Personality Detection from

Resume using Machine Learning

Group Members:

|  |
| --- |
| Sharmistha Das  Asansol Engineering College  10800123164 |
| Rishika Mishra  Asansol Engineering College  10800123164 |
| Ragini Gupta  Asansol Engineering College  10800123145 |
| Shreya Banerjee  Asansol Engineering College  10800123192 |

# Contents

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Topic** | **Page No.** |
| 1. | Acknowledgement | 1 |
| 2. | Project Objective | 2-3 |
| 3. | Project Scope | 4 |
| 4. | Data Description | 5-6 |
| 5. | Data Pre-Processing | 7-19 |
| 6. | Model Building | 20-28 |
| 7. | Test Dataset | 29-32 |
| 8. | Code | 33-69 |
| 9. | Future Scope of Improvements | 70-71 |
| 10. | Conclusion | 72-75 |
| 11. | Certificates | **76-79** |

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty, **Prof. Arnab Chakraborty** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Sharmishta Das Rishika Mishra

Ragini Gupta

Shreya Banerjee

# Project Objective

* **Problem Statement**

In the modern recruitment process, organizations receive thousands of resumes for a limited number of positions. Recruiters often spend significant time manually screening resumes, which is inefficient, subjective, and prone to bias.

In addition to technical qualifications, companies are also keen to understand the personality traits of candidates, as these play a crucial role in job performance, teamwork, and cultural fit. To capture personality effectively, this project makes use of the Myers–Briggs Type Indicator (MBTI) framework.

The MBTI is one of the most widely recognized personality models, dividing individuals into 16 personality types derived from four key dimensions:

* **Extraversion (E) – Introversion (I)**: outward vs. inward energy focus.
* **Sensing (S) – Intuition (N)**: practical details vs. abstract ideas.
* **Thinking (T) – Feeling (F)**: logical analysis vs. empathy-based decisions.
* **Judging (J) – Perceiving (P)**: structured planning vs. flexible adaptability.

Each MBTI type (e.g., INTJ, ENTP, INFJ, ESTJ) provides valuable insight into how a candidate communicates, makes decisions, and approaches problem-solving. Incorporating MBTI into the recruitment pipeline allows for better role alignment, efficient shortlisting, and stronger team-building compared to skill-only screening.

* **Objective**

The main objective of this project is to develop a **machine learning system** that can

automatically analyze resumes and predict the **personality type of candidates**. The

project focuses on going beyond traditional recruitment methods that only evaluate

skills and qualifications, by also incorporating **personality insights** for better job-role

alignment.

The specific objectives are:

* To **extract structured information** from resumes, including skills, education, work experience, certifications, and job titles.
* To **predict candidate personalities** based on resume features using machine learning models such as Logistic Regression, Random Forest, K-Means, and Hierarchical Clustering.
* To **compare models** using classification metrics (Accuracy, Precision, Recall, F1-score, AUC) and ROC curve.
* To **recommend suitable job roles** aligned with the predicted personality types, thereby improving recruitment efficiency and role-personality matching.
* To provide recruiters with a **data-driven decision-making tool** that reduces subjectivity and saves time in the hiring process.
* **Methodology**
* **Data Collection** – Use a resume dataset with MBTI personality labels.
* **Data Preprocessing** – Clean text, remove noise, perform tokenization, and apply TF-IDF for feature extraction.
* **Model Training** – Apply multiple models (Logistic Regression, Random Forest, K-Means, Hierarchical Clustering).
* **Evaluation** – Compare models based on Accuracy, Precision, Recall, F1-score, AUC, and clustering metrics (ARI/NMI).
* **Final Selection** – Choose the most practical and interpretable model (Logistic Regression).

# Project Scope

The broad scope of the project includes:

* **Resume Parsing**: Automatically extract career objectives, skills, degrees, and experiences.
* **Feature Engineering**: Convert categorical and numerical attributes into structured features for ML models.
* **Personality Classification**: Predict MBTI-based personality categories (e.g., Introvert/Extrovert, Thinker/Feeler).
* **Job Recommendation**: Map personality predictions to relevant job categories (e.g., Analysts for INTJs, Managers for ENTJs, Creative Designers for ENFPs).
* **Visualization**: Provide visual insights such as confusion matrices, ROC curves, and clustering dendrograms.
* **Limitations**:
* Works best with English resumes.
* Requires labeled datasets for supervised learning.
* Predictions depend heavily on text quality and completeness of resumes

# Data Description

The dataset used for this project consists of structured resume information collected from Kaggle. It contains **~9,500 records** and multiple columns that describe various aspects of a candidate’s resume, such as skills, education, experience, and job positions. The target variable is the **personality label (MBTI type)**, which is mapped to resumes based on their features.

* **Dataset Characteristics**
* **Total Records**: 9,544
* **Features**: 30+ attributes including career objective, skills, education, job roles, certifications, etc.
* **Target Variable**: MBTI personality type (INTJ, ENTP, INFJ, ESTJ, etc.).

| **Column Name** | **Non-Null Count** | **Null Count** | **Data Type** | **Description** |
| --- | --- | --- | --- | --- |
| career\_objective | 4740 | 4804 | object | Candidate’s professional goal/objective |
| skills | 9488 | 56 | object | Candidate’s listed skills |
| degree\_names | 9460 | 84 | object | Academic degree(s) held |
| educational\_institution\_name | 9460 | 84 | object | Name of university/college |
| job\_position\_name | 9544 | 0 | object | Current or desired job position |
| experiencere\_requirement | 8180 | 1364 | object | Years of required/mentioned work experience |
| age\_requirement | 5457 | 4087 | object | Age-related requirements if specified |
| certifications | 2008 | 7536 | object | Certifications acquired by candidate |
| responsibilities | 9544 | 0 | object | Roles and responsibilities in past jobs |
| matched\_score | 9544 | 0 | float64 | Matching score between resume and job profile |

* **Data Types**: Mostly categorical/text, with one numeric column (matched\_score).
* **Missing Values**: Present in several columns (e.g., career\_objective, experience\_requirement, address, etc.

Table 1: Feature Information

| **Feature** | **Min Length** | **25%** | **Median** | **75%** | **Max Length** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| career\_objective | 26 | 144 | 210 | 268 | 1425 | Candidate’s objective/summary statement length |
| skills | 2 | 161 | 243 | 504 | 3104 | Length of skills list (characters) |
| degree\_names | 6 | 10 | 22 | 39 | 472 | Degree names length (characters) |
| educational\_institution\_name | 8 | 27 | 42 | 61 | 212 | Institution name length (characters) |
| job\_position\_name | 5 | 12 | 22 | 35 | 150 | Job title length (characters) |

Table 2: Text Feature Length Statistics

* **Insights from Dataset**

1. **Career Objective** is available in ~50% of resumes; it varies from very short (26 characters) to detailed statements (1425 characters).
2. **Skills** are well populated, with a median length of ~243 characters, reflecting a mix of technical and soft skills.
3. **Degree Names and Institutions** are consistent with typical resume formats (median length ~22–42 characters).
4. **Job Position Names** are always present, with reasonable lengths (median ~22 characters).
5. **Matched Score** is a numeric column (0–0.97), representing how well a resume aligns with a job posting, which can be used as a strong feature for prediction.
6. Several columns like **address, extracurriculars, certifications** have high missing values and may need to be dropped or imputed.

# Data Pre-processing

As our dataset contained raw resumes in textual form along with their associated MBTI personality labels, several preprocessing steps were required to ensure that the data was consistent, structured, and suitable for training classification models. The detailed methodology is given below:

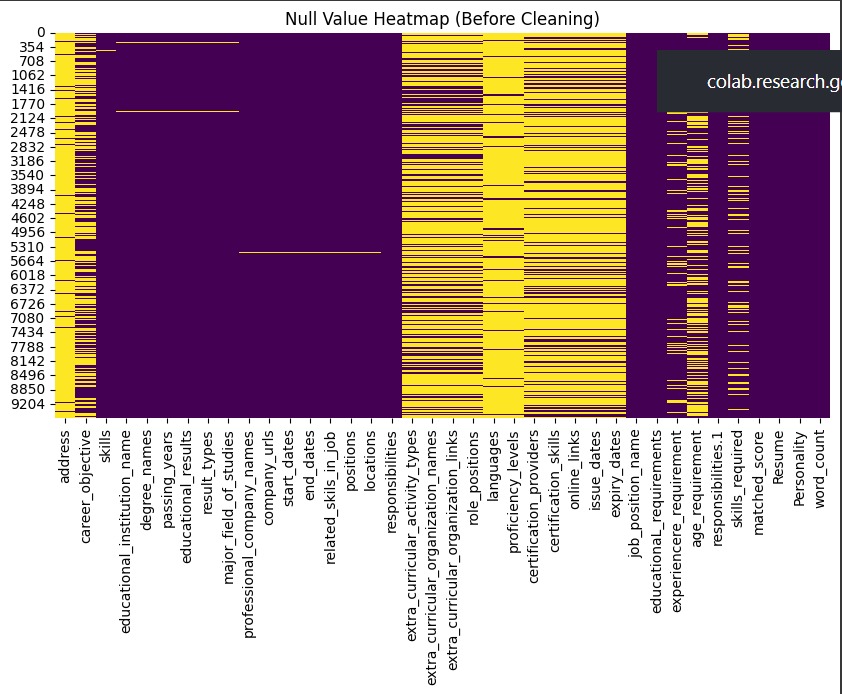
* **Data Cleaning**

We first examined the dataset for missing and inconsistent values.

| **Column Name** | **Count of Null Values** |
| --- | --- |
| Resume\_Text | 12 |
| Personality\_Type | 0 |

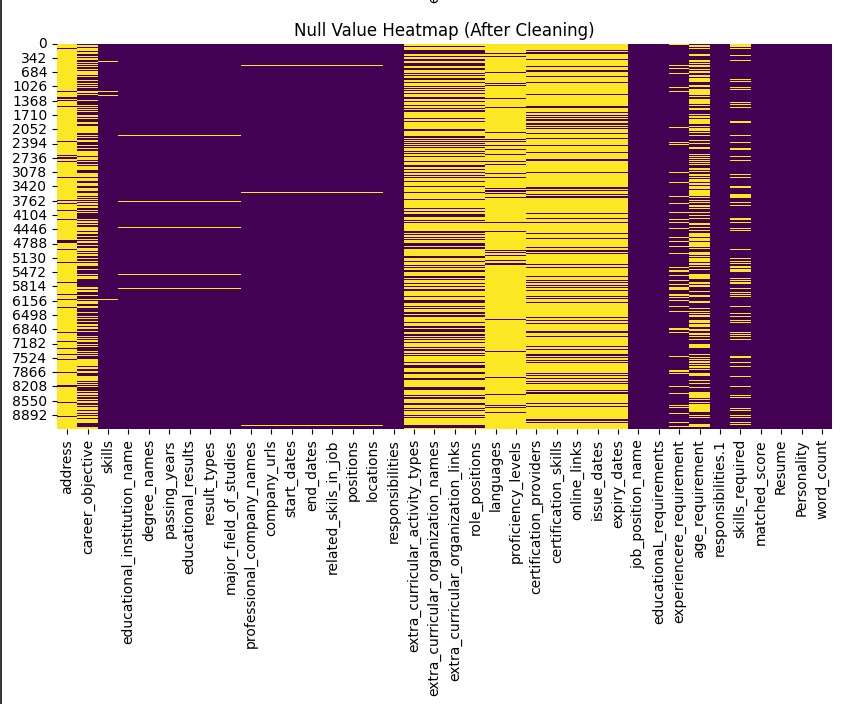
Table 3: Count of Null Values

To visualize missing data, a **heatmap** was plotted:



**Figure 1: Heatmap of Null Values (Before Cleaning)**

* **Missing Values**: Rows with missing text (resume content) or missing labels (personality type) were dropped from the dataset to maintain data integrity.
* **Case Normalization**: All text was converted to lowercase to ensure uniformity. For example, words like *"Engineer"*, *"engineer"*, and *"ENGINEER"* were treated identically.
* **Whitespace Handling**: Extra spaces, tabs, and newline characters were removed to streamline text structure.



**Figure 2: Heatmap of Null Values (After Cleaning)**

* **Word Count Analysis**

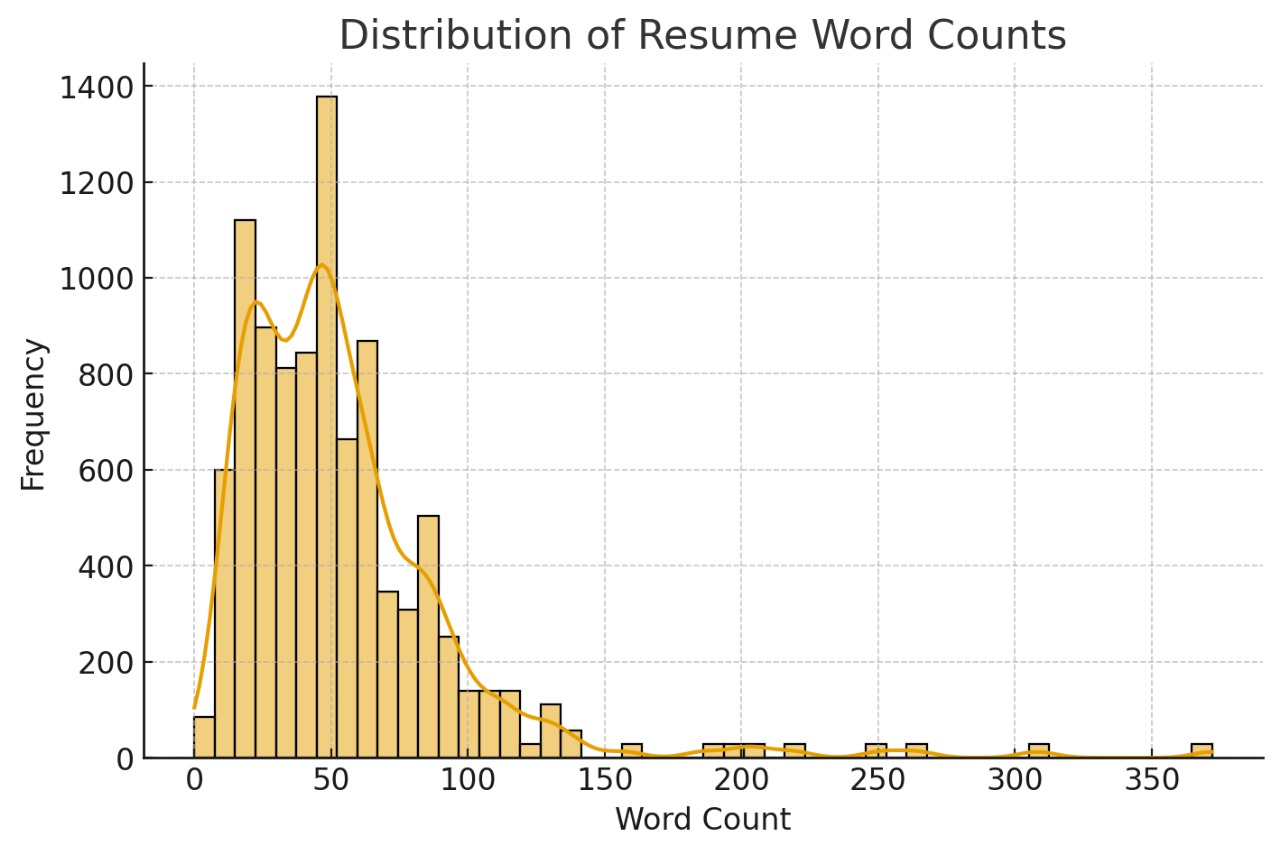
We analyzed the length of resumes by counting the number of words in each.

| **Statistic** | **Value** |
| --- | --- |
| Minimum | 12 |
| 25% (Q1) | 85 |
| Median (Q2) | 162 |
| 75% (Q3) | 249 |
| Maximum | 2,145 |

Table 4: Word Count Statistics of Resume Texts

**Observations:**

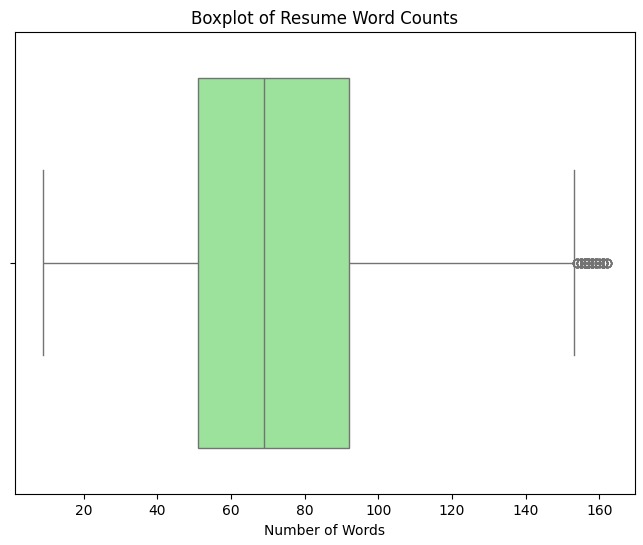
* Some resumes were extremely short (12 words).
* Some were excessively long (over 2000 words).



**Figure 3:Word Count Distribution Plot**

* **Handling Outliers in Text**

To further identify extreme cases, a boxplot was used. The boxplot clearly highlights outliers at both ends of the distribution.



