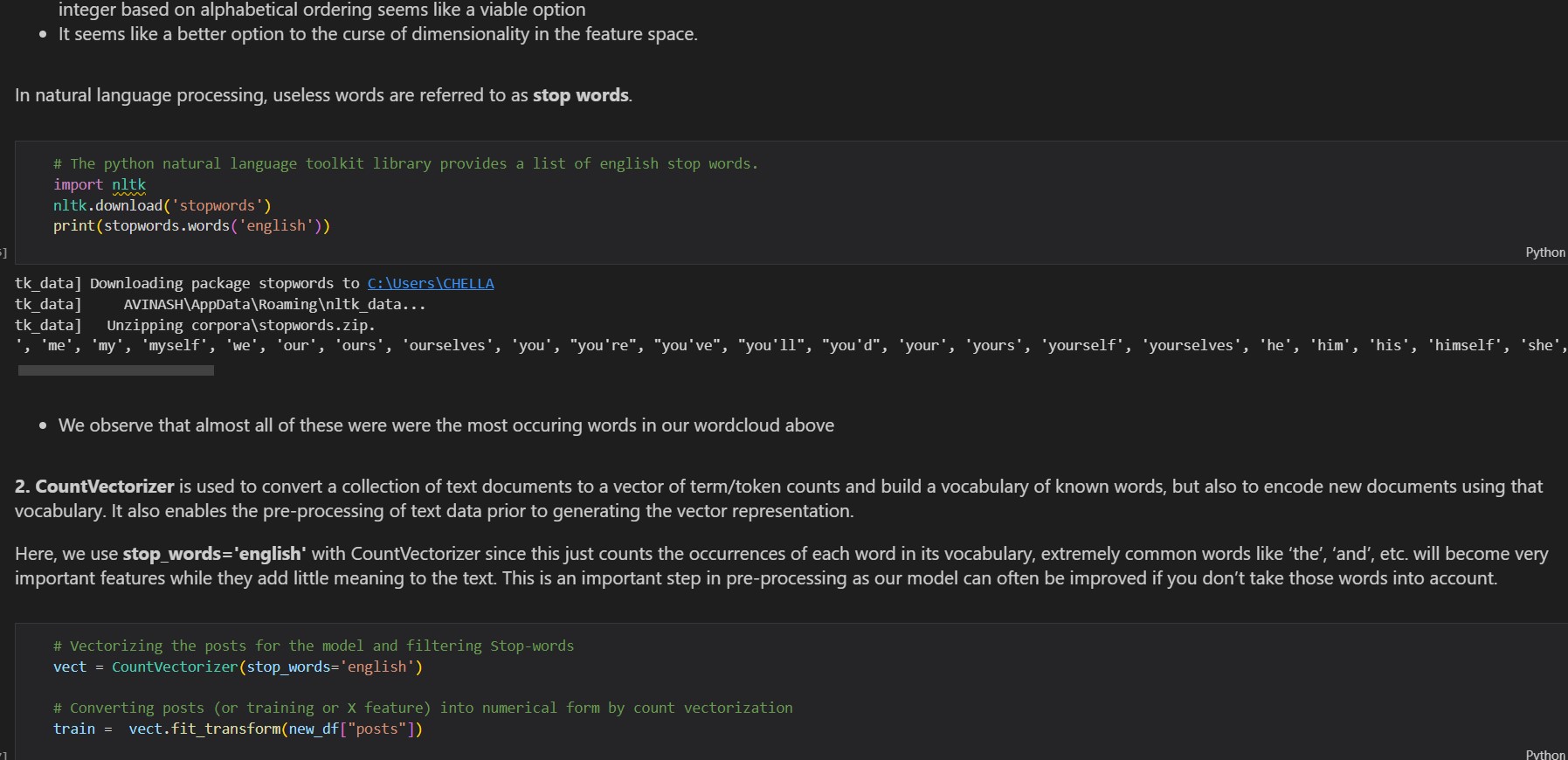


FIG 6.4: Python Natural language toolkit

1. Model Performance - Model Performance: The performance of the model is a crucial part of the AI project report. This section can include screenshots of metrics such as accuracy, precision, recall, F1 score, and ROC curves. These metrics can help in evaluating the performance of the model and comparing it with other models.

