Machine Learning Hackathon