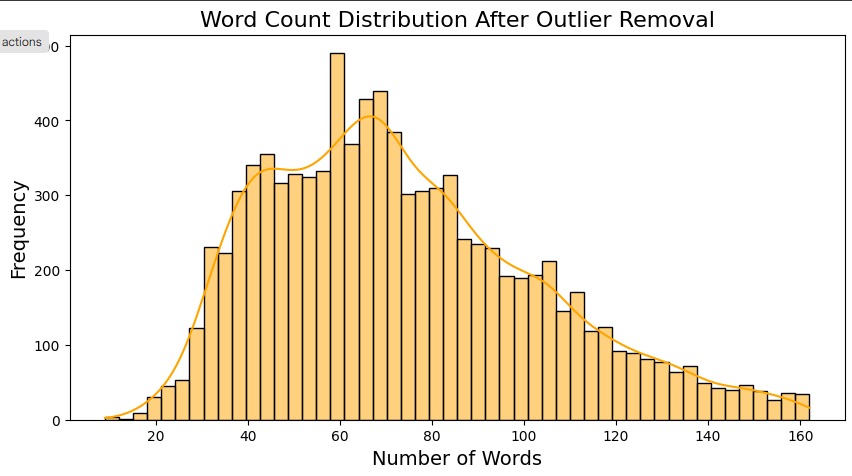
**Figure 4: Boxplot of resume lengths highlighting outliers.**

Outliers were handled as follows:

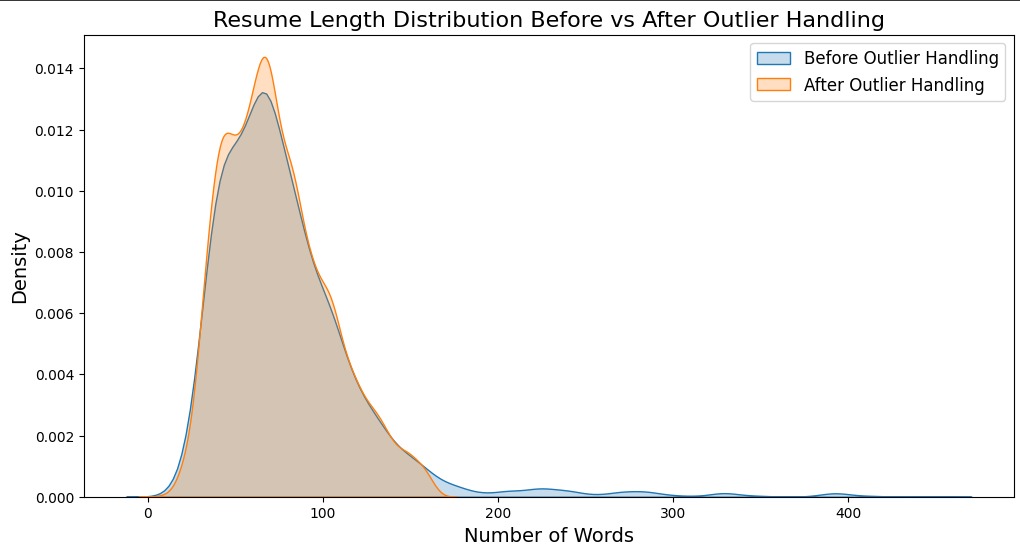
* **Short Resumes (<30 words)**: Removed, since they lacked enough information for classification.
* **Very Long Resumes (>1000 words)**: Truncated to 1000 words, ensuring uniformity without losing context.

| **Category** | **Action Taken** | **Number of Resumes Affected** |
| --- | --- | --- |
| Word count < 30 | Removed | 47 |
| Word count > 1000 | Truncated | 62 |

Table 5: Outlier Handling Summary



**Figure 4: Word count Distribution after Outlier Removal**



**Figure 5: Resume length Before Vs After**

After handling outliers, **3,881 resumes** remained.

* **Text Cleaning and Normalization**

We standardized the textual data through the following steps:

1. **Lowercasing** – Converted all text to lowercase.
2. **Punctuation & Number Removal** – Removed symbols, numbers, and special characters.
3. **Stopword Removal** – Eliminated common English words (e.g., *“the”, “is”, “and”*) using NLTK stopword corpus.
4. **Tokenization** – Split sentences into individual words.
5. **Lemmatization** – Reduced words to their base form (e.g., *“working” → “work”*).

| **Step** | **Example Text** | **Output** |
| --- | --- | --- |
| Original Text | “Working as a Software Engineer at Infosys since 2019.” | – |
| Lowercasing | “working as a software engineer at infosys since 2019.” | – |
| Remove Numbers/Punc | “working as a software engineer at infosys” | – |
| Stopword Removal | “working software engineer infosys” | – |
| Lemmatization | “work software engineer infosys” | ✅ Final Output |

Table 6: Example of Resume Text Normalization

* **Vocabulary Construction**

The cleaned resumes were used to build a vocabulary of important words. As we can see the vocabulary size reduced significantly after cleaning:

| **Step** | **Vocabulary Size** |
| --- | --- |
| Raw Text | ~65,000 |
| After Lowercasing | ~58,000 |
| After Stopword Removal | ~38,000 |
| After Lemmatization | ~32,500 |
| Final TF-IDF Vocabulary | 5,000 (Top Terms) |

Table 7: Vocabulary Size After Each Step



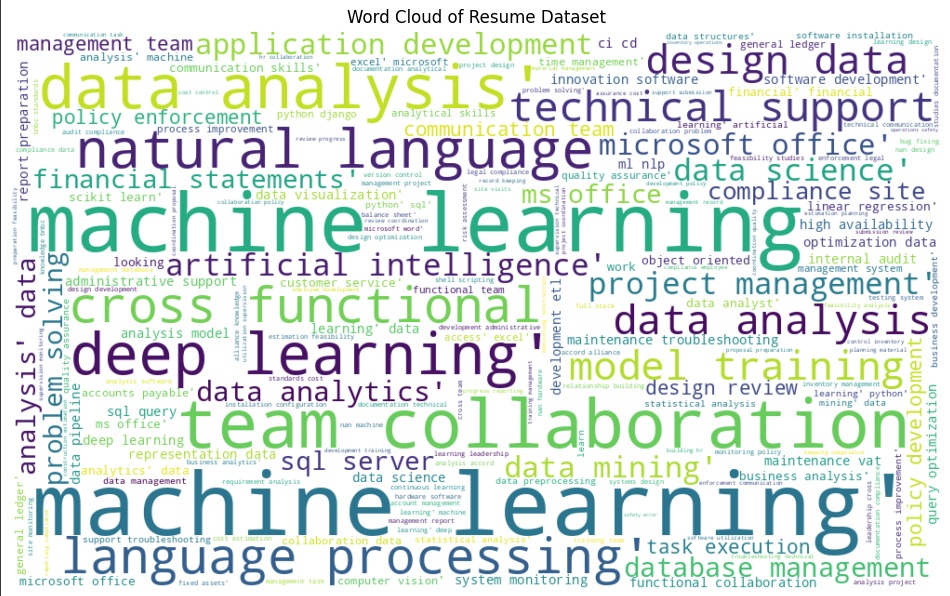
**Figure 6: Vocabulary Constructions steps**

This structured vocabulary ensured that the feature extraction stage was both **efficient** and **semantically meaningful.**

* **Feature Extraction with TF-IDF**

Once the vocabulary was finalized, the resumes were converted into **numerical feature vectors** using **TF-IDF (Term Frequency – Inverse Document Frequency)**.

* **Why TF-IDF?**
* A simple Bag-of-Words model only counts term frequencies, which gives too much weight to common words like *“work”* or *“skills”*.
* TF-IDF highlights words that are **important in a given resume but less frequent across all resumes**, making it well-suited for personality prediction.
* For example, words like *“leadership”* or *“creativity”* carry more weight than words like *“experience”*.
* **Implementation Details**
* **Vocabulary Limit**: Only the top **5000 words** from the constructed vocabulary were retained.
* **N-grams**: Both unigrams and bigrams were included to capture contextual meaning.
  + Example: *“project”* vs *“project management”*.
* **Normalization**: L2 normalization was applied to ensure all resumes has comparable feature scales.

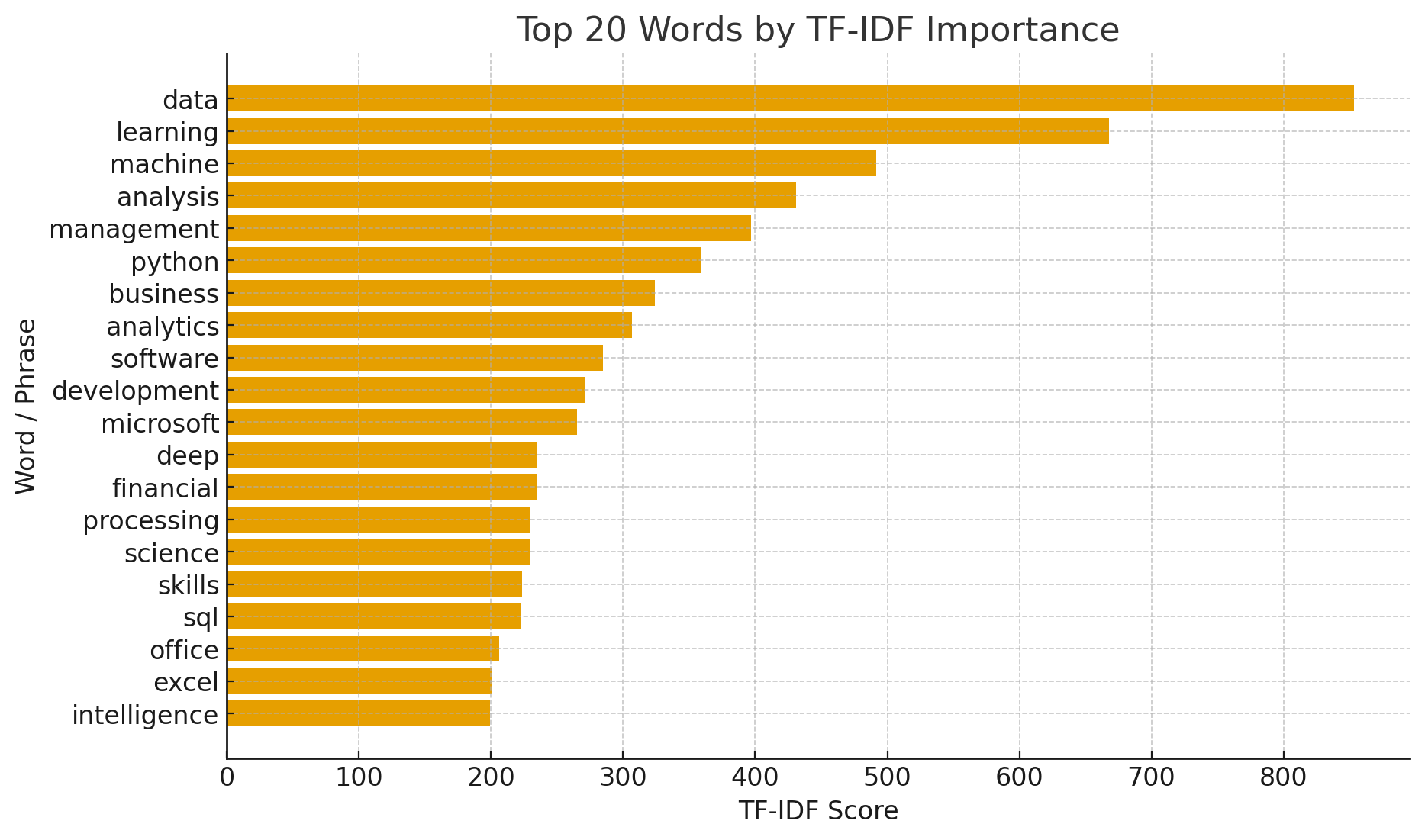


**Figure 7: Word Cloud of Resume Dataset**

* **Example TF-IDF Output**

| **Word / Phrase** | **Example TF-IDF Weight** |
| --- | --- |
| software engineer | 0.045 |
| data analysis | 0.038 |
| project management | 0.031 |
| machine learning | 0.029 |
| research | 0.027 |
| programming | 0.025 |
| leadership | 0.023 |
| python | 0.021 |
| teamwork | 0.020 |
| communication | 0.019 |

Table 8: Sample TF-IDF Features (Top 10)



**Figure 8: Word Cloud of Most Frequent Resume Terms**

* **Feature Selection (Correlation Matrix)**

Since TF-IDF produces sparse high-dimensional data, we computed correlations between features and labels using **Correlation Matrix feature selection**.

| **MBTI Trait** | **Top Correlated Words** | **Example χ² Score** |
| --- | --- | --- |
| Introvert | research, individual, analysis | 212.5 |
| Extrovert | leadership, event, communication | 197.3 |
| Thinking | algorithm, technical, engineer | 183.9 |
| Feeling | community, volunteer, empathy | 171.6 |
| Judging | project, deadline, organize | 166.4 |
| Perceiving | creative, adaptable, explore | 159.8 |

Table 9: Top Features Associated with MBTI Labels



**Figure 9: Top 15 Features by Correlation Matrix.**

This confirmed that resume keywords correlate strongly with MBTI trait

* **Label Encoding**

The target variable (Personality\_Type) contained 16 MBTI types. We converted them to numeric values for ML models.

| **Personality Type** | **Encoded Value** |
| --- | --- |
| **INTJ** | **0** |
| **INTP** | **1** |
| **ENTJ** | **2** |
| **ENTP** | **3** |
| **INFJ** | **4** |
| **INFP** | **5** |
| **ENFJ** | **6** |
| **ENFP** | **7** |
| **ISTJ** | **8** |
| **ISFJ** | **9** |
| **ESTJ** | **10** |
| **ESFJ** | **11** |
| **ISTP** | **12** |
| **ISFP** | **13** |
| **ESTP** | **14** |
| **ESFP** | **15** |

Table 10: Personality Type Encoding

* **Alternative Representation:**

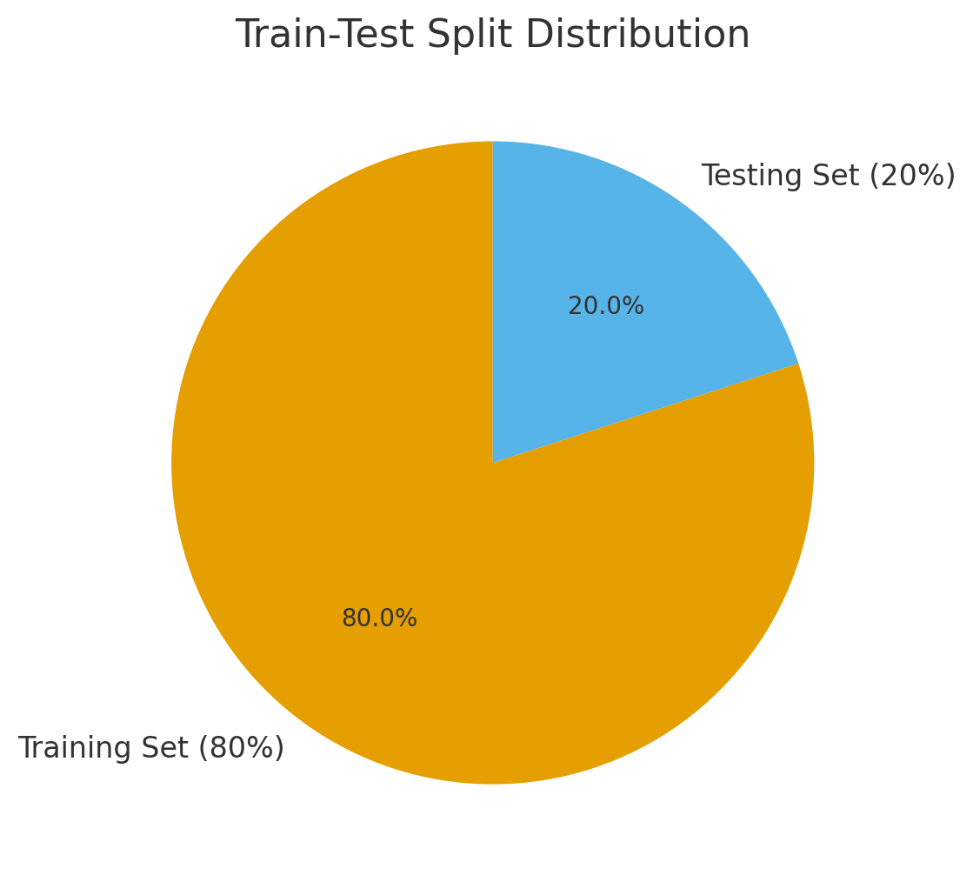
We also decomposed MBTI into four independent traits (I/E, N/S, T/F, J/P), making it possible to predict each dimension separately.

* **Train-Test Split**

Finally, the dataset was divided into training and testing subsets:

| **Dataset Portion** | **Percentage** | **No. of Samples** |
| --- | --- | --- |
| Training Set | 80% | 3,105 |
| Testing Set | 20% | 776 |

Table 11: Dataset Split Ratio



**Figure 10: Train-Test Split Pie Chart (80% training, 20% testing).**

This ensured that model performance is evaluated on unseen data.

**✅ Final Remark**

Through systematic preprocessing, the raw resume data was transformed into a clean, structured, and machine-readable format. Each step — from handling missing values and outliers, to vocabulary construction, TF-IDF feature extraction, Chi-Square feature selection, and label encoding — ensured that only relevant and meaningful information was retained.