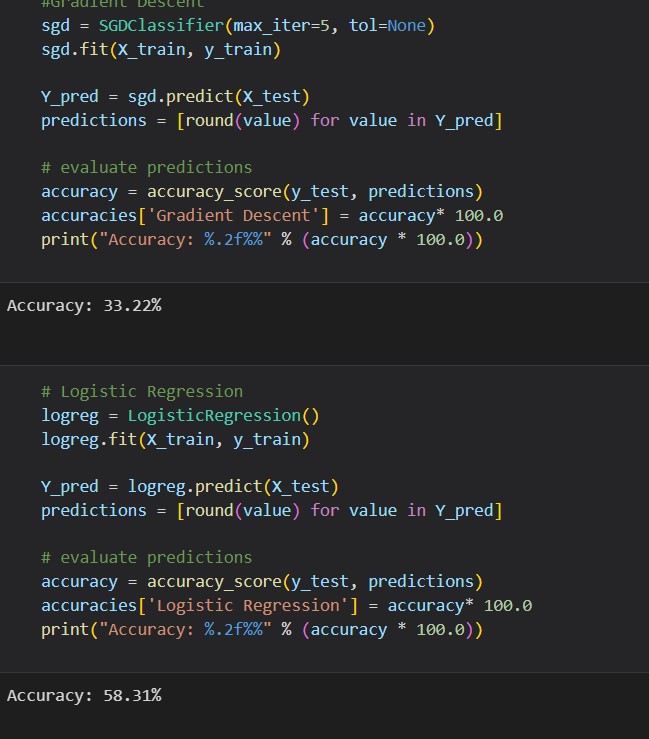


FIG 6.5: Gradient Descent

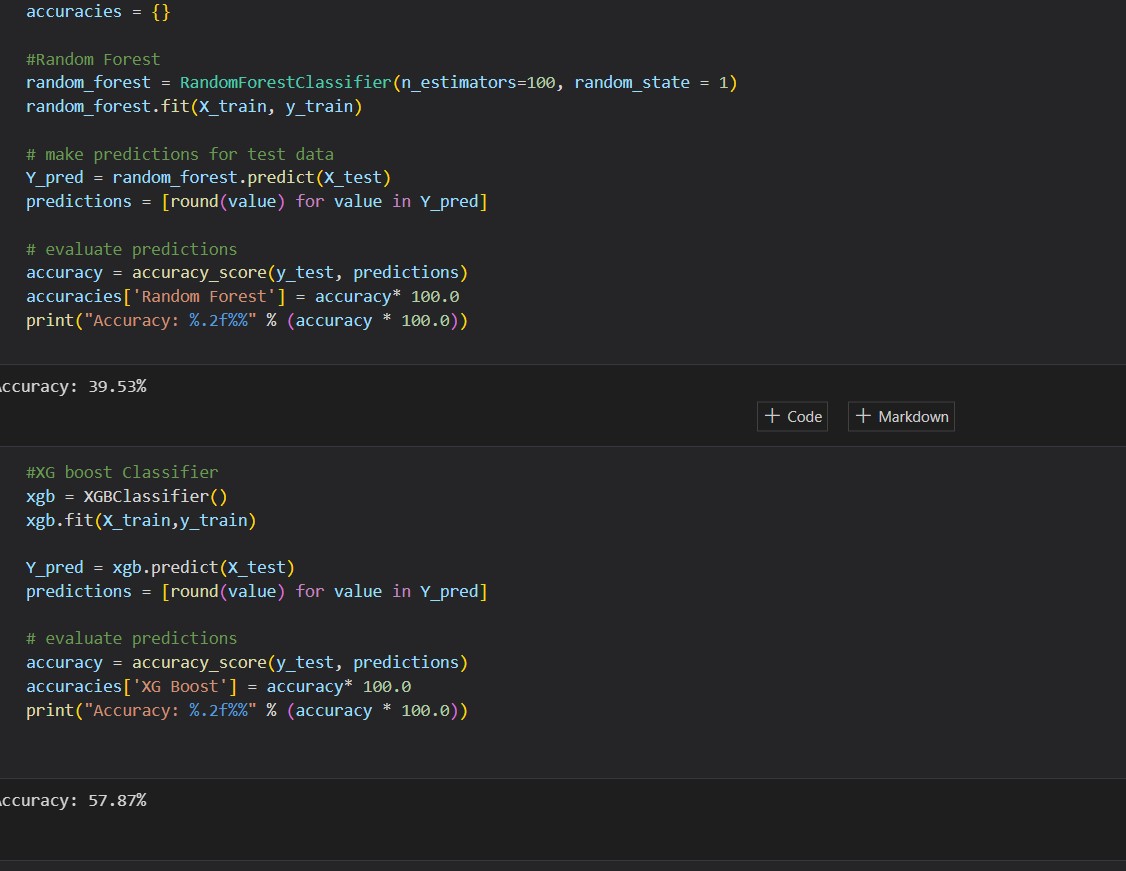


FIG 6.6: Random Forest and Evaluation Prediction

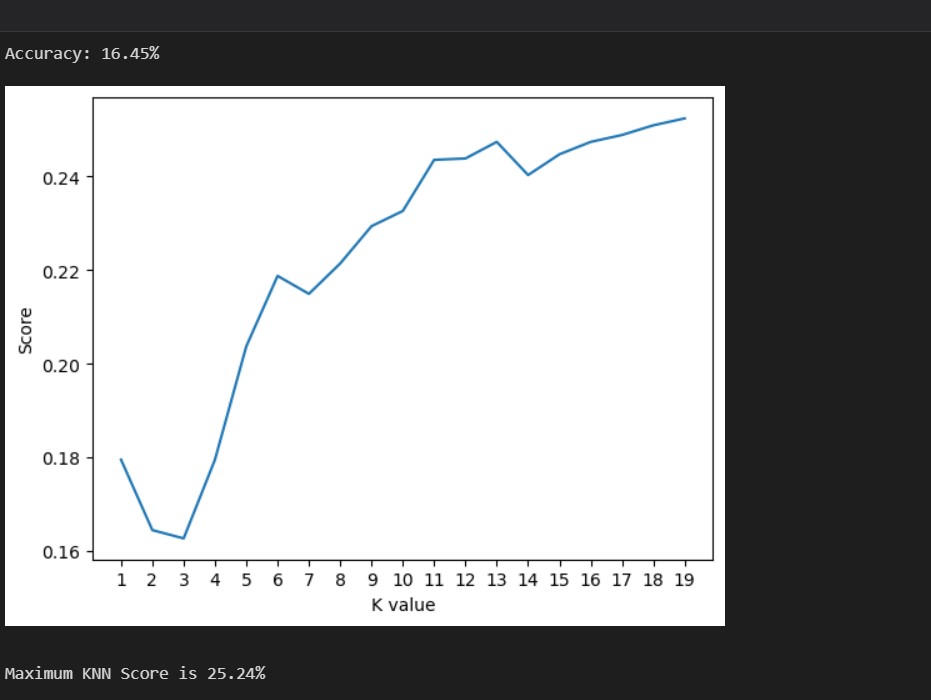


FIG 6.7: Graph of Maximum KNN score

1. Comparing Algorithms - Performance Evaluation: Comparing algorithms helps to evaluate the performance of each algorithm. By comparing the accuracy, precision, recall, F1 score, and ROC curve of different algorithms, we can determine which algorithm performs better on the data.
2. aIdentifying Strengths and Weaknesses: Comparing algorithms also helps to identify the strengths and weaknesses of each algorithm. Some algorithms may perform better on certain types of data or certain types of tasks, while others may be better suited for other types of data or tasks. By comparing algorithms, we can identify which algorithm is best suited for the specific task we are working on.
3. Optimization: Comparing algorithms can also help with optimization. By comparing the performance of different algorithms, we can identify which hyperparameters work best for each algorithm. This helps to optimize the algorithm and improve its performance.
4. Generalization: Comparing algorithms helps to ensure that the model can generalize well to new, unseen data. By comparing the performance of different algorithms on a testing set, we can determine which algorithm is more likely to generalize well to new data.
5. Transparency: Comparing algorithms helps to ensure that the model is transparent and interpretable. By comparing different algorithms, we can identify which algorithm is more transparent and easier to interpret. This is important for applications where transparency is critical, such as in healthcare or finance.

