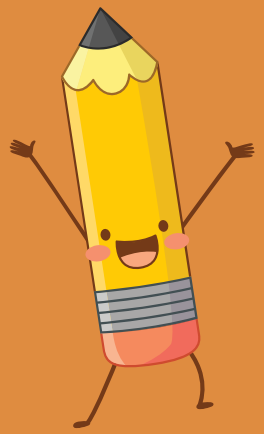
Team-96

Automatic Questions Tagging



Welcome!

- Fatma Ahmed Mohamed Eid Elnoby Level 3
- Mayan Mohammad Helmi Sayed Level 3
- Hagar Salem Odeh Hossein Level 3
- Hagar Adel Hamed Mohammed Level 3

Introduction

• The goal of this project is to perform text classification using various machine learning models.

• We will explore the process of preparing the dataset, preprocessing the text data.

• Extracting features using TF-IDF vectorization, and training/testing different models.

•The accuracy of each model will be evaluated, and the results will be visualized for better understanding.

Dataset Preparation:

- The datasets used in this project are Questions.csv, Answers.csv, and Tags.csv.
- The code reads the datasets and displays the first few rows of each dataset.
- The columns of Questions and Answers datasets are renamed for clarity.
- Unnecessary columns are dropped from the Answers dataset.
- The answers are grouped based on the 'Id' column and joined into a single string.
- The data type of the 'Tag' column is changed from object to string.
- Tags are grouped by 'Id' and joined into a single string.
- All datasets are merged into a single dataset based on the 'Id' column.
- Unnecessary columns are dropped, resulting in a new dataset.
- A new column 'TagCount' is created to count the occurrences of each tag.
- The 'TagCount' column is merged with the existing dataset.
- Null values are checked and dropped from the dataset.

Dataset Preprocessing:

• The WordNetLemmatizer and punctuation_remover functions are defined for text preprocessing.

- The functions are applied to the 'title', 'answer', and 'question' columns of the dataset.
- Texts are converted to lowercase and HTML tags are removed.
- The texts are split into words and lemmatized.
- Stopwords are removed from the texts.

Features Extraction (TF-IDF Vectorization):

- The TfidfVectorizer is used to convert text data into numerical features.
- The 'title', 'answer', and 'question' columns are vectorized separately.
- Label encoding is applied to the 'tag' column.

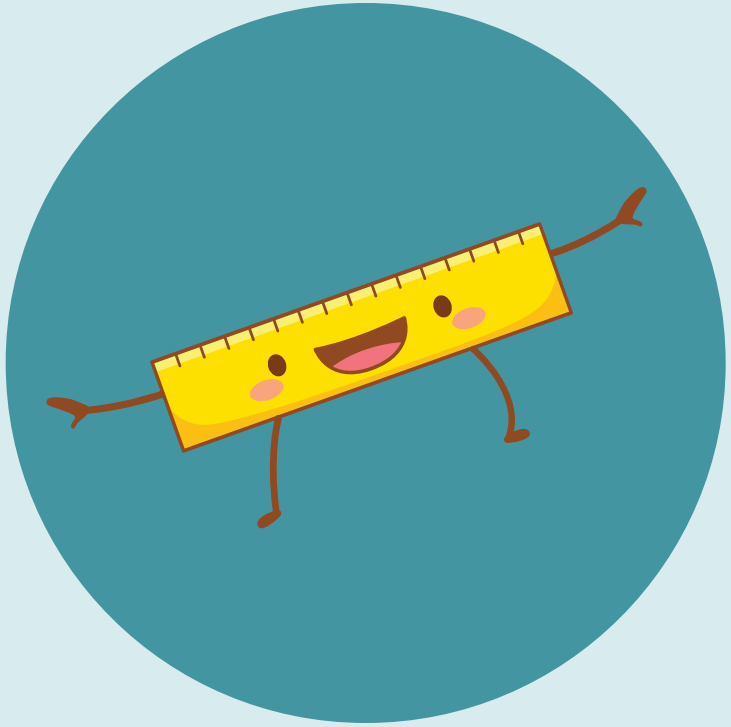
Model Training and Testing:

- The dataset is split into train and test sets.
- Various machine learning models, including KNN, SVM, Random Forest, Decision Tree, GBM, and Logistic Regression, are defined.
- Models are trained and tested using the train and test sets.
- The accuracies of each model are calculated and printed.