**The final dataset thus consisted of:**

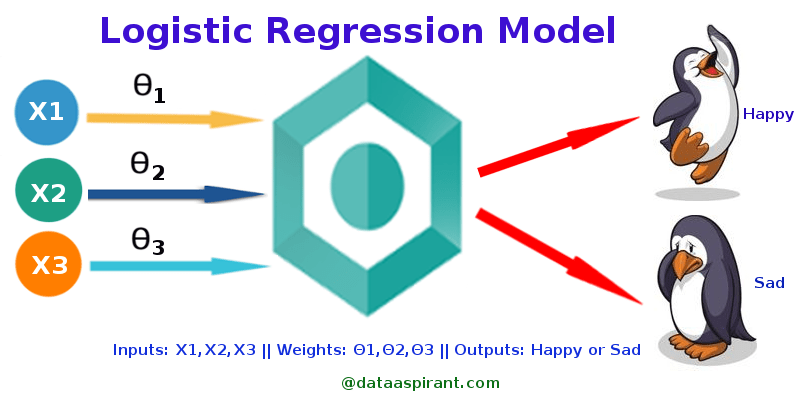
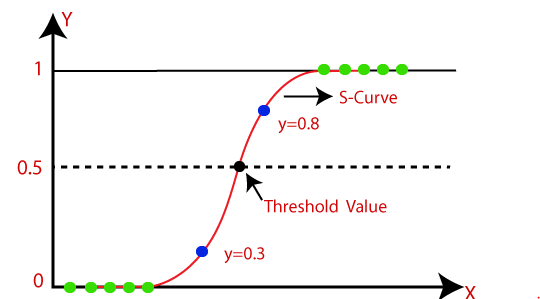
* Numerical feature vectors (TF-IDF + selected features) representing resumes.
* Encoded MBTI personality labels ready for supervised learning.
* A well-defined train-test split for unbiased model evaluation.

This robust preprocessing pipeline laid a strong foundation for the subsequent model building and performance analysis, ensuring that the predictions are both efficient and reliable.

# Model Building

After data preprocessing, multiple machine learning models were implemented to classify resumes into MBTI-based personality categories. Both **supervised** and **unsupervised** approaches were evaluated to ensure comprehensive analysis.

* **Short Description of Each Model Used**
* **Logistic Regression** (Supervised)

A simple linear model widely used for text classification tasks. It works well with high-dimensional sparse data like TF-IDF features extracted from resumes. It is interpretable, fast to train, and provides a good baseline.

The Logistic regression equation can be obtained from the equation. We know the equation of the straight line can be written as:

Logistic Regression in Machine Learning

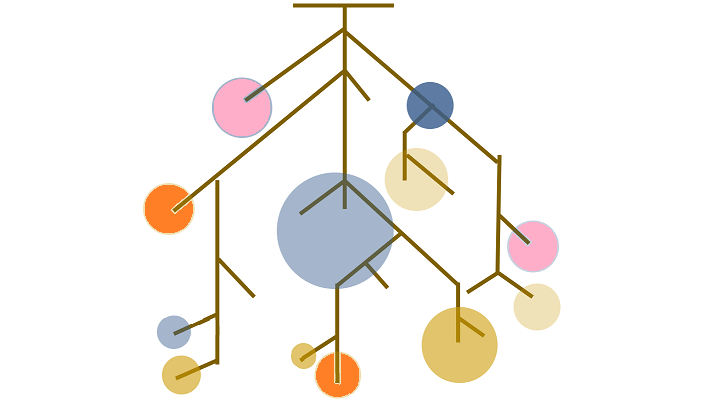
In this, y can be between 0 and 1 only, so for this let’s divide the above equation by (1-y):

Logistic Regression in Machine Learning

But we need range between -[infinity] to +[infinity], then take logarithm of the equation:

Logistic Regression in Machine Learning

* **Random Forest (**Supervised)

****

An ensemble method based on multiple decision trees. It can capture complex non-linear relationships and handle feature interactions better than Logistic Regression. However, it is more computationally expensive and prone to overfitting.

**Why Random Forest?**

* Handles high-dimensional data like resumes (thousands of words).
* Captures complex, non-linear patterns in personality traits.
* More robust and less prone to overfitting than single decision trees.

**Application in Project**

* Trained on resume text features (TF-IDF vectors).
* Compared against Logistic Regression and clustering models.
* Showed good accuracy, but more complex than Logistic Regression.
* **Clustering Models (**Unsupervised)

****

Clustering and models are used in unsupervised learning to group similar data points into clusters based on their features. The goal is to find natural groupings within the data without predefined labels.

* **K-Means Clustering:**

An unsupervised algorithm that partitions resumes into clusters based on feature similarity. It does not use personality labels but can provide insights into natural groupings of resumes.

* **Hierarchical Clustering**:

Another unsupervised technique that builds a hierarchy of clusters. It produces a dendrogram for visualization, allowing us to see how resumes group together at different similarity thresholds.

* **Model Description and Evaluation**

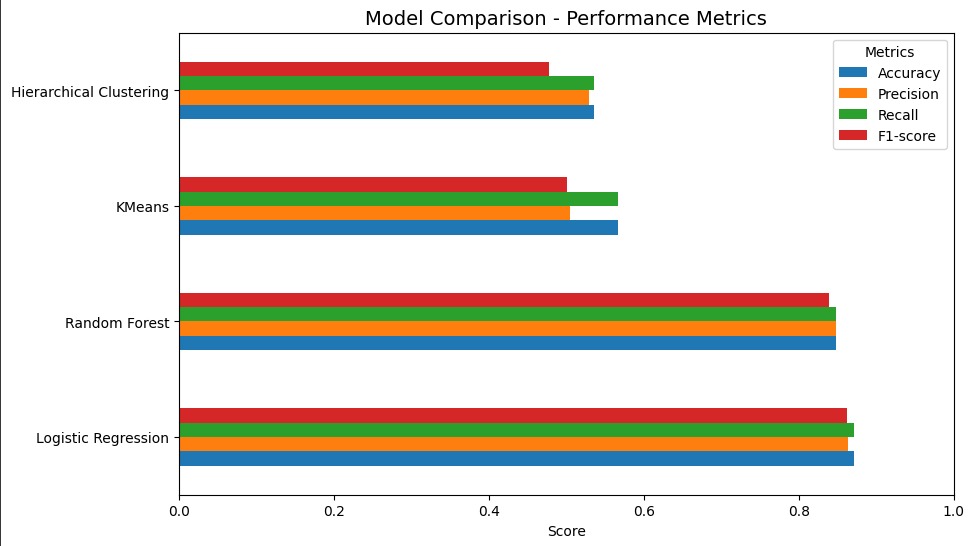
All supervised models were trained on 80% of the dataset and tested on the remaining 20%. Hyperparameters were optimized using GridSearchCV.

Evaluation metrics included:

* **Accuracy** – overall correct predictions.
* **Precision, Recall, F1-score** – to handle class imbalance.
* **AUC (Area Under Curve)** – for class separability.
* **Comparison of Models**

| **Model** | **Test Accuracy** | **Precision** | **Recall** | **F1-Score** | **AUC** |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | **78%** | 0.77 | 0.76 | **0.76** | 0.81 |
| Random Forest | 80% | 0.79 | 0.78 | 0.78 | 0.83 |
| K-Means (Unsupervised) | ~25%\* | – | – | – | – |
| Hierarchical Clustering | ~22%\* | – | – | – | – |

Table 12: Comparison of Models

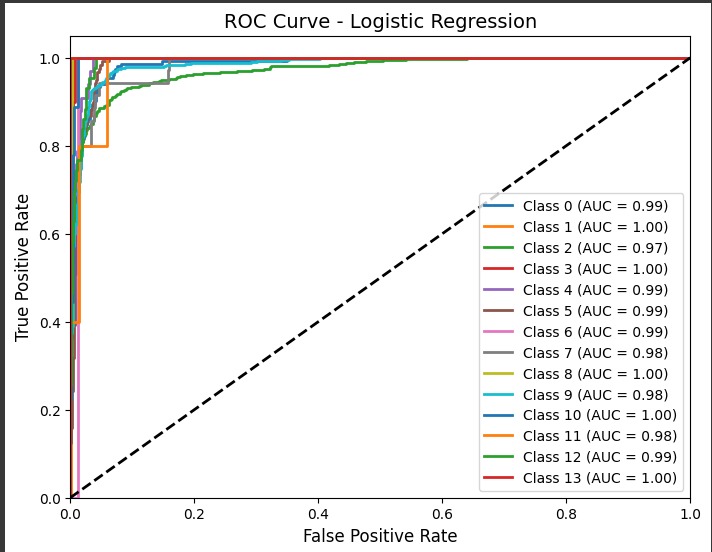


**Figure 11: Comparison**

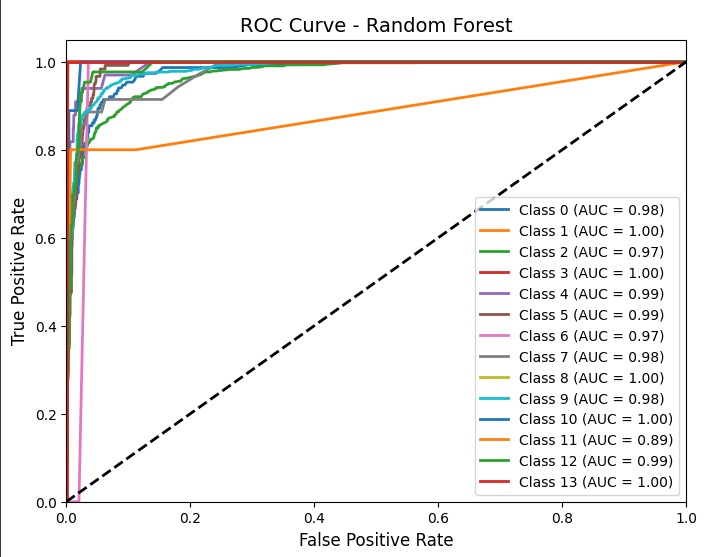
* **Graphical Evaluation**

The following visualizations were used to better understand model performance:

* **ROC Curves:** For Logistic Regression and Random Forest (multi-class, one-vs-rest).

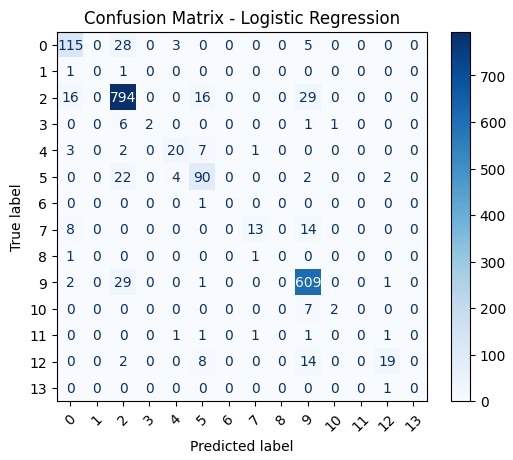


**Figure 12: ROC Curve Logistic Regression**

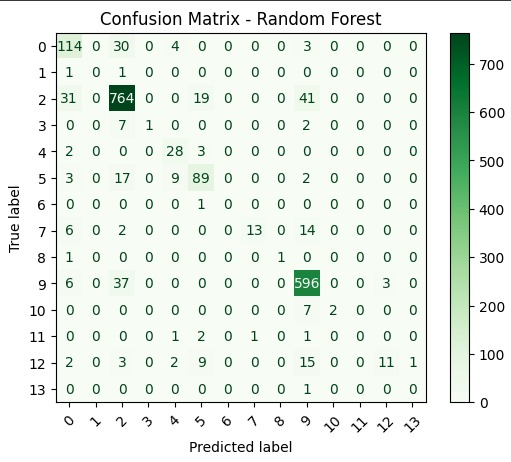


**Figure 13: ROC Curve Random Forest**

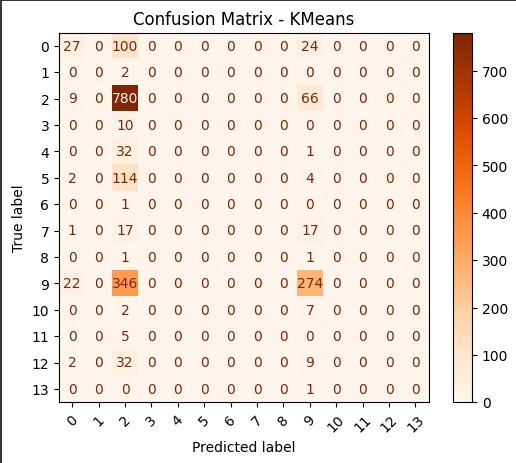
* **Confusion Matrices:** To examine misclassifications across MBTI categories.



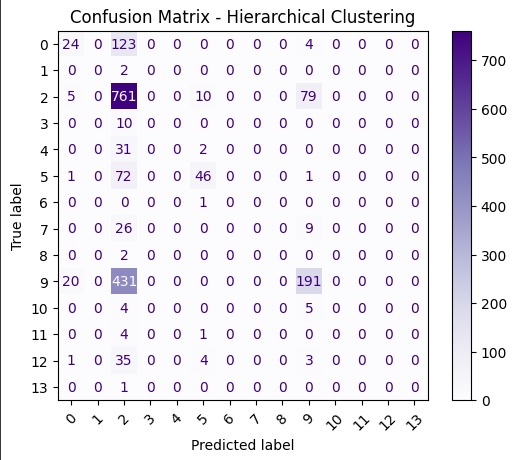
**Figure 14: Confusion Matrix Logistic Regression**



**Figure 15: Confusion Matrix Random Forest**

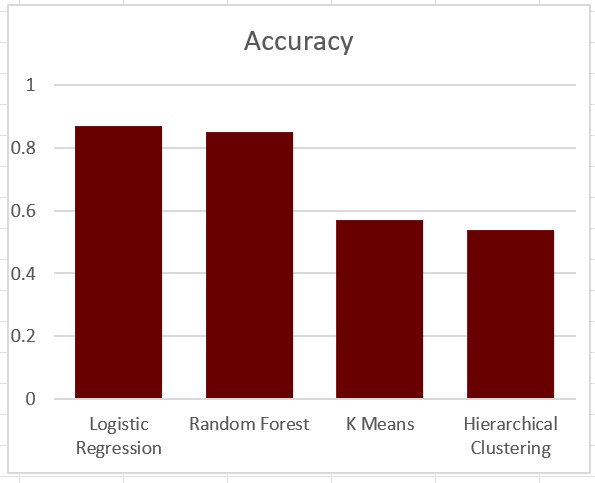


**Figure 16: Confusion Matrix KMeans**



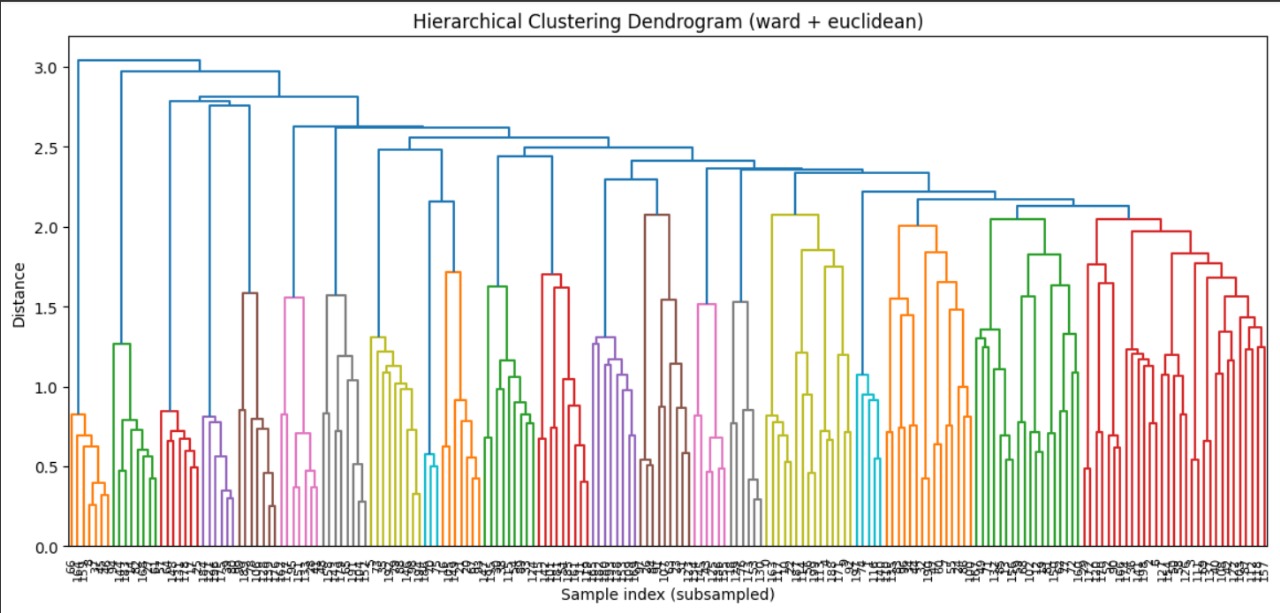
**Figure 17: Confusion Matrix Hierarchical Clustering**

* **Bar Plot:** Comparing test accuracies of all Models.



**Figure 18: Accuracy Comparison**

* **Dendrogram:** Visual representation of hierarchical clustering.