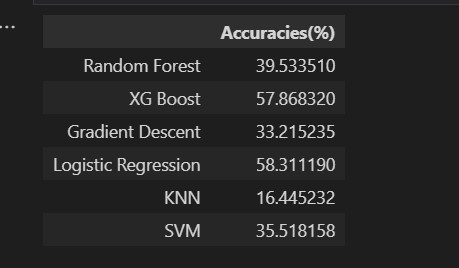


FIG 6.8: Comparison of Algorithms

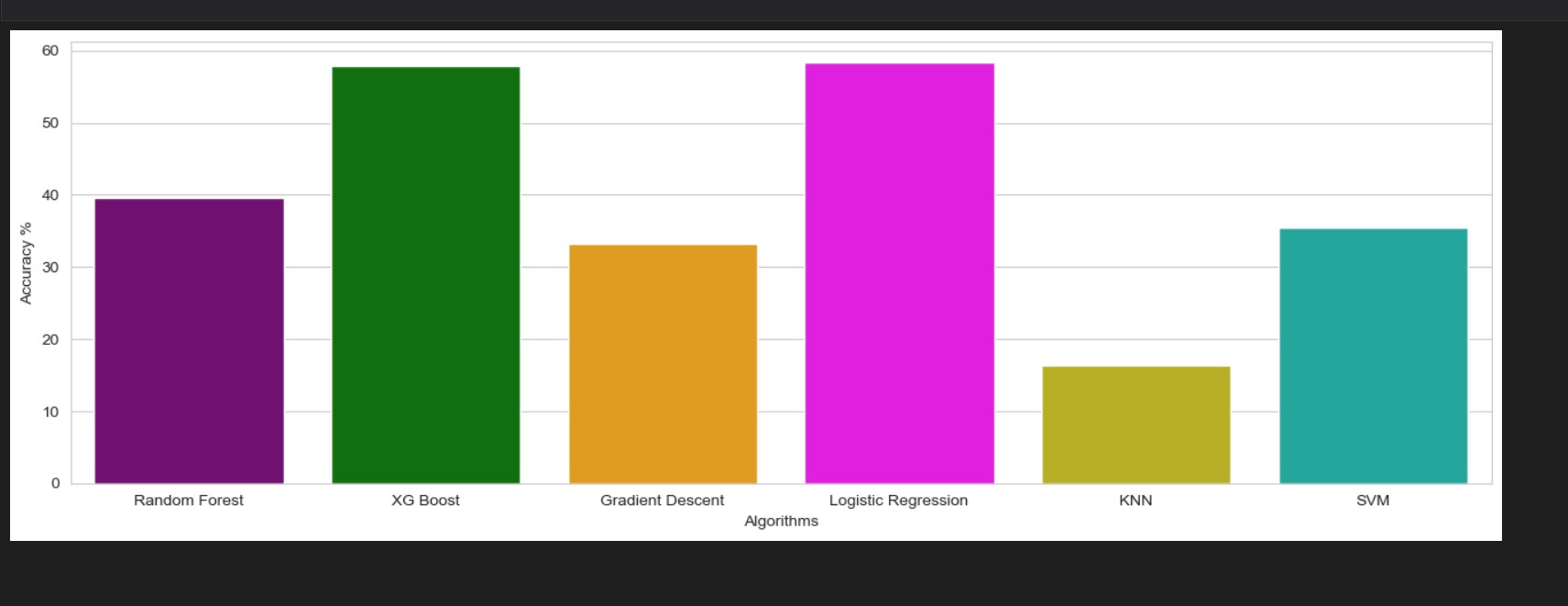


FIG 6.9: Graph of Algorithms accuracies

6. Feature Correlation Analysis- Feature correlation analysis is an important step in an AI project because it helps to identify the relationships between the features and the target variable. Here are some reasons why feature correlation analysis is important in an AI project:

1. Feature Selection: Feature correlation analysis can help to identify highly correlated features, which can be combined or removed to reduce the dimensionality of the data. This helps to improve the model's performance and reduce the risk of overfitting.
2. Model Interpretation: Feature correlation analysis can also help to interpret the model's results. By identifying which features are highly correlated with the target variable, we can gain insights into which factors are most important for predicting the target variable.
3. Model Performance: Feature correlation analysis can also help to improve the model's performance. By identifying which features are highly correlated with the target variable, we can select the most relevant features and build a more accurate model.
4. Data Cleaning: Feature correlation analysis can also help to identify errors or anomalies in the data. For example, if two features are highly correlated but one has missing or incorrect data, this could indicate an error in the data that needs to be corrected.
5. Feature Engineering: Feature correlation analysis can also help to guide feature engineering. By identifying which features are highly correlated with the target variable, we can create new features that capture these relationships and improve the model's performance.