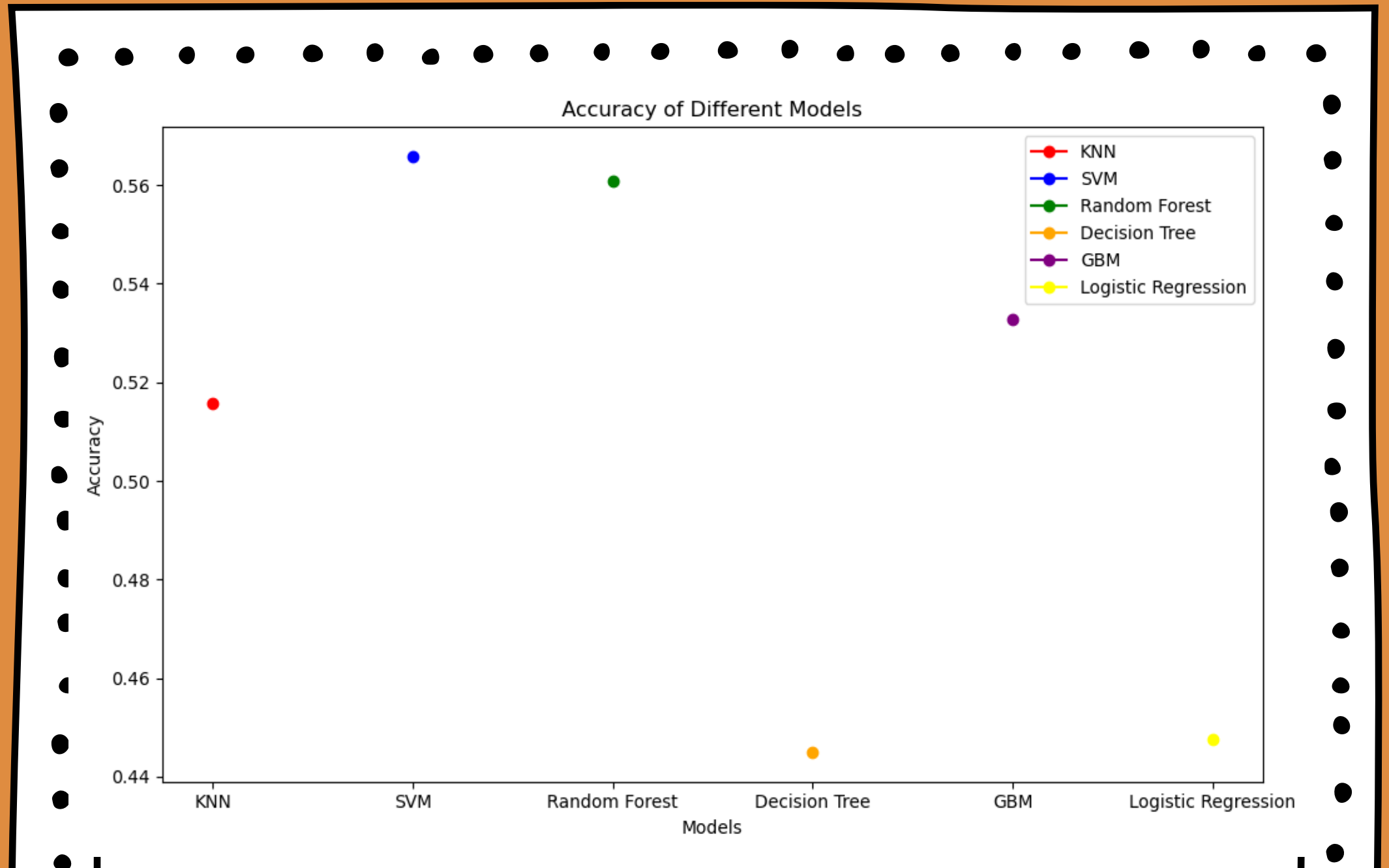
Results Visualization:

•The accuracies of SVM, Random Forest, Decision Tree, GBM, and Logistic Regression models are visualized using bar plots.

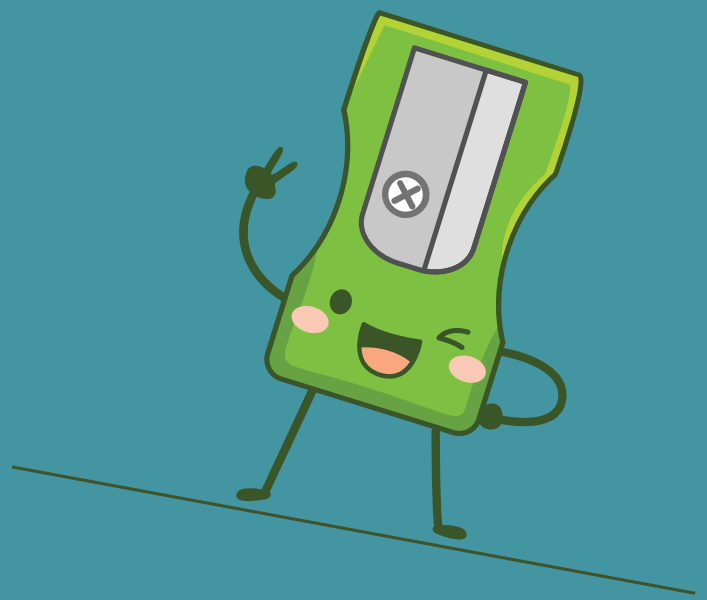
The accuracies of each model:



- Accuracy of KNN: 0.5157715260017051
- Accuracy of SVM: 0.5657254138266796
- Accuracy of Random Forest: 0.5609548167092924
- Accuracy of Decision Tree: 0.44501278772378516
- Accuracy of GBM: 0.5328218243819267
- Accuracy of Logistic Regression: 0.4475703324808184


Accuracy of Different Models
KNN
SVM
Random Forest
Decision Tree
GBM
Logistic Regression
0.56
0.54
0.52
0.50
0.48
0.46
0.44
Accuracy
KNN
SVM
Random Forest
Decision Tree
GBM
Logistic Regression
Models

Conclusion:



- SVM (Support Vector Machine) achieved the highest accuracy of 56.57%. It outperformed all other models in terms of accuracy, indicating its effectiveness in classifying the data.
- Random Forest and GBM (Gradient Boosting Machine) also achieved relatively high accuracies of 56.10% and 53.28% respectively. These ensemble learning models showed good performance in capturing complex relationships in the data.
- KNN (K-Nearest Neighbors) achieved an accuracy of 51.58%, which is relatively lower compared to other models. This indicates that the KNN model might not be the most suitable for this particular dataset.
- Logistic Regression and Decision Tree models achieved the lowest accuracies of 44.76% and 44.50% respectively. These models might not have captured the underlying patterns in the data as effectively as the other models.

Thank you