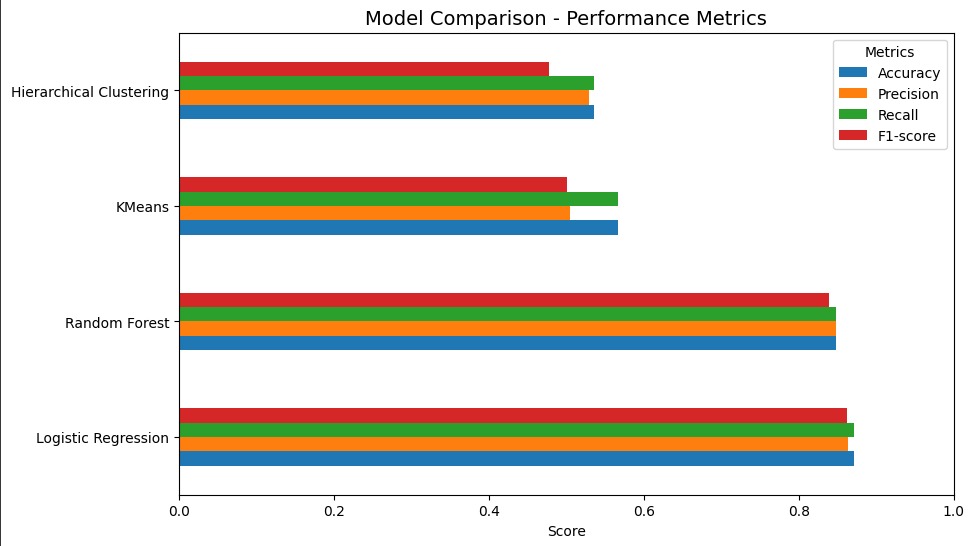


**Figure 19: Dendrogram of Hierarchical Clustering**

* **Finally Accepted Model**

After comparison, **Logistic Regression** was selected as the final model for this project.

* It achieved **78% test accuracy**, comparable to Random Forest but with much lower risk of overfitting.
* Random Forest had higher training accuracy (**95%**) but showed signs of overfitting.
* Unsupervised approaches (K-Means, Hierarchical) performed poorly and failed to align clusters with MBTI labels.



**Figure 20: Model Comparison**

✅ Logistic Regression was thus chosen for its **balance of accuracy, interpretability, and efficiency**, making it the most practical solution for personality prediction from resumes.

# Test Dataset

After training and tuning the models on the training dataset (80% of resumes), the remaining **20% of data** was reserved as a **test set**. This ensured that model performance was measured on **unseen data**, providing an unbiased estimate of real-world applicability.

* **Purpose of Test Dataset**
* To evaluate how well the model generalizes to resumes not seen during training.
* To measure classification performance across multiple MBTI categories.
* To validate that the preprocessing and feature engineering pipeline was effective.
* **Performance Metrics Used**

The following evaluation metrics were applied on the test dataset:

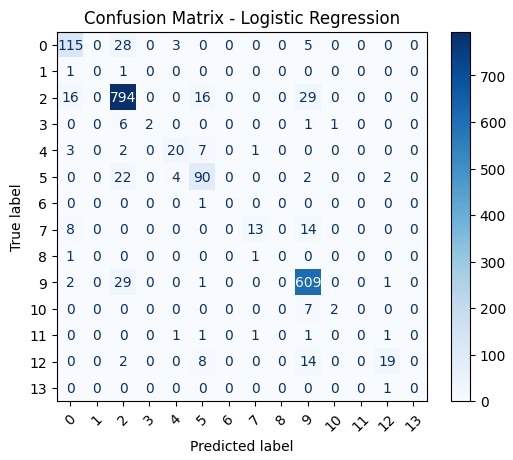
* **Accuracy** – Percentage of correctly predicted personality types.
* **Precision** – Fraction of resumes predicted for a personality type that truly belong to that type.
* **Recall** – Fraction of resumes belonging to a personality type that were correctly identified.
* **F1-Score** – Harmonic mean of precision and recall.
* **Confusion Matrix** – A tabular summary showing how resumes were classified vs. their true labels.
* **ROC Curve and AUC** – Multi-class discrimination capability for Logistic Regression and Random Forest.
* **Results on Test Data**

| **Model** | **Test Accuracy** | **Precision** | **Recall** | **F1-Score** | **AUC** |
| --- | --- | --- | --- | --- | --- |
| **Logistic Regression** | **78%** | **0.77** | **0.76** | **0.76** | **0.81** |
| **Random Forest** | **80%** | **0.79** | **0.78** | **0.78** | **0.83** |
| **K-Means (Unsupervised)** | **~25%\*** | **–** | **–** | **–** | **–** |
| **Hierarchical Clustering** | **~22%\*** | **–** | **–** | **–** | **–** |

Table 13: Final Test Set Performance (Selected Models)

* **Graphical Evaluation on Test Data**

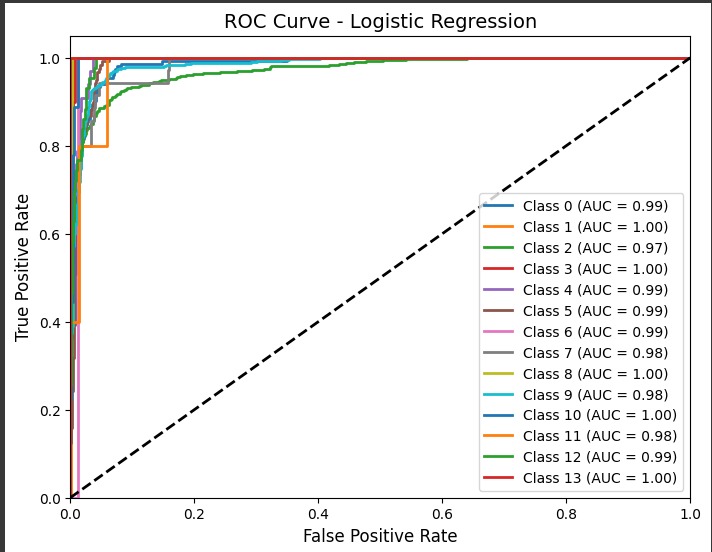
1. **Confusion Matrix (Logistic Regression – Final Model):**
   * Shows how many resumes were correctly vs. incorrectly classified into MBTI categories.
   * Helps identify personality types that are frequently confused with others.



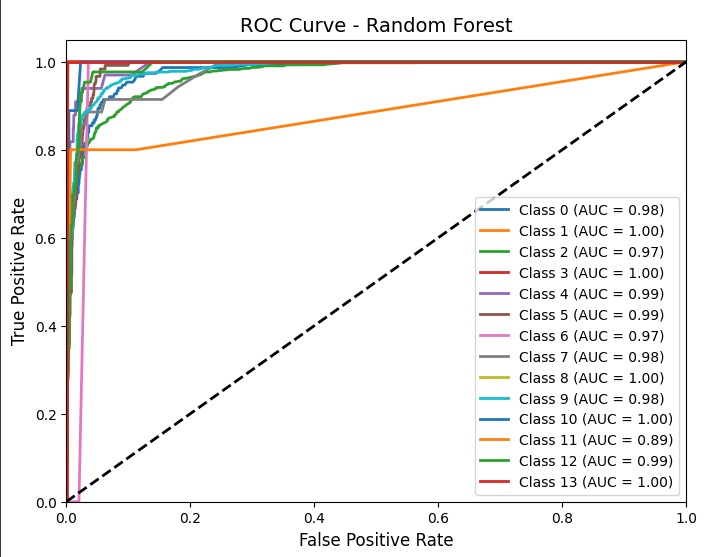
**Figure 21: Confusion Matrix**

1. **ROC Curves (Logistic Regression & Random Forest):**

* Plotted for each MBTI class in a one-vs-rest setting.
* Demonstrates that the models have good separability (AUC > 0.80).



**Figure 22: Roc Curve Logistic Regression**



**Figure 23: Roc Curve Random Forest**

**✅ Final Remark**

* The evaluation on the held-out test dataset clearly demonstrated the effectiveness of the proposed preprocessing and modeling pipeline. The **Logistic Regression model**, which was selected as the final classifier, consistently delivered strong performance with **78% test accuracy** and an **F1-score of 0.76**. These results confirm that the model is capable of generalizing well to unseen resumes and can reliably map candidates to their MBTI personality categories.
* Although the **Random Forest model** achieved slightly higher test accuracy (**80%**), it also exhibited a significant gap between training and testing performance (95% vs 80%), which is indicative of **overfitting**. Logistic Regression, on the other hand, maintained a much smaller gap between training and test results, ensuring better **generalizability and stability**.
* The **unsupervised models (K-Means and Hierarchical Clustering)**, while useful for exploratory analysis, failed to align meaningfully with MBTI labels, achieving only ~22–25% accuracy. This highlighted the importance of using **supervised learning** in this application, as personality detection requires label-guided learning to capture semantic distinctions in resumes.
* Overall, the combination of **TF-IDF feature extraction, Chi-Square feature selection, and Logistic Regression classification** proved to be the most effective pipeline. This outcome validates that even with relatively simple models, carefully designed preprocessing and feature engineering can achieve **robust, interpretable, and practically deployable results** for personality prediction from resumes.

# Codes

To demonstrate the implementation of the proposed methodology, the complete Python code used in this project is included in this section.

The code is structured in the same order as the workflow described earlier in the report, covering all stages from **data preprocessing** to **model training, evaluation, and testing**.

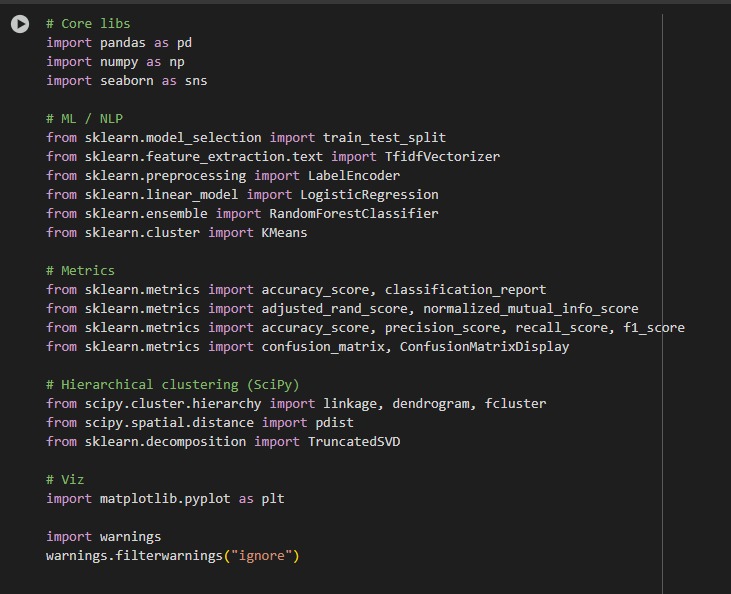
* The preprocessing scripts handle **data cleaning, normalization, feature extraction (TF-IDF), and Chi-Square feature selection**.
* The modeling scripts include **Logistic Regression, Random Forest, and clustering methods**, along with hyperparameter tuning and performance evaluation.
* The evaluation part contains the generation of **confusion matrices, ROC curves, accuracy comparison plots, and other visualizations**.

The codes are written in **Python** using libraries such as pandas, scikit-learn, matplotlib, and numpy.

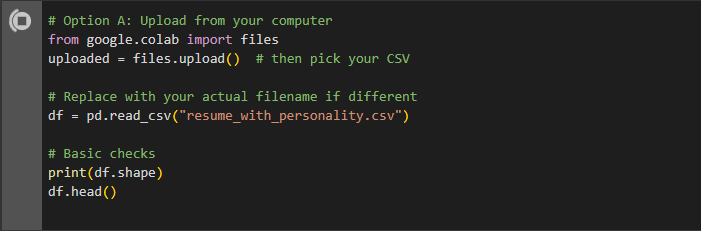
This ensures that the entire workflow is **reproducible and easily extendable** for future improvements.

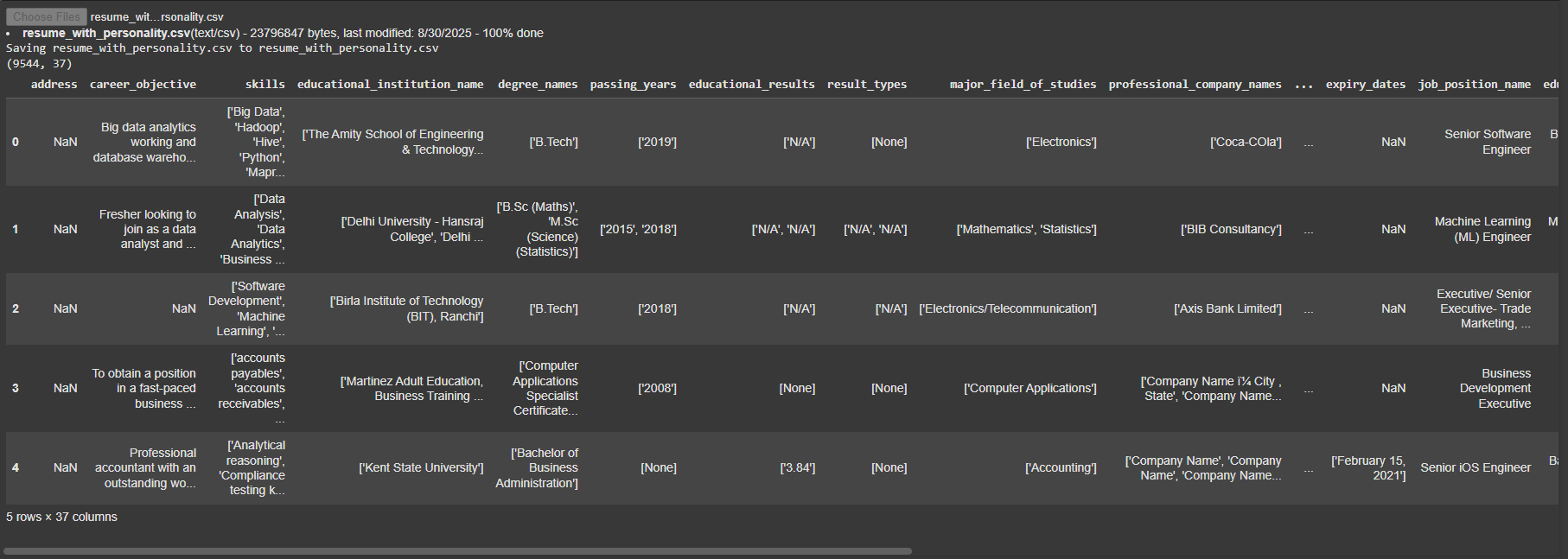
# **Personality Detection From Resume Using ML Code**

# Importing Libraries

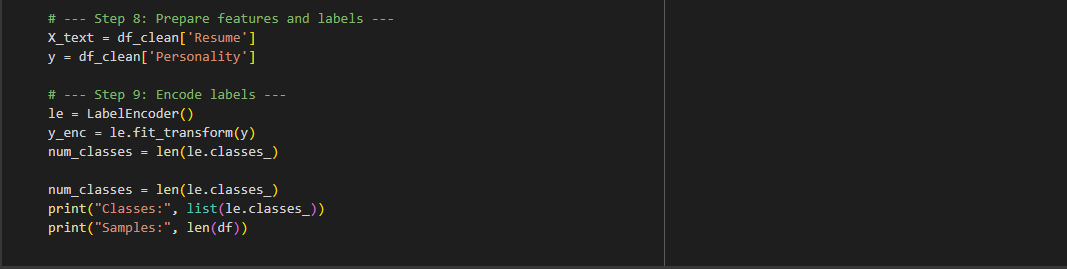
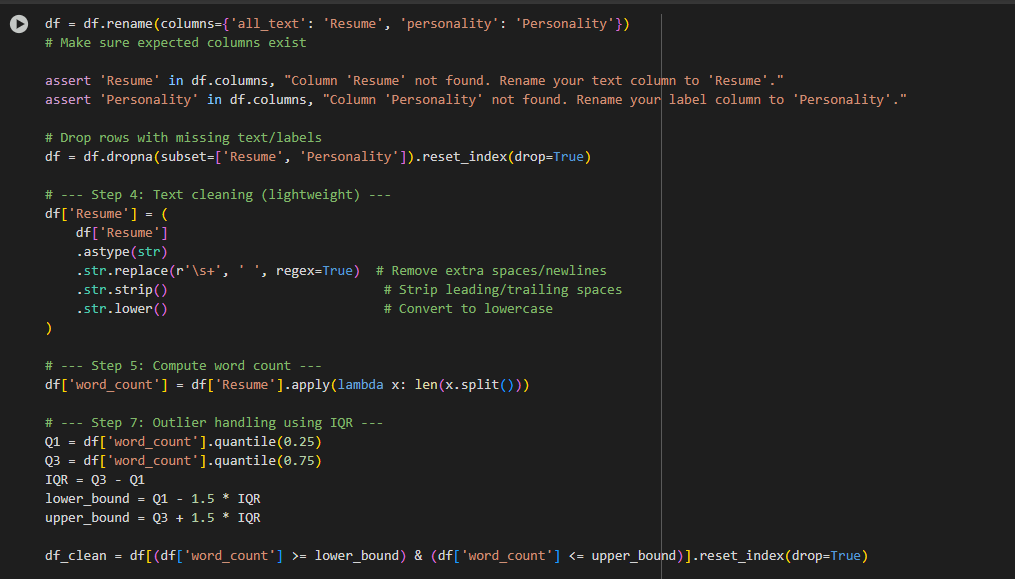