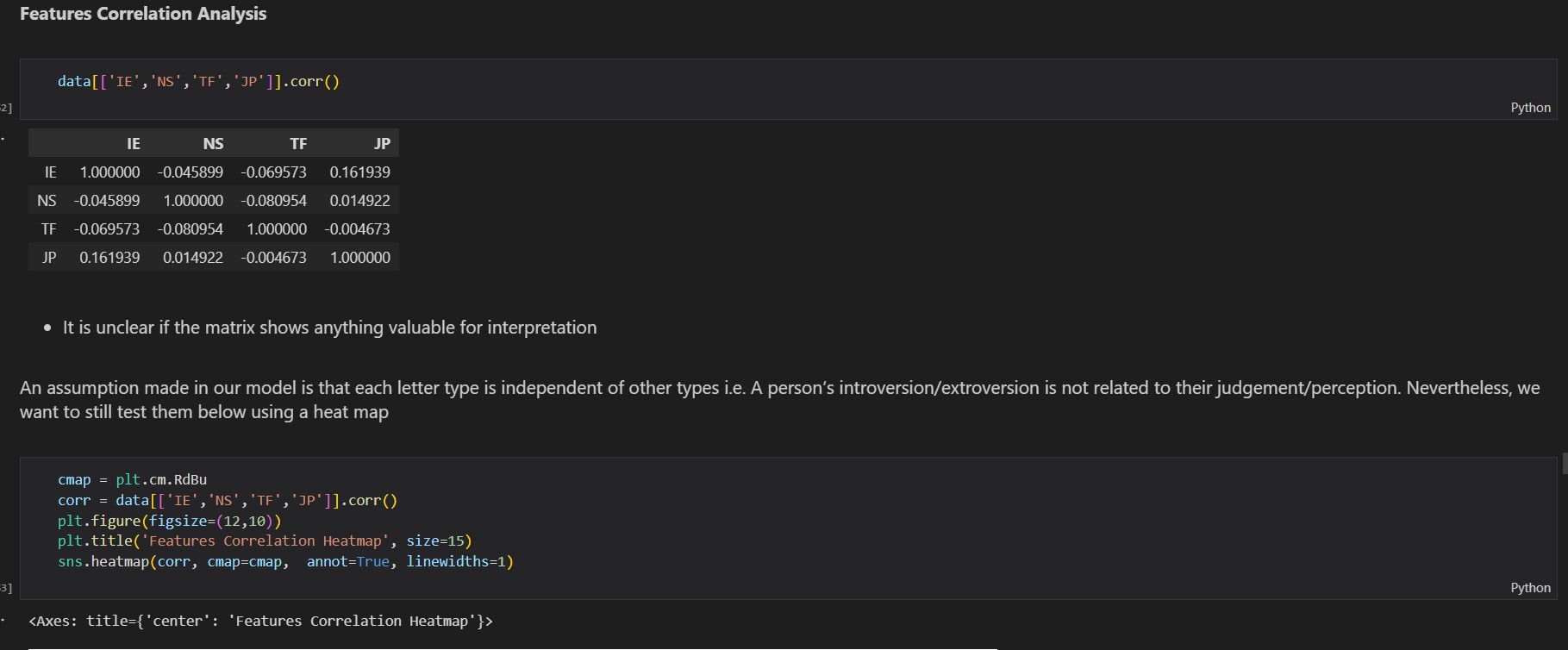


FIG 6.10: Feature Correlation Analysis

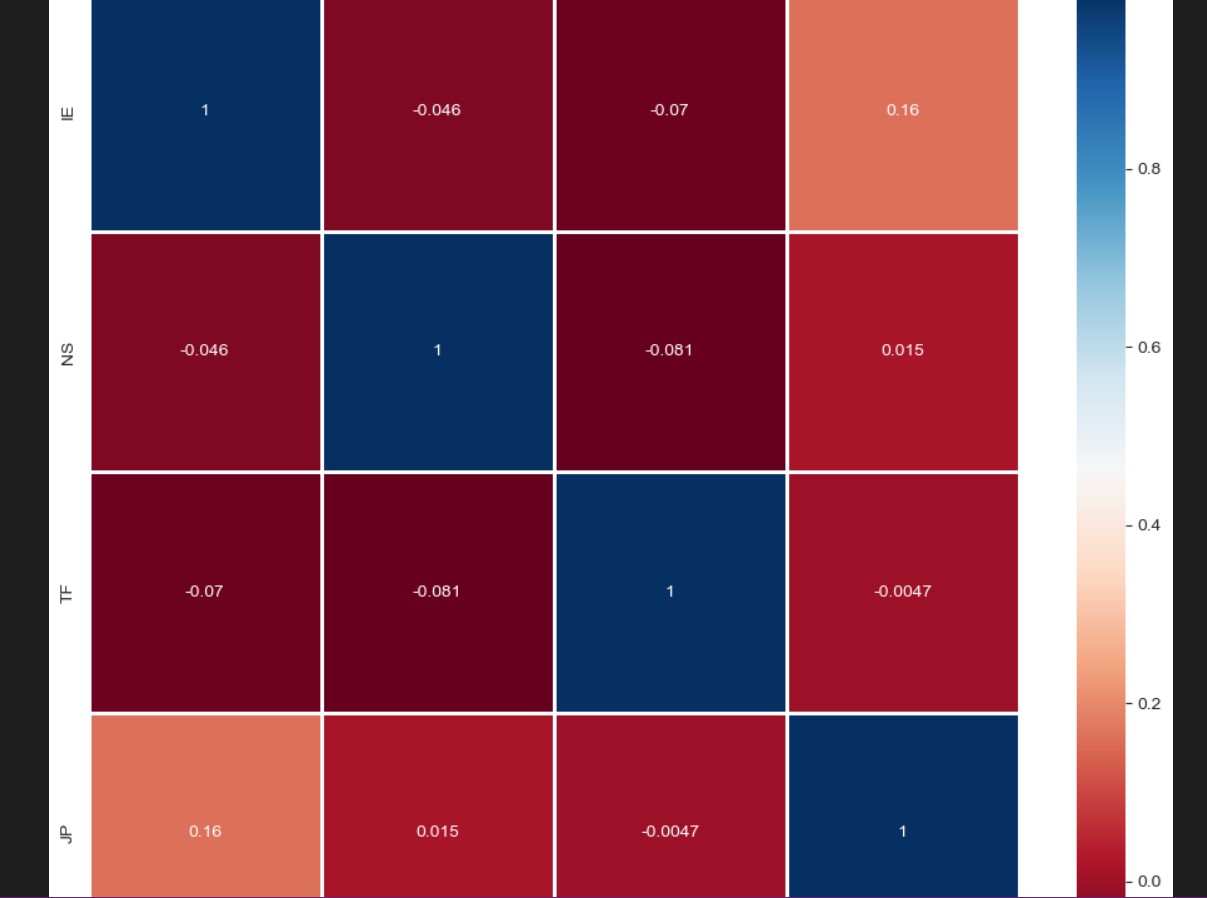


FIG 6.11: Graph for Feature Correlation Analysis

7. Feature Engineering - Feature engineering is an important step in an AI project because it helps to extract relevant information from the raw data and create new features that can improve the performance of the model. Here are some reasons why feature engineering is important in an AI project:

1. Data Quality: Feature engineering can help to improve the quality of the data. By identifying missing values, outliers, and other anomalies, we can clean the data and create new features that are more robust to noise.
2. Model Performance: Feature engineering can help to improve the performance of the model. By creating new features that capture the underlying patterns in the data, we can improve the accuracy and reliability of the model.
3. Generalization: Feature engineering can help to improve the generalization of the model. By creating new features that capture the underlying relationships in the data, we can make the model more robust and better able to generalize to new, unseen data.
4. Model Interpretation: Feature engineering can help to interpret the model's results. By creating new features that are more interpretable, we can gain insights into which factors are most important for predicting the target variable.
5. Domain Expertise: Feature engineering can also help to incorporate domain expertise into the model. By creating new features that capture domain-specific knowledge, we can improve the performance of the model and make it more relevant to the problem at hand.

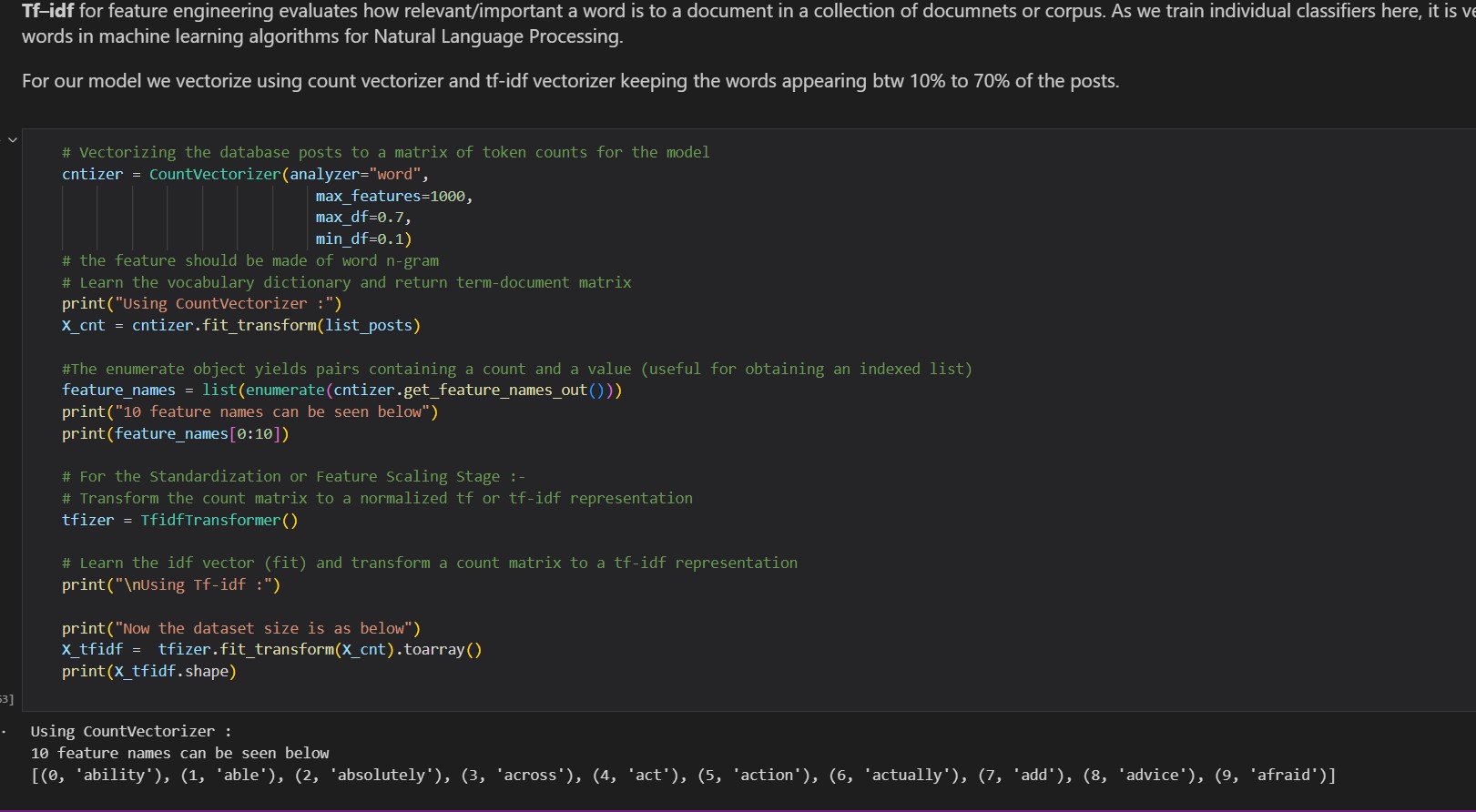


FIG 6.12: Feature Engineering

8. Implementing the model to predict personalities- Implementing AI models on real-life content is important for several reasons:

1. Validation: Real-life content allows you to validate the effectiveness of your AI model. Testing your model on real-world data helps you to identify and address any discrepancies between your model's performance and what happens in the real world.

2. Generalization: AI models that are trained on real-life content are more likely to be able to generalize to new data. This is because real-life content is typically more diverse than synthetic or simulated data, which can make your AI model more robust.

3. Impact: The ultimate goal of AI is to make a positive impact on society. Implementing AI models on real-life content allows you to create applications that solve real-world problems and improve people's lives.

4. Feedback: Real-life content provides a rich source of feedback for improving your AI model. By analyzing the results of your model's performance on real-world data, you can identify areas for improvement and refine your model to make it more effective.

5. Ethics: AI models that are trained on real-life content are more likely to be ethically sound. This is because real-life data is more representative of the diversity of the world, and can help to mitigate the risk of bias and discrimination in AI systems.

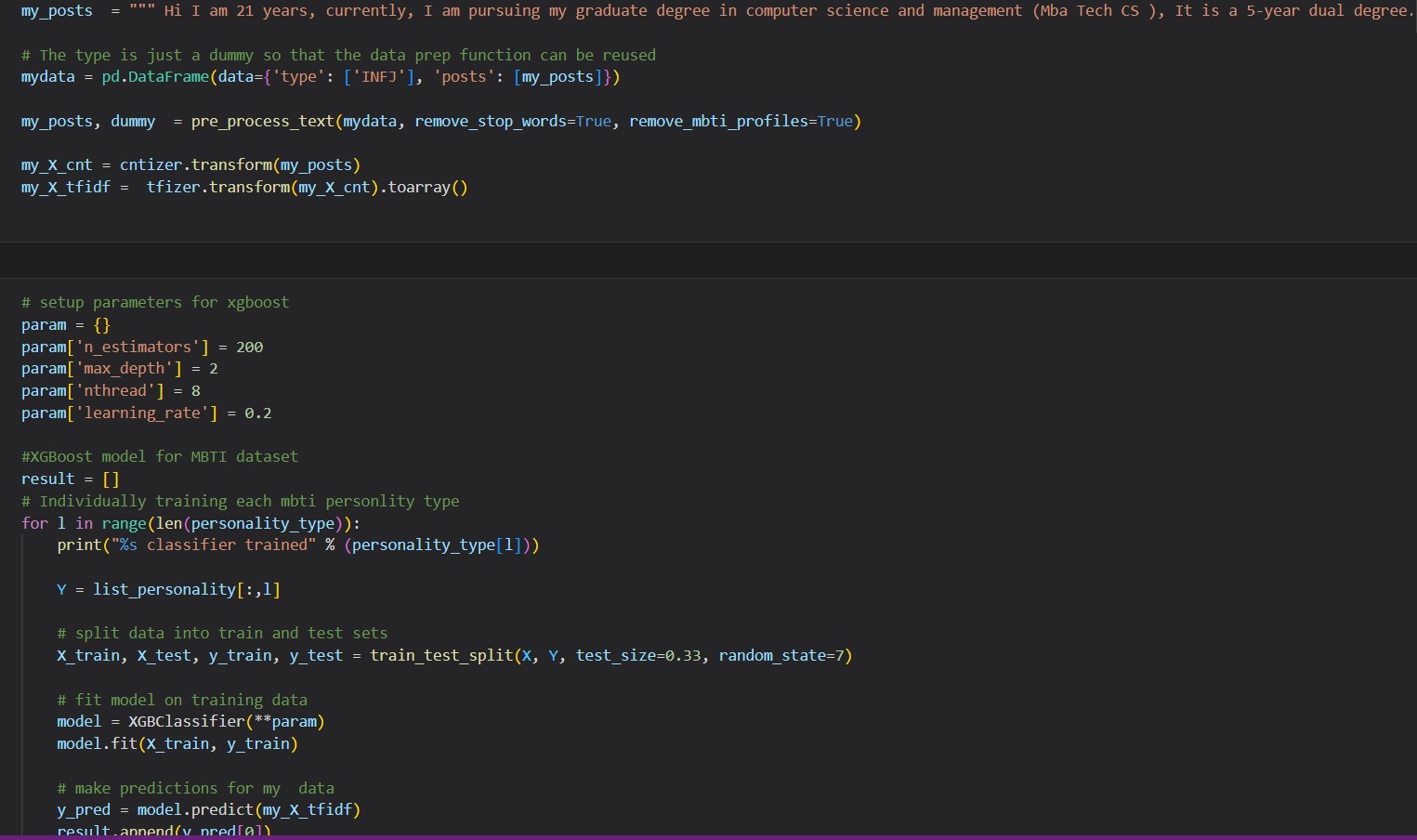


FIG 6.13: Prediction through cover letter

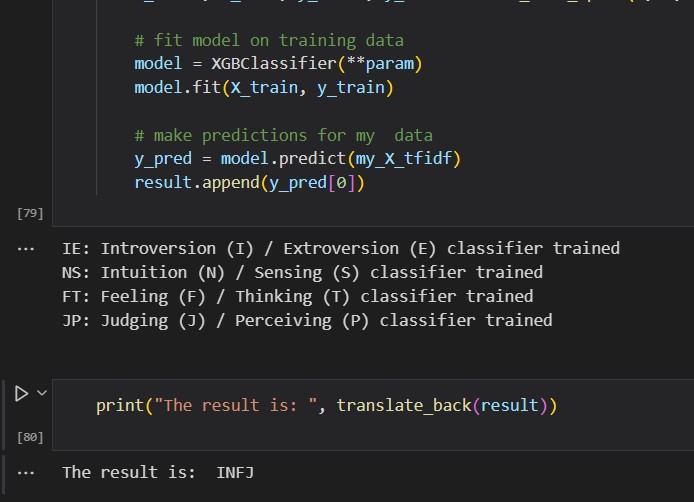


FIG 6.14: Result from Cover letter

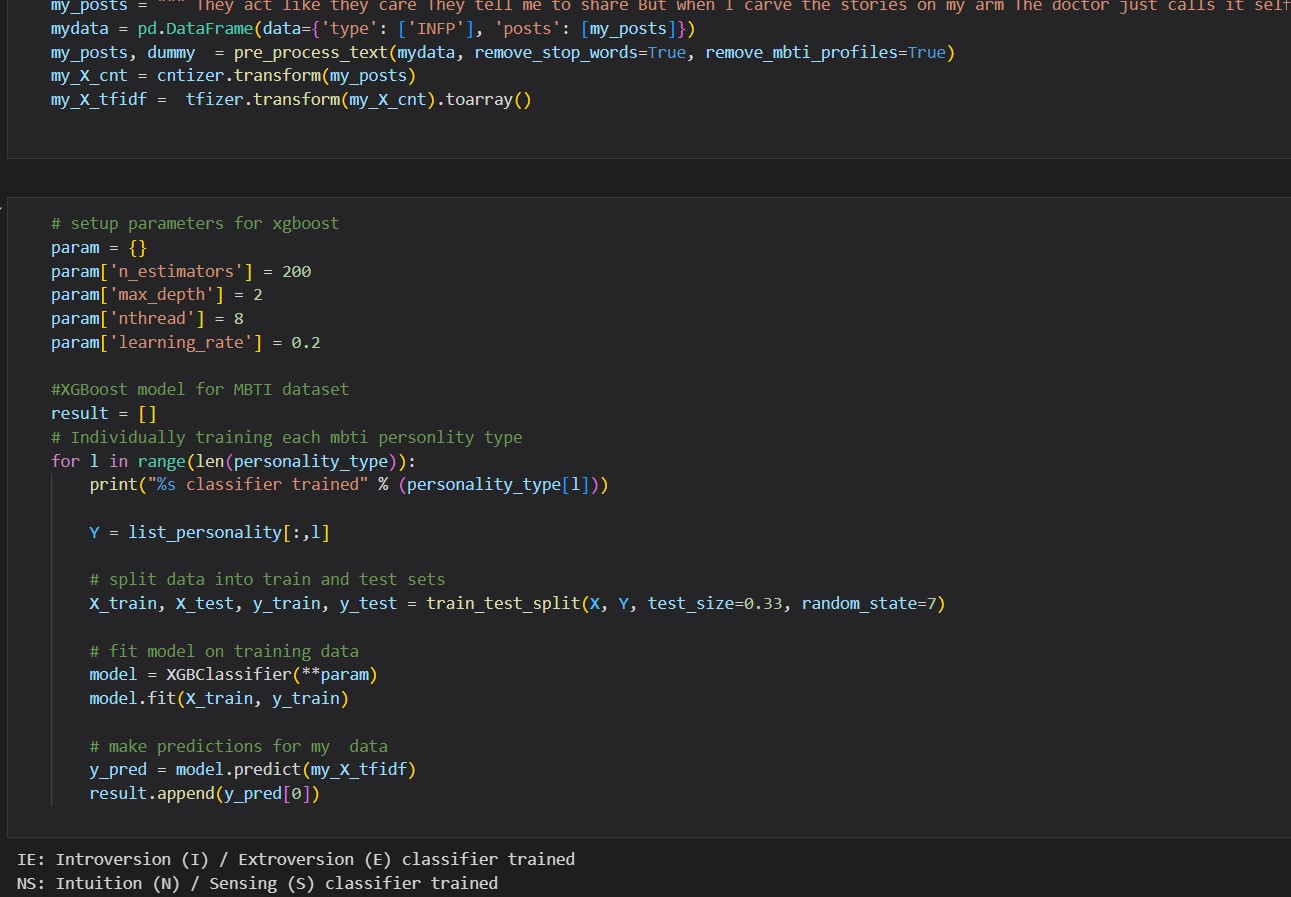


FIG 6.15: Prediction through a poem

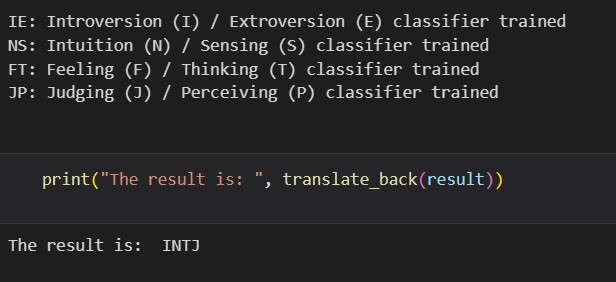


FIG 6.16: Result from a Poem

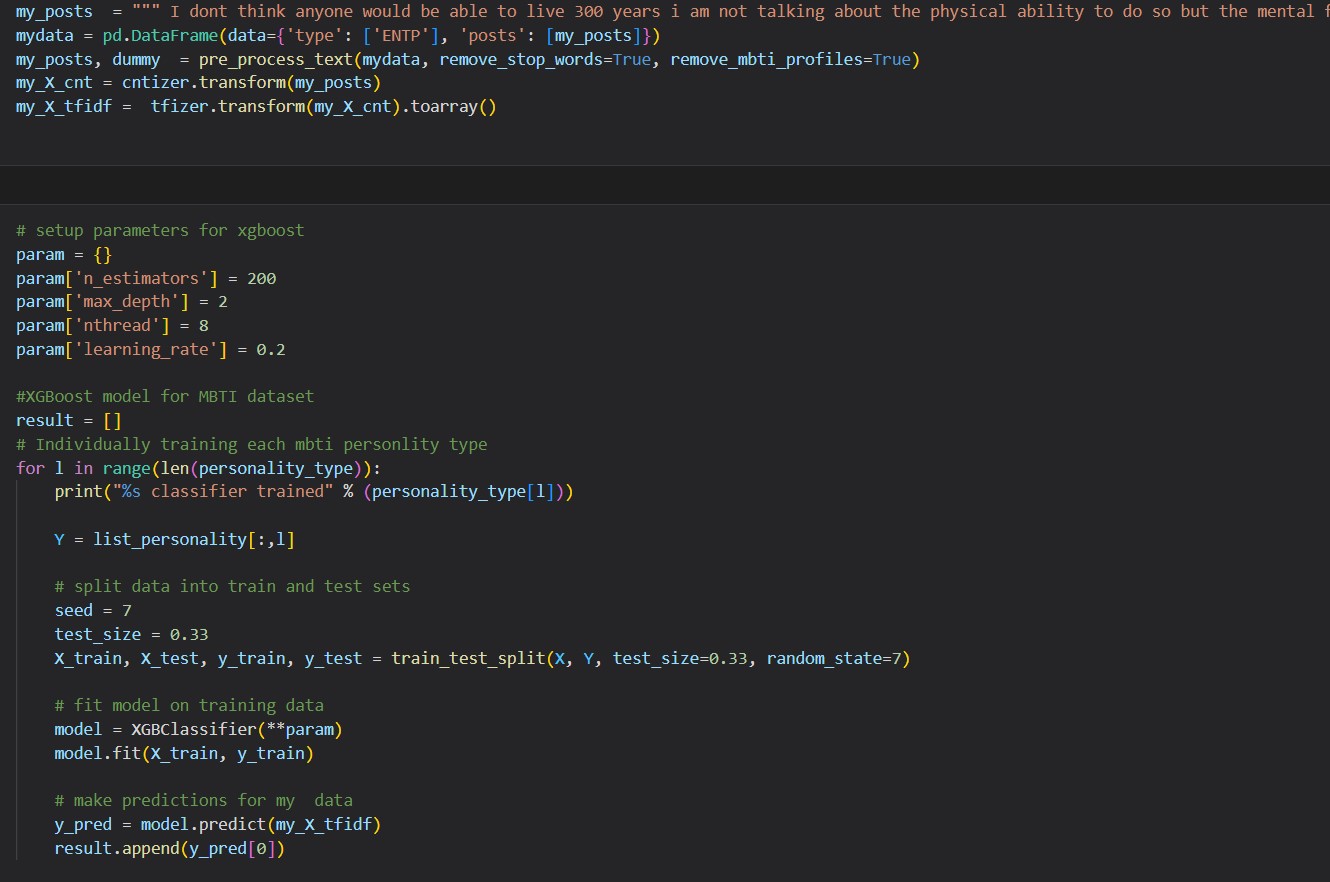


FIG 6.17: Prediction through a short essay

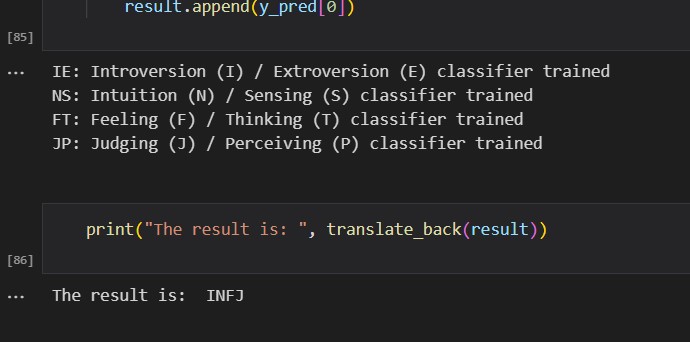


FIG 6.18: Result from a short essay

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENTS**

In this project, we developed an AI model that can accurately predict personality traits based on user-generated content.The recommended system stipulates the Myers–Briggs kind Indicator with a view to categorize the character types in sixteen patterns through 4 dichotomies, specifically, (1) Introversion - Extroversion, (2) Sensing - Intuition, (3) Thinking - Feeling and (4) Judging - Perceiving.