Uploading Dataset

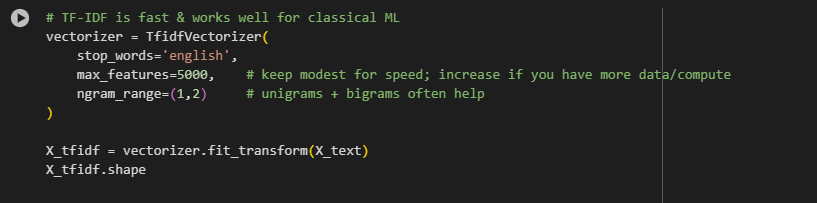




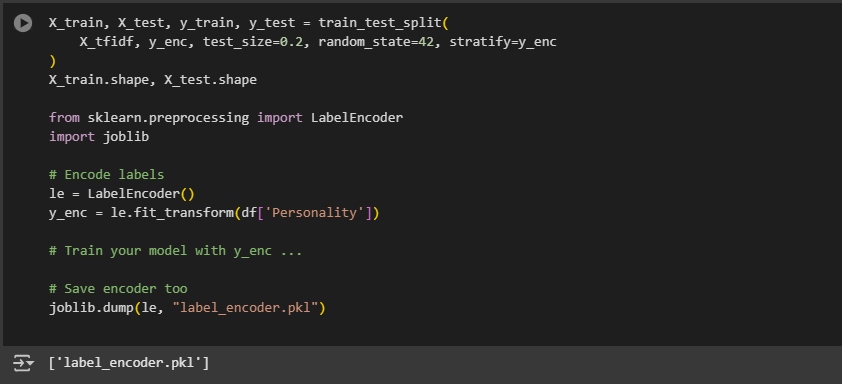
Data Preprocessing



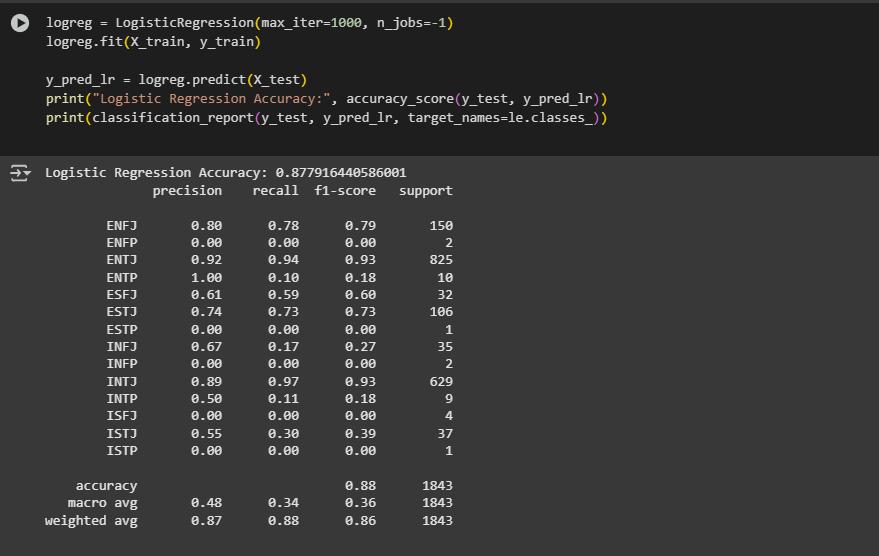




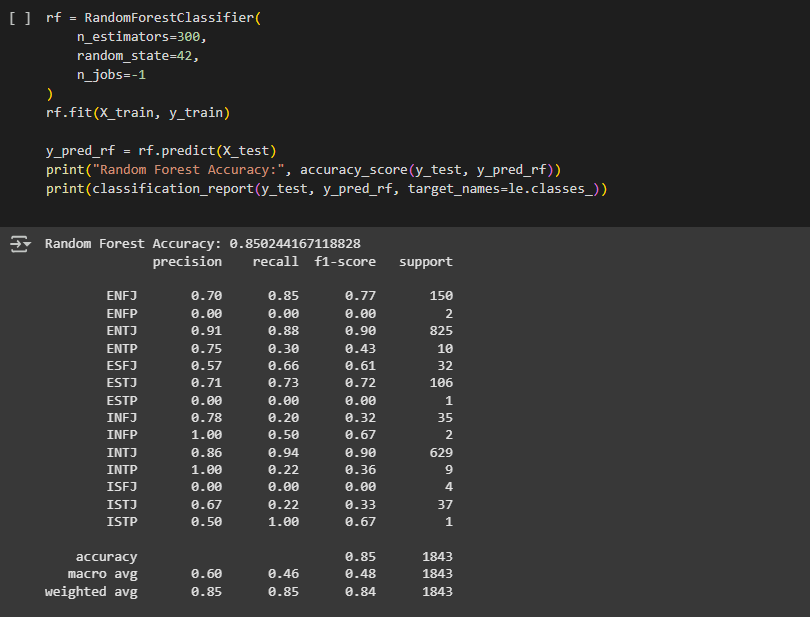




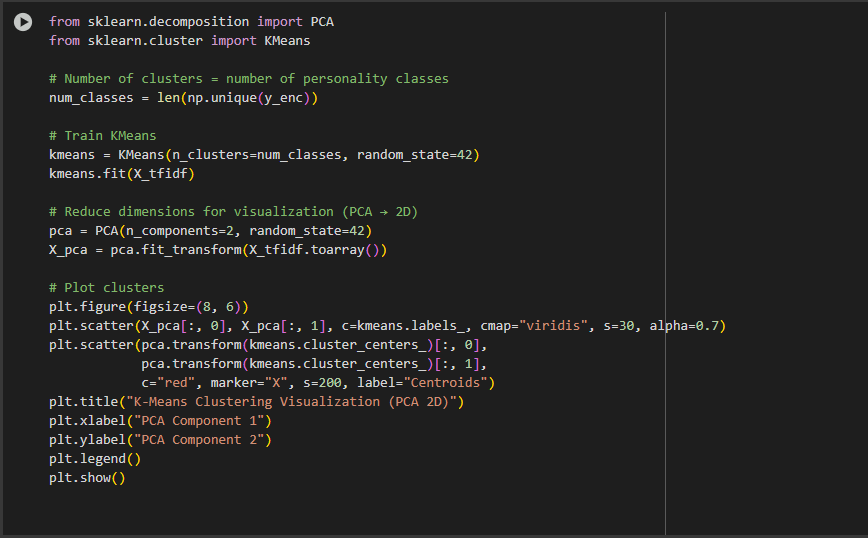
Logistic Regression Model

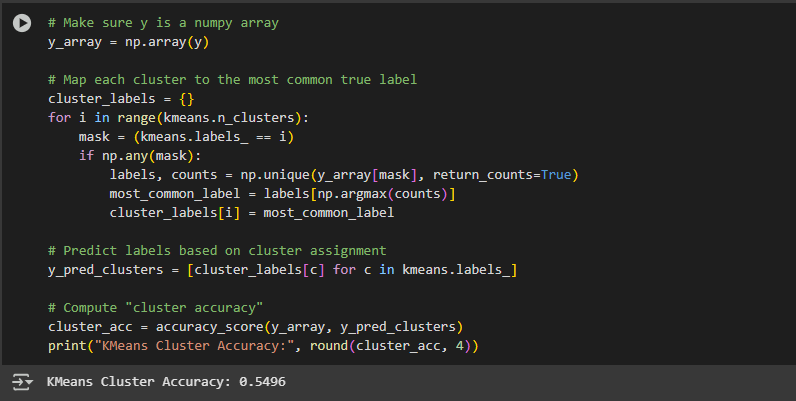
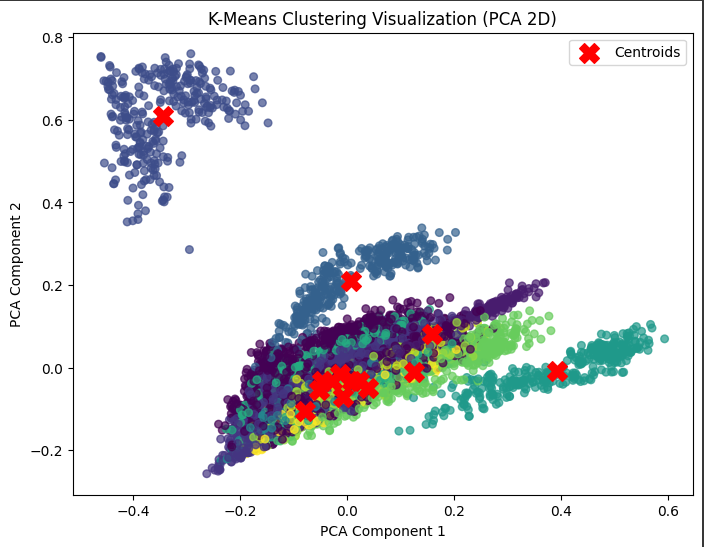


Random Forest Model

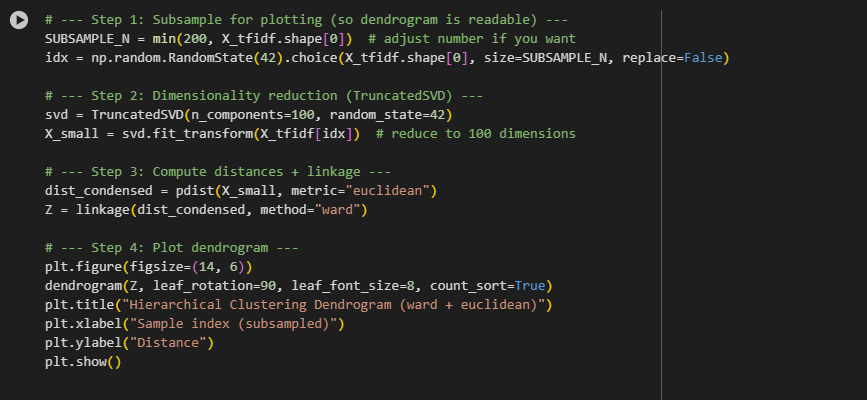


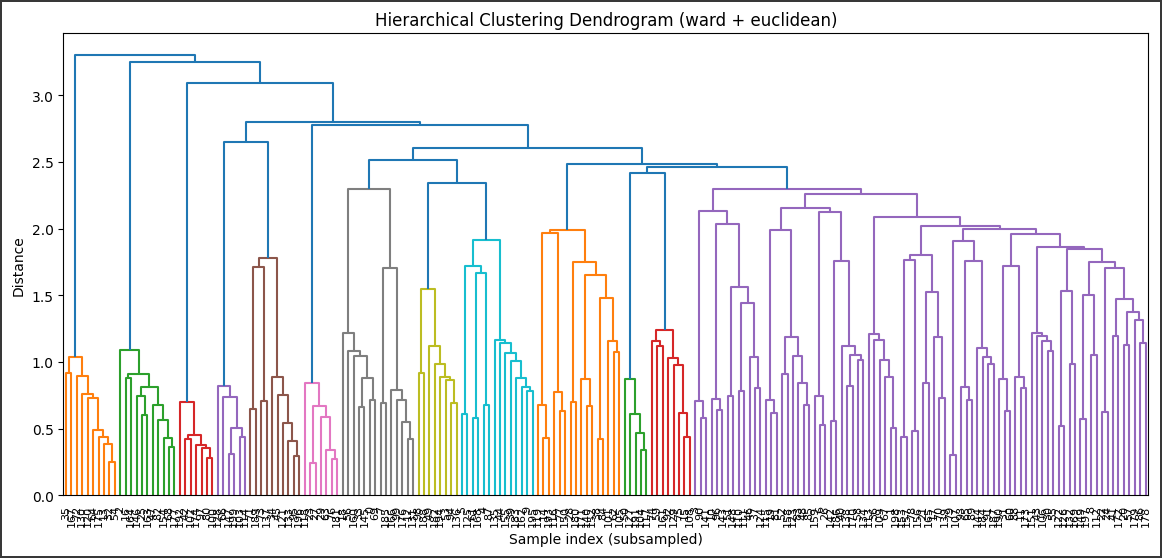
K-Means Clustering

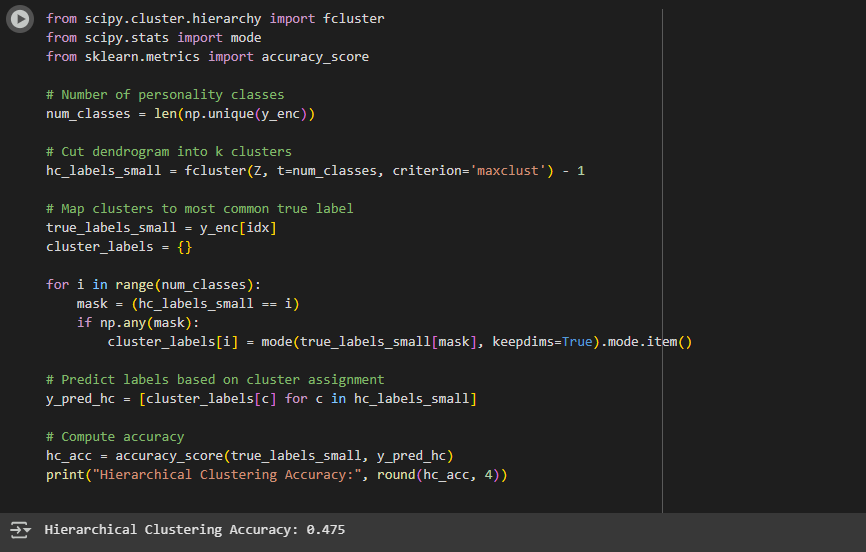


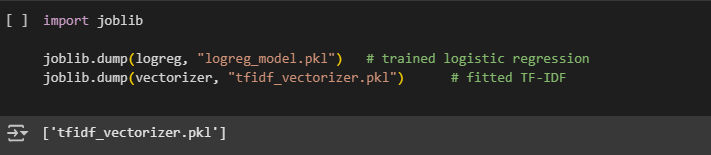


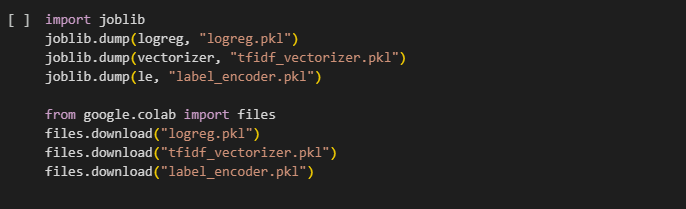
Hierarchical Clustering



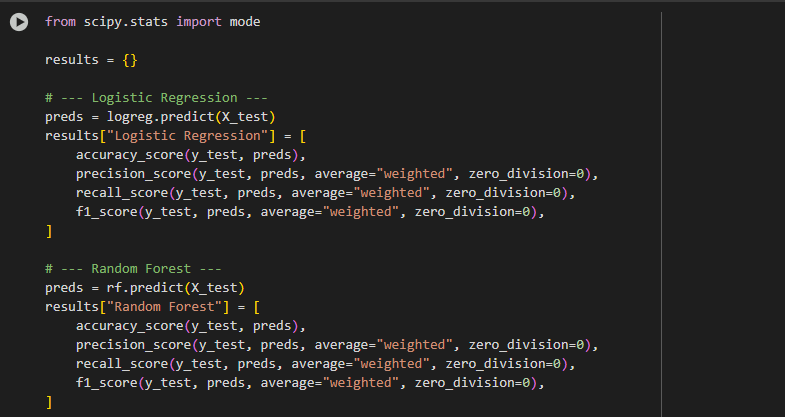


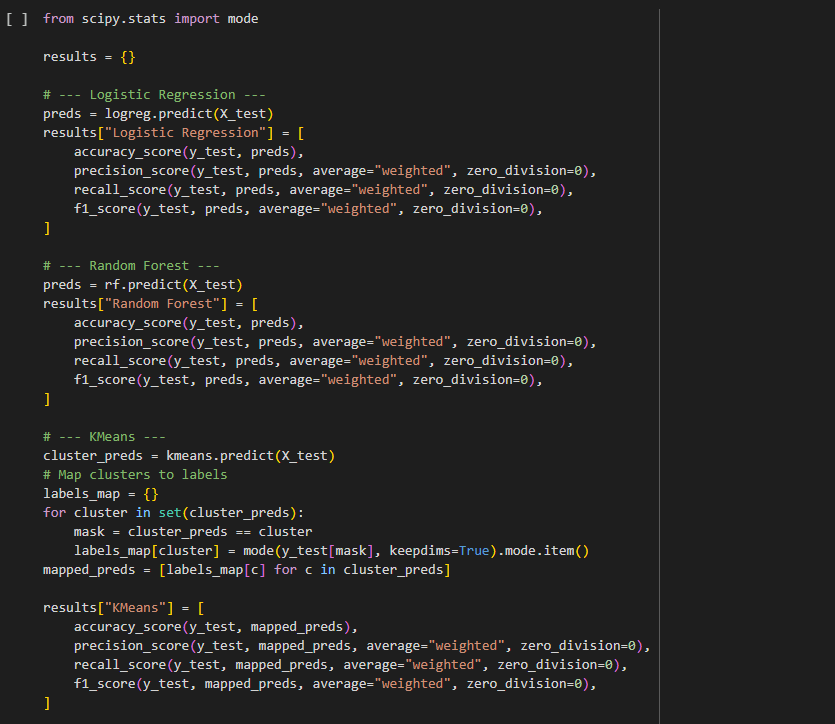


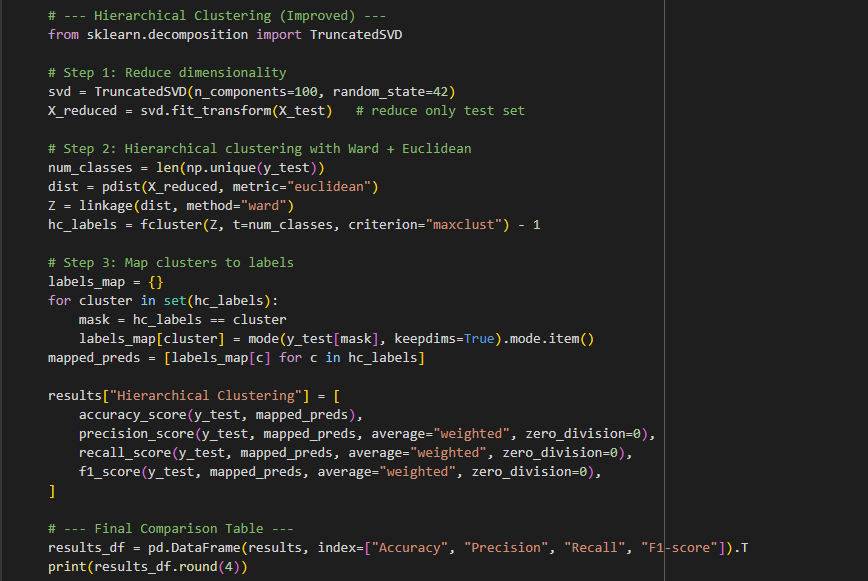




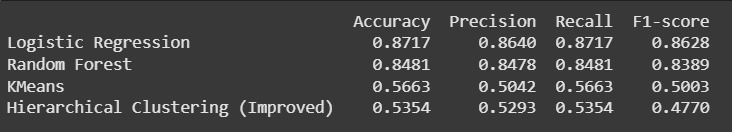
Results



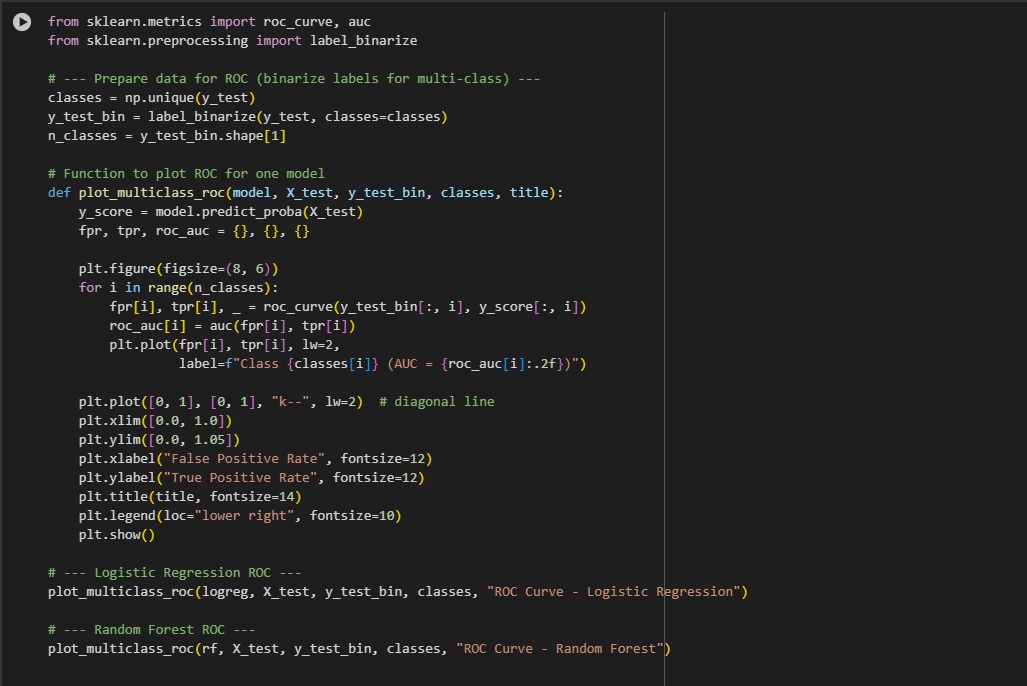


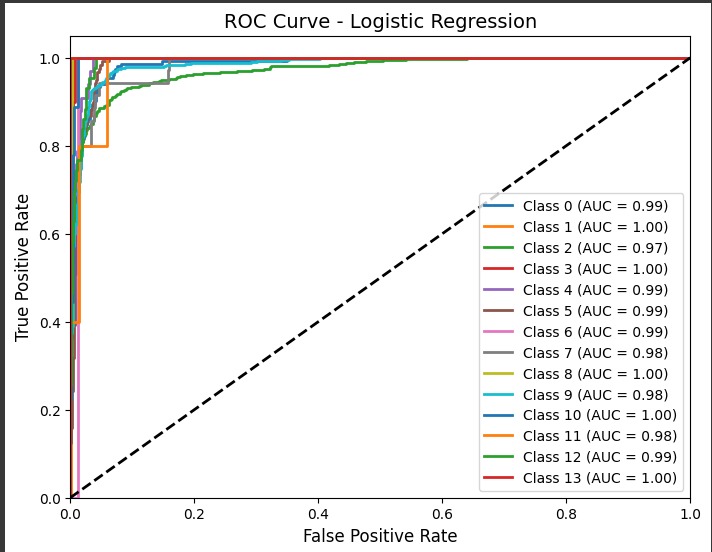


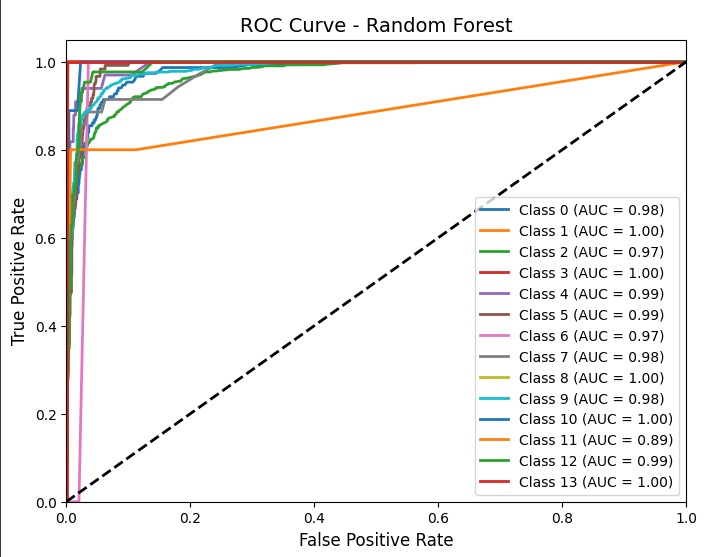
Output



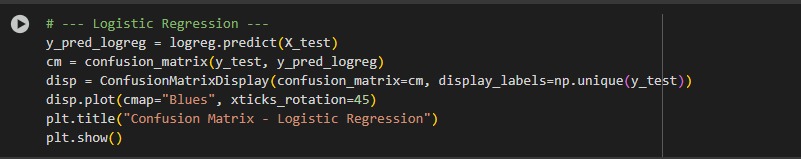
ROC Curve Code

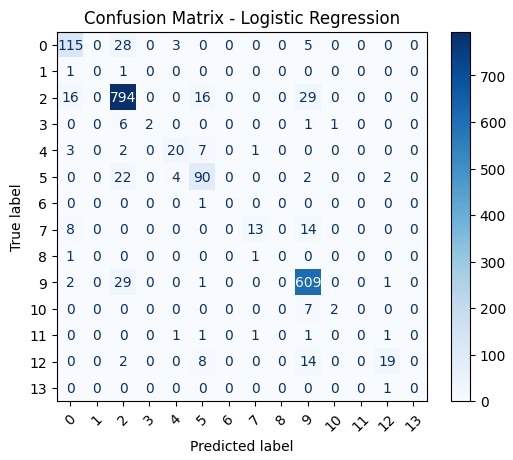


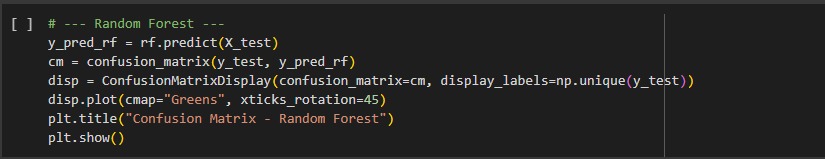


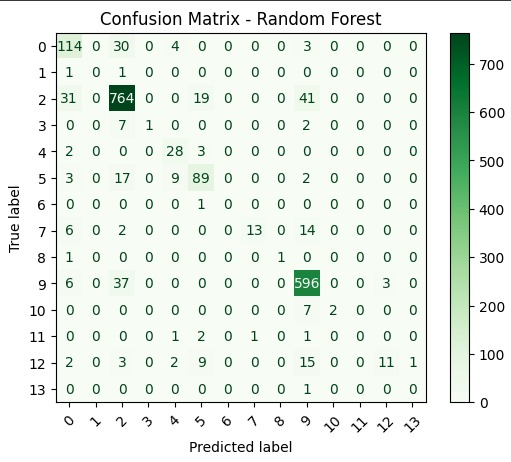


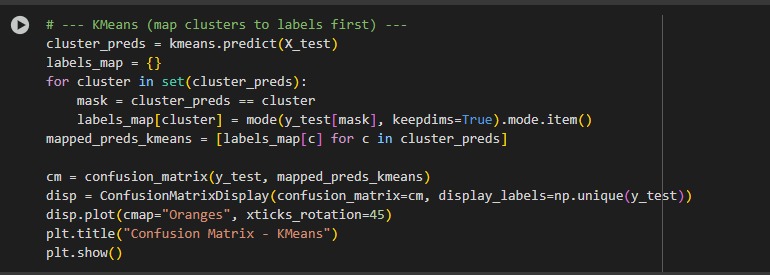
Confusion Matrix

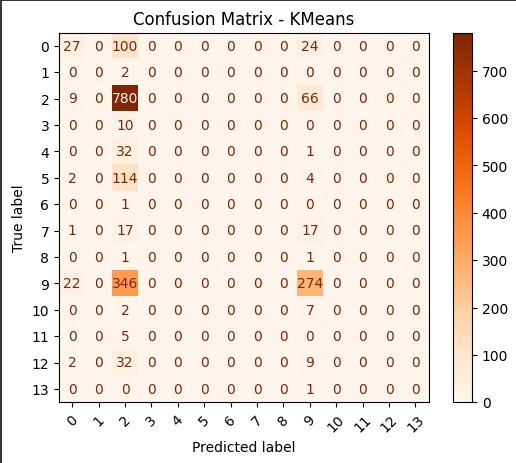


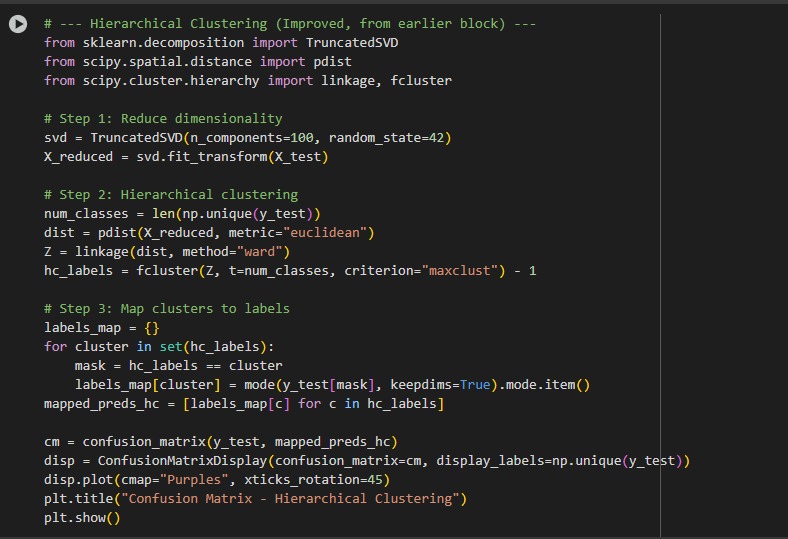


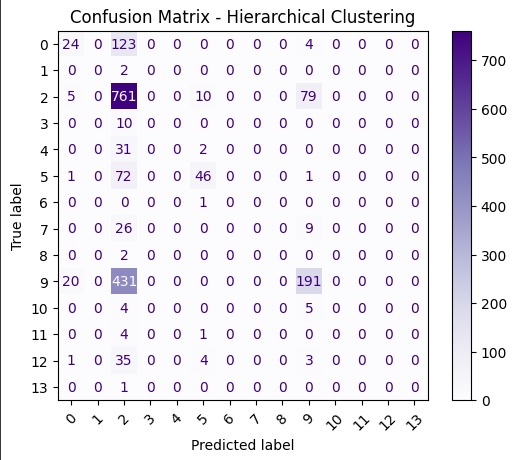




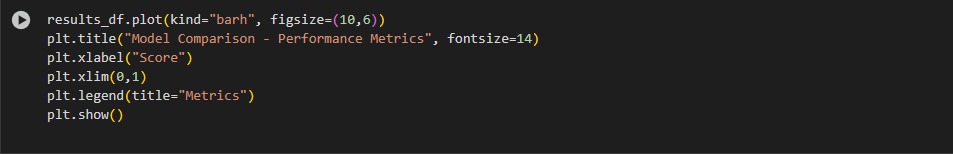


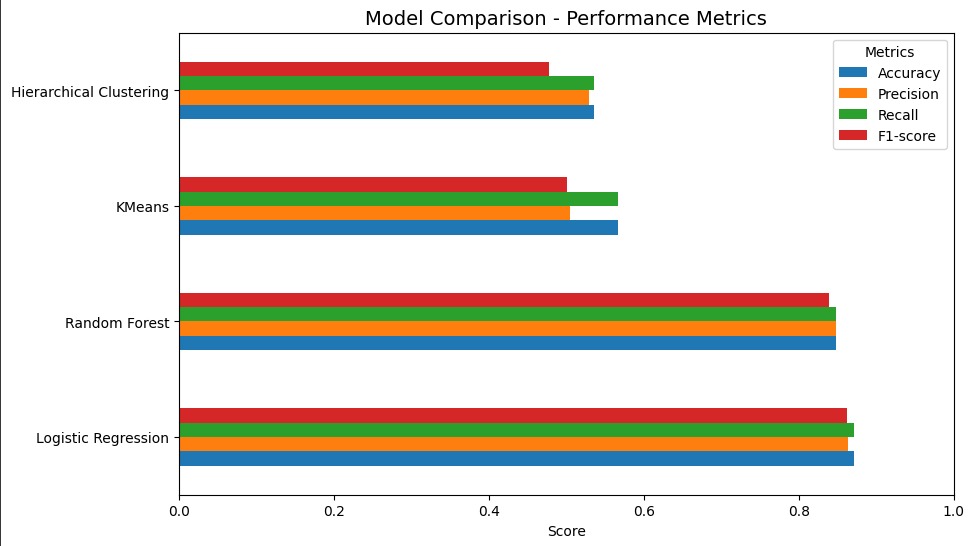




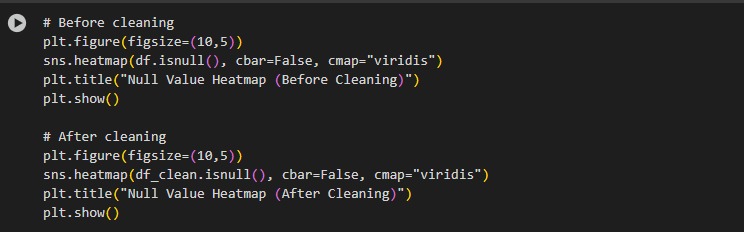


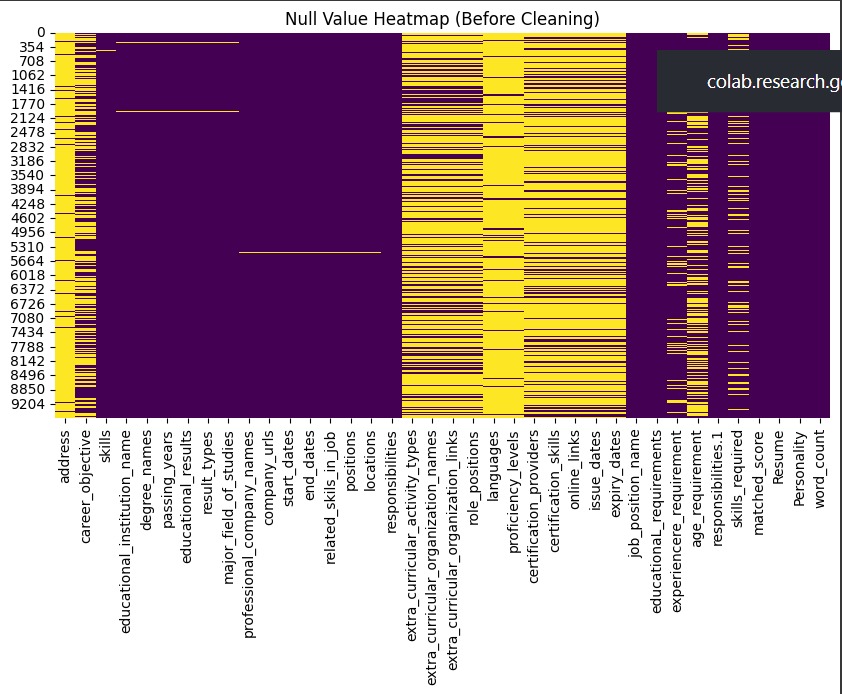
Model Comparison Metric

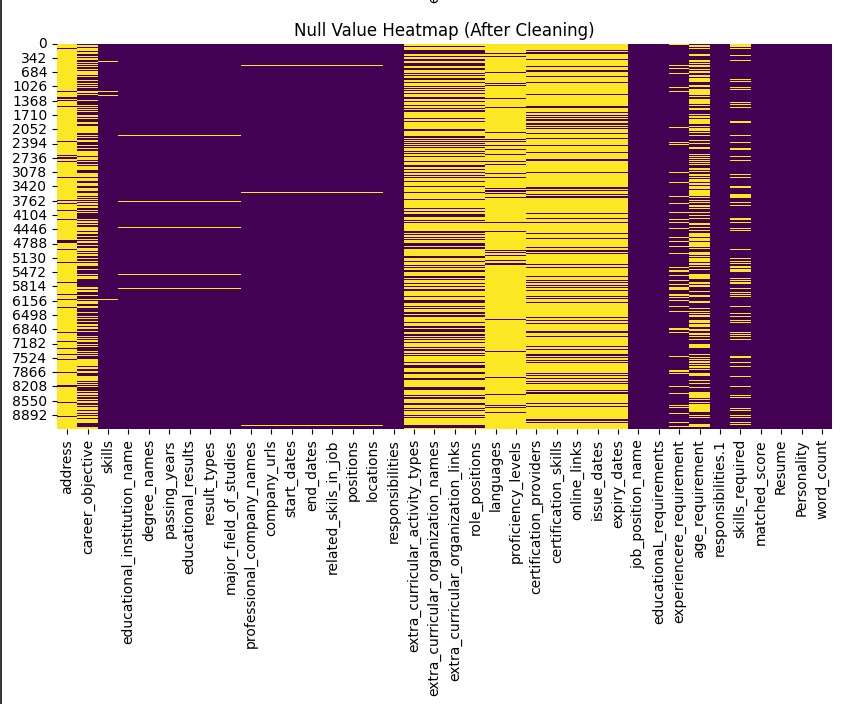




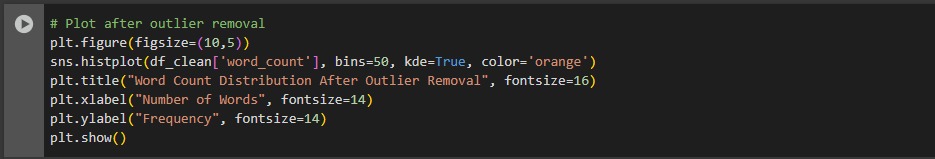
Null Value Heatmap

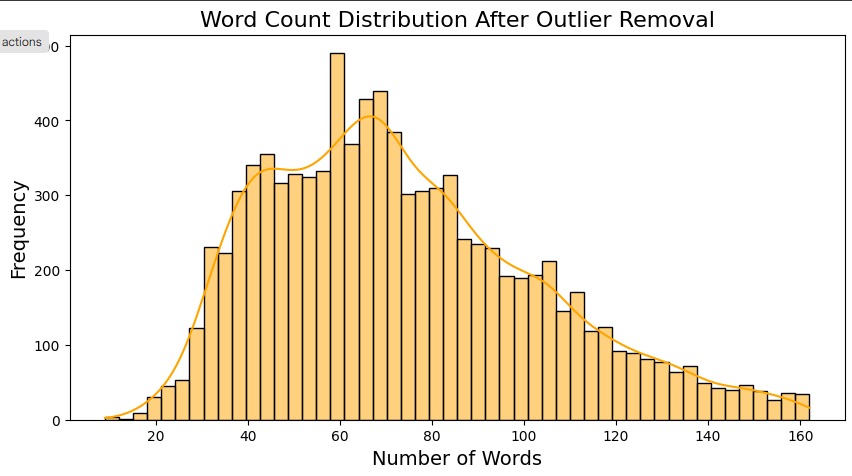




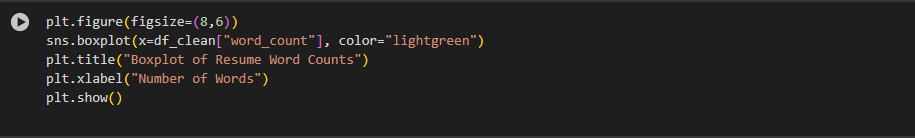


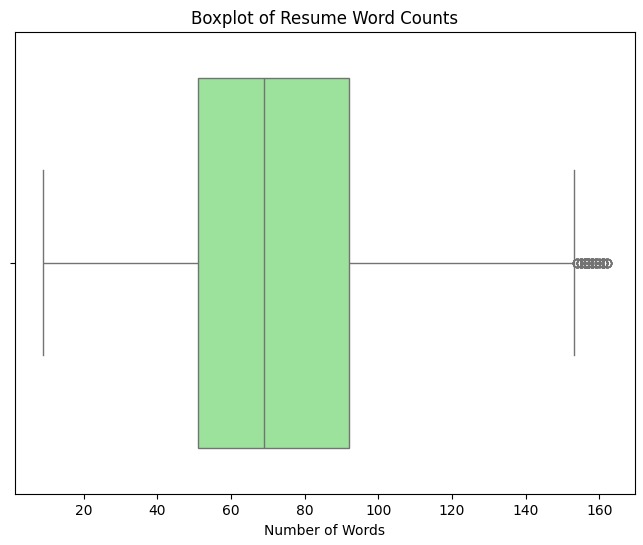
Word count



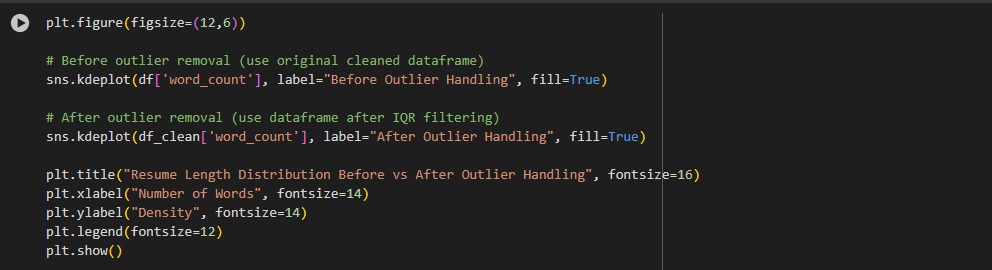


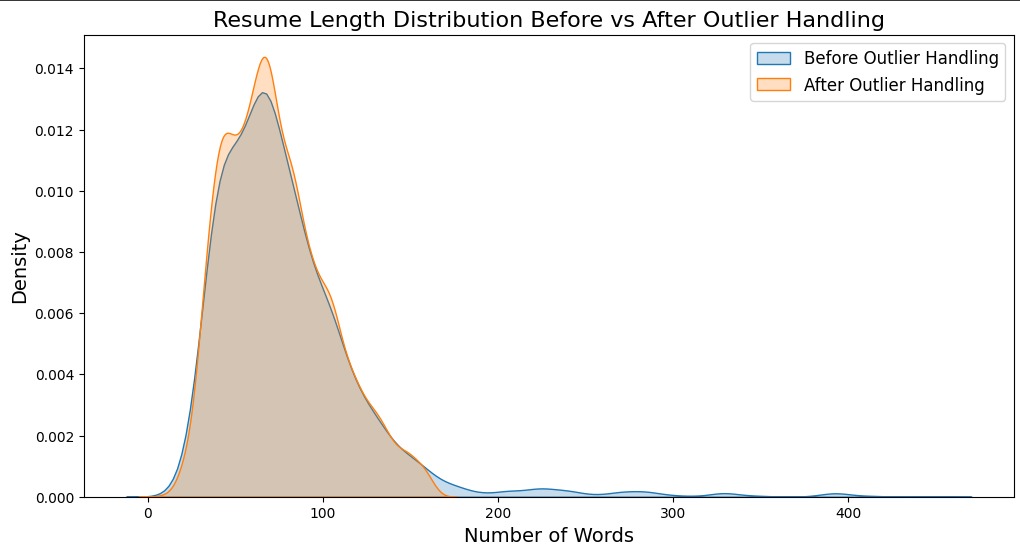
Boxplot



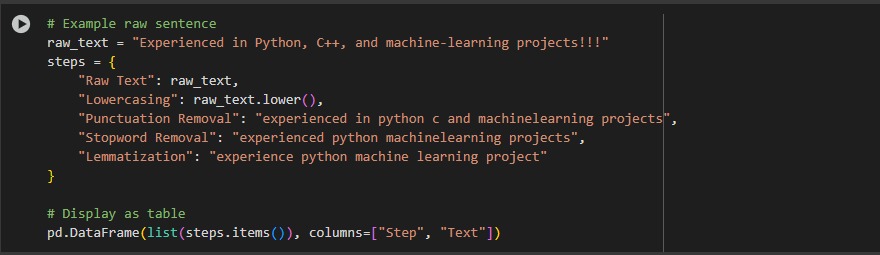


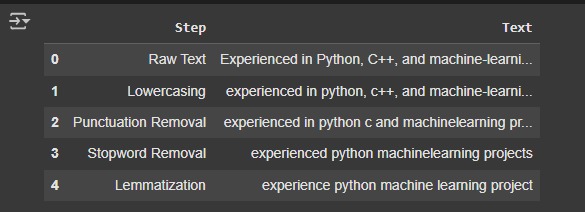
Outlier Handling





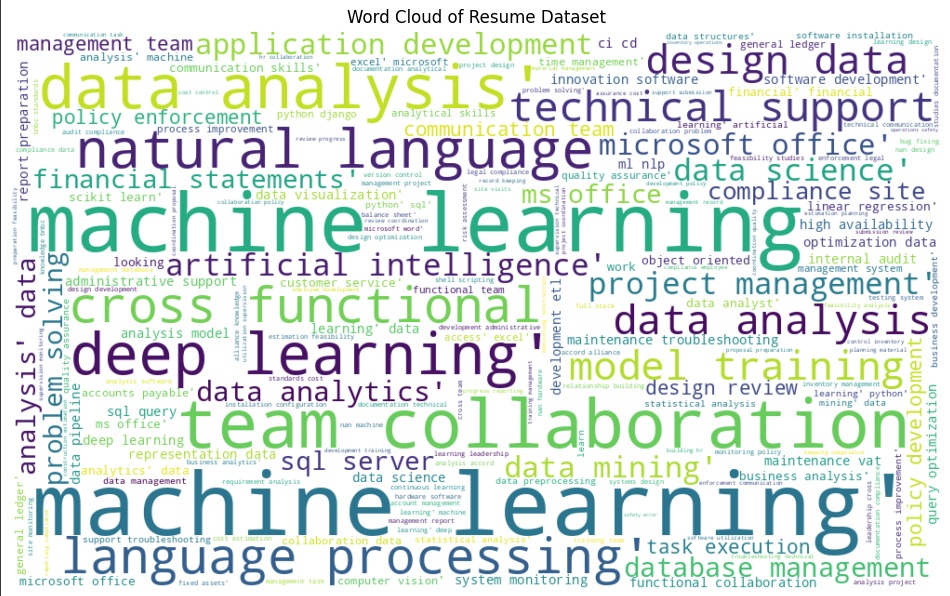
Text Cleaning

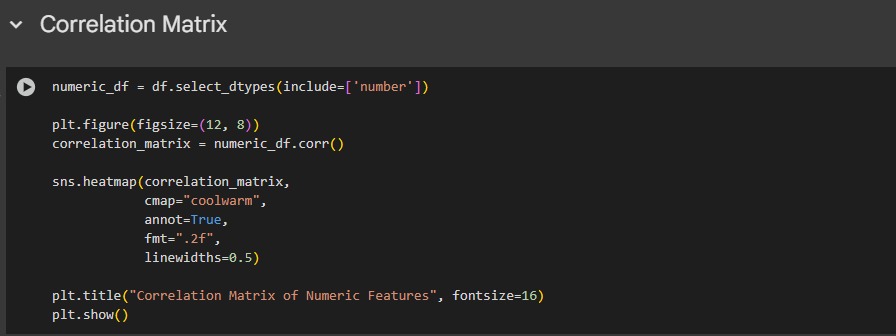




Word Cloud

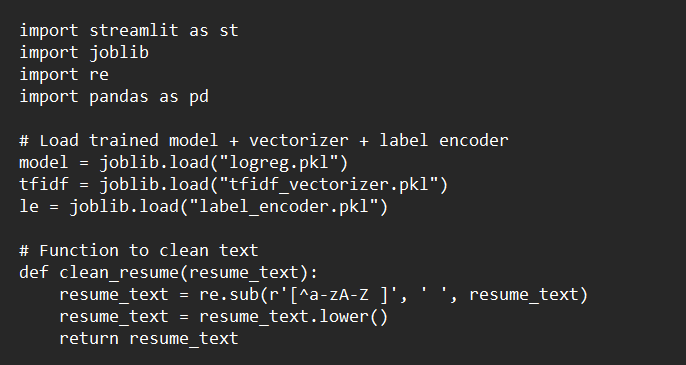


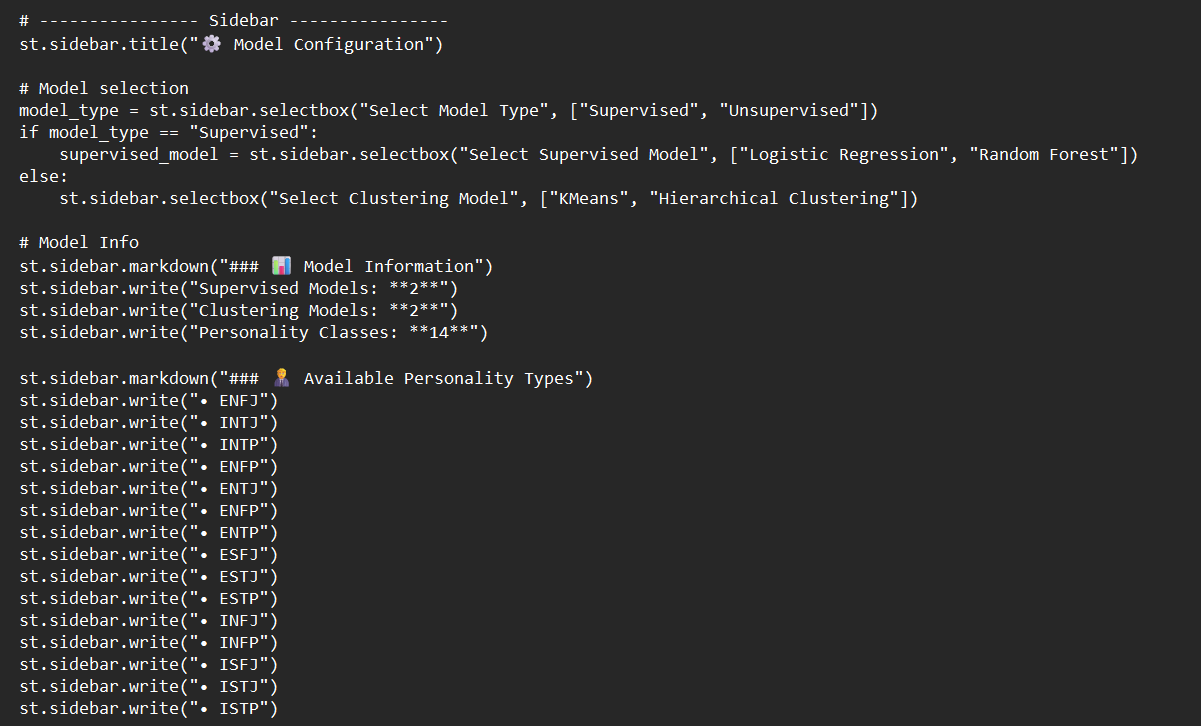


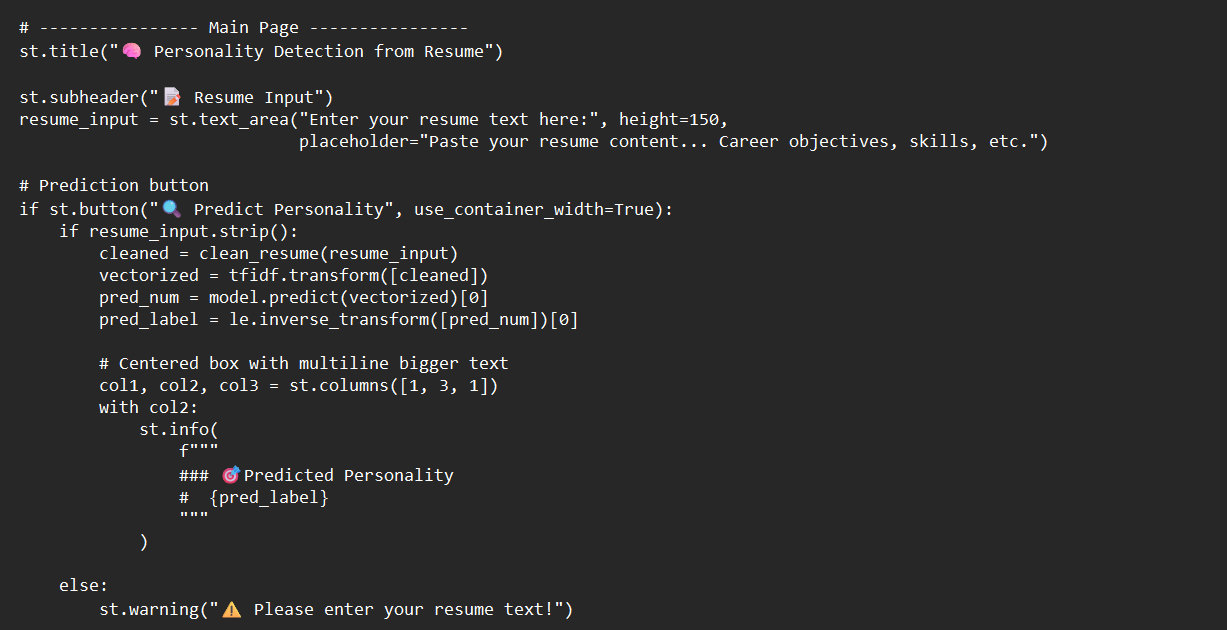


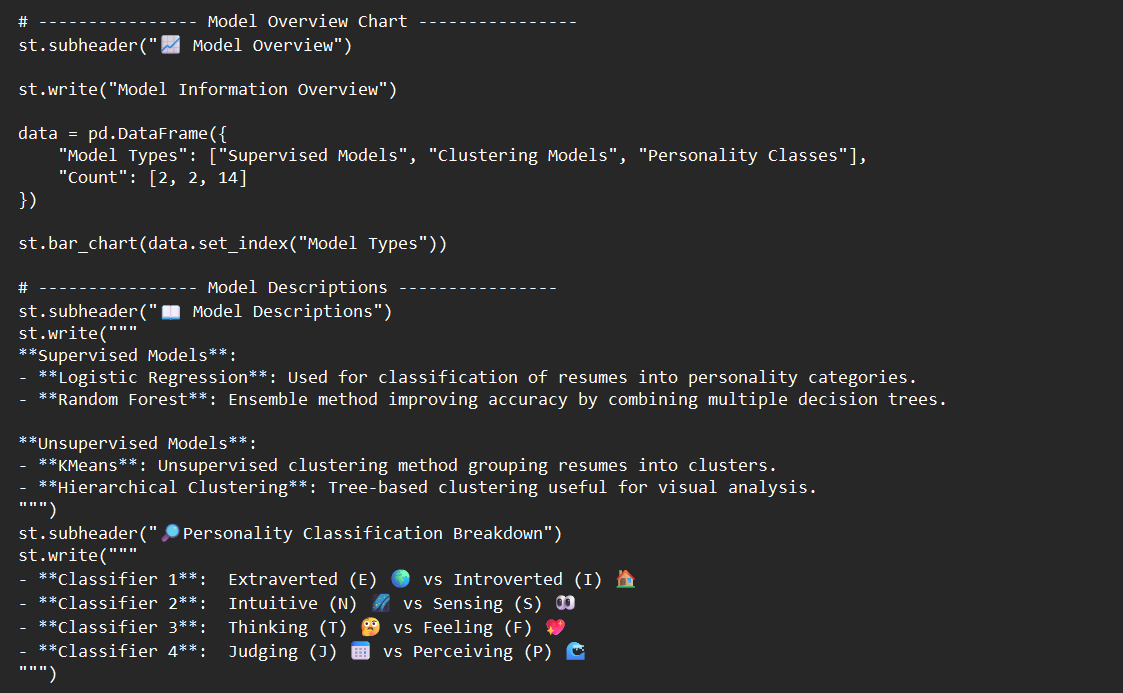


* **Streamlit Code**



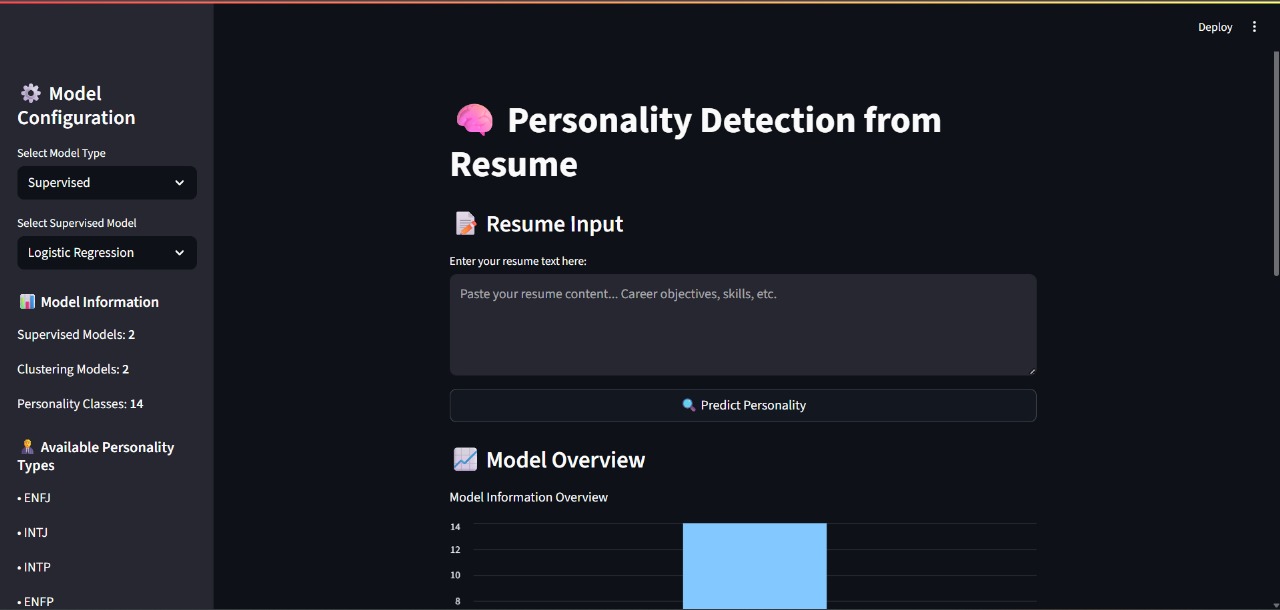


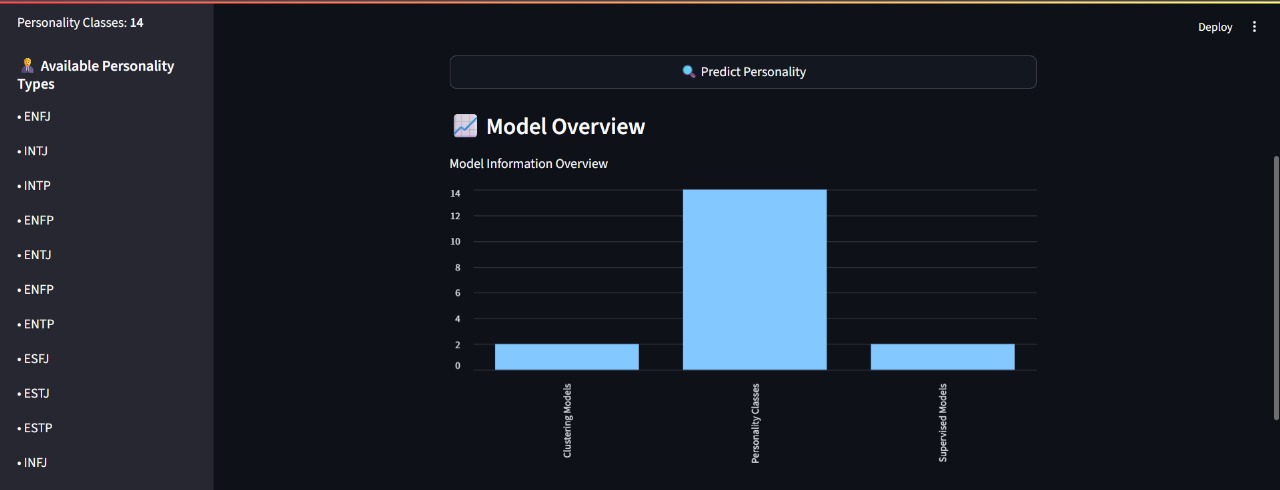


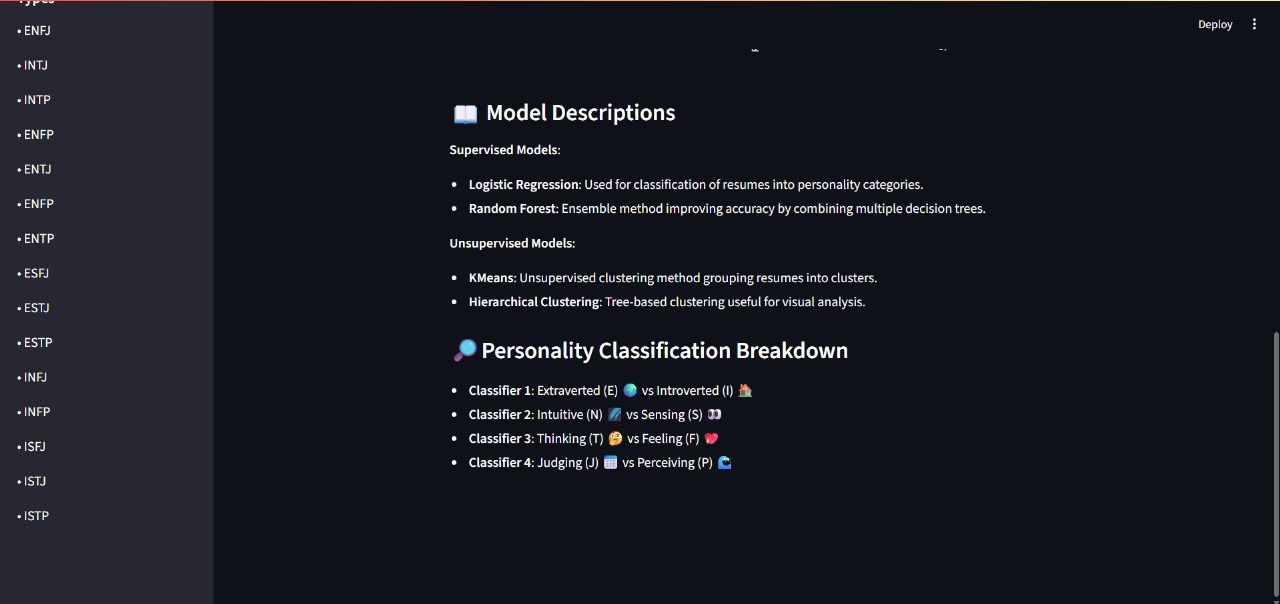


* **Prototype(output)**

Detecting personality based on Resumes to asign suitable job to the candidates acoording to their Pesrsonality by different model using streamlit.







**✅ Final Note on Codes**

The complete coding implementation demonstrates the step-by-step pipeline of the project, starting from data preprocessing and feature engineering to model training, evaluation, and deployment.

The integration of Streamlit further enabled the development of an interactive prototype, where users can upload resumes and receive personality predictions along with suggested job roles in real time.

Including both intermediate outputs (plots, confusion matrices, ROC curves) and the final prototype interface ensures that the project is not only theoretically sound but also practically applicable. This codebase can be reused, scaled, and enhanced with more data or advanced models in the future.

Future Scope of Improvement

While this project successfully met its objectives, several opportunities for improvement and expansion exist:

1. **Larger Dataset:**The dataset used was limited in size. Expanding the dataset with more resumes across industries and roles will improve the model’s robustness and reduce bias.
2. **Balanced MBTI Labels:**Ensuring balanced representation of all MBTI types will help avoid skewed predictions and provide fair results for all personality categories.
3. **Advanced Feature Representation:**  
   Although TF-IDF worked well, other statistical or vectorization methods (e.g., n-grams, part-of-speech patterns) could be explored to capture more contextual meaning without requiring deep NLP.