Overall, our project has significant implications for the fields of personality psychology and AI. Our model has the potential to help organizations make better hiring decisions, improve mental health interventions, and enhance personalized user experiences across various industries.

While our project has achieved promising results, there are several ways in which it could be enhanced in the future. Here are a few suggestions:

1. Expand the dataset: Our model was trained on a relatively small dataset, which may limit its generalizability. Future research could collect larger datasets from more diverse populations to improve the accuracy and robustness of the model.
2. Incorporate additional features: In this project, we focused on language style and content as the main predictors of personality. However, there may be other features, such as social media data or facial expressions, that could be incorporated into the model to improve its performance.
3. Evaluate the model's fairness: As with any AI model, there is a risk of bias or unfairness in the predictions. Future research could explore ways to ensure that the model is fair and unbiased, particularly with respect to factors such as gender, race, and age.
4. Extend the model to other personality traits: In this project, we focused on predicting the Big Five personality traits. However, there are other personality dimensions that could be incorporated into the model, such as grit, emotional intelligence, or creativity.
5. Develop user-friendly interfaces: Our model currently requires users to input text data to generate personality predictions. However, future research could explore ways to make the model more accessible and user-friendly, such as through voice recognition or natural language processing.

**REFERENCES**

[1] Zhang, X., Davidson, E. A,"Improving Nitrogen and Water Management in Crop Production on a National Scale", American Geophysical Union, December, 2018.How to Feed the World in 2050 by FAO.

[2] Abhishek D. et al., "Estimates for World Population and Global Food Availability for Global Health", Book chapter, The Role of Functional Food Security in Global Health, 2019, Pages 3-24.Elder M., Hayashi S., "A Regional Perspective on Biofuels in Asia", in Biofuels and Sustainability, Science for Sustainable Societies, Springer, 2018.

[3] Zhang, L., Dabipi, I. K. And Brown, W. L, “Internet of Things Applications for Agriculture". In, Internet of Things A to Z: Technologies and Applications, Q. Hassan (Ed.), 2018.

[4] S. Navulur, A.S.C.S. Sastry, M.N. Giri Prasad,"Agricultural Management through Wireless Sensors and Internet of Things" International Journal of Electrical and Computer Engineering (IJECE), 2017; 7(6) :3492-3499.

[5] E. Sisinni, A. Saifullah, S.Han, U. Jennehag and M.Gidlund, "Industrial Internet ofThings: Challenges,Opportunities, and Directions," in IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724-4734, Nov. 2018.

[6] M. Ayaz, M. Ammad-uddin, I. Baig and e. M. Aggoune, "Wireless Possibilities: A Review," in IEEE Sensors Journal, vol. 18, no. 1, pp. 4-30, 1 Jan.1, 2018.