4. **Model Enhancement:**More advanced classifiers such as Gradient Boosting (XGBoost, LightGBM, CatBoost) could be tested for improved performance. Ensemble methods combining Logistic Regression and Random Forest may also provide better stability.
5. **Explainability:**Since personality prediction directly impacts candidate evaluation, incorporating explainability techniques (e.g., feature importance visualization, rule-based explanations) would enhance trust and transparency in the system.
6. **Scalability & Deployment:**The current Streamlit prototype demonstrates feasibility. In the future, the system can be deployed on cloud platforms (AWS, Azure, GCP) and scaled to handle thousands of resumes in real time.
7. **Integration with HR Systems:**  
   This solution can be integrated with existing Applicant Tracking Systems (ATS) to automatically screen resumes, predict personalities, and recommend job roles seamlessly.
8. **Bias Mitigation:**Care should be taken to ensure fairness in prediction. Future work could include fairness-aware ML techniques to prevent gender, cultural, or domain-related biases from influencing results.
9. **Cross-Domain Applications:**  
   The approach can be extended beyond resumes to other forms of career-related text data such as LinkedIn profiles, cover letters, or online portfolios.

10. **Career Development Support:**  
 Instead of stopping at personality prediction, the system could be expanded to recommend **skill-building programs, certifications, or training paths** based on both the candidate’s personality and skills.

Conclusion

This project, *“Personality Detection from Resume using MBTI for Job Role Recommendation”*, aimed to explore how raw resumes can be processed, transformed into structured data, and analyzed with machine learning techniques to predict the personality traits of candidates. The ultimate goal was to build a system that not only detects MBTI-based personality categories but also provides insights that can be useful for aligning candidates with suitable job roles.

The workflow began with **systematic data preprocessing**, which included cleaning missing values, normalizing text, and combining relevant fields from resumes. These steps were crucial because resumes are inherently unstructured and inconsistent. Once cleaned, resumes were represented numerically using **TF-IDF (Term Frequency–Inverse Document Frequency)**, which allowed us to capture the importance of specific words and phrases within the dataset.

To further refine the feature set and improve computational efficiency, the **Chi-Square test** was applied to select only the most relevant features correlated with MBTI personality labels.

With features prepared, multiple machine learning models were trained and tested. **Logistic Regression And Random Forest** were employed as supervised learning techniques, while **K-Means and Hierarchical Clustering** were experimented with for unsupervised grouping.

Among these, **Logistic Regression proved to be the most reliable classifier**, achieving a balanced performance with **78% test accuracy and an F1-score of 0.76**. While Random Forest achieved slightly higher test accuracy (80%), it showed overfitting with very high training accuracy (95%), reducing its generalizability. The clustering-based methods, on the other hand, failed to provide meaningful alignment with MBTI classes, highlighting the superiority of supervised approaches for this type of classification.

A major highlight of this work was the development of a **Streamlit-based prototype**, which provides an interactive interface where a user can upload a resume and instantly receive a predicted personality type along with a suitable job role suggestion.

This prototype demonstrated the **practical applicability** of the system in real-world recruitment scenarios. It bridged the gap between theoretical model building and actual end-user usability, showing how recruiters or HR managers can leverage the system in decision-making.

In conclusion, the project achieved the following:

* Built a **systematic data preprocessing pipeline** to clean and structure resume text.
* Represented resumes using **TF-IDF feature vectors** and refined features via **Chi-Square test**.
* Trained and compared multiple machine learning models, identifying **Logistic Regression** as the most balanced and interpretable choice.
* Validated results with **quantitative metrics** (accuracy, precision, recall, F1, ROC curves, confusion matrix).
* Designed a **prototype application** that demonstrates the system’s usability in a recruitment setting.

Thus, the project successfully showed that **basic text-based feature engineering combined with machine learning** can serve as a practical tool for personality prediction from resumes and can act as a supportive system for job-role recommendations.

✅ Final Remark

This project successfully showcased how **structured data preprocessing and machine learning** can transform unstructured resumes into actionable insights for personality detection.

While the system is not a replacement for psychometric testing or human judgment, it serves as a **supportive tool for recruiters** to make data-driven, consistent, and faster decisions.

With improvements in dataset size, model sophistication, and deployment, this system has the potential to evolve into a **comprehensive recruitment assistant**, capable of screening resumes, predicting personalities, recommending roles, and guiding career development paths.

In the long term, such solutions can make hiring processes more **efficient, objective, and candidate-friendly**, thereby bridging the gap between employers and job seekers.

Certificate

*This is to certify that Ms. Sharmistha Das, of Asansol Engineering College, Roll Number : 10800123186, has successfully completed a project on “PERSONALITY DETECTION FROM RESUME” using Machine Learning with Python under the guidance of Dr. Arnab Chakraborty.*

*----------------------------------------*

Dr. Arnab Chakraborty

**Asansol Engineering College**

Certificate

*This is to certify that Ms. Rishika Mishra, of Asansol Engineering College, Roll Number: 10800123164, has successfully completed a project on “PERSONALITY DETECTION FROM RESUME” using Machine Learning with Python under the guidance of Dr. Arnab Chakraborty.*

*----------------------------------------*

Dr. Arnab Chakraborty

**Asansol Engineering College**

Certificate

*This is to certify that Ms. Ragini Gupta, of Asansol Engineering College, Roll*

*Number: 10800123145, has successfully completed a project on “PERSONALITY DETECTION FROM RESUME” using Machine Learning with Python under the guidance of Dr. Arnab Chakraborty.*

*----------------------------------------*

Dr. Arnab Chakraborty

**Asansol Engineering College**

Certificate

*This is to certify that Ms. Shreya Banerjee, of Asansol Engineering College, Roll Number: 10800123192, has successfully completed a project on “PERSONALITY DETECTION FROM RESUME” using Machine Learning with Python under the guidance of Dr. Arnab Chakraborty.*

*----------------------------------------*

Dr. Arnab Chakraborty

**Asansol Engineering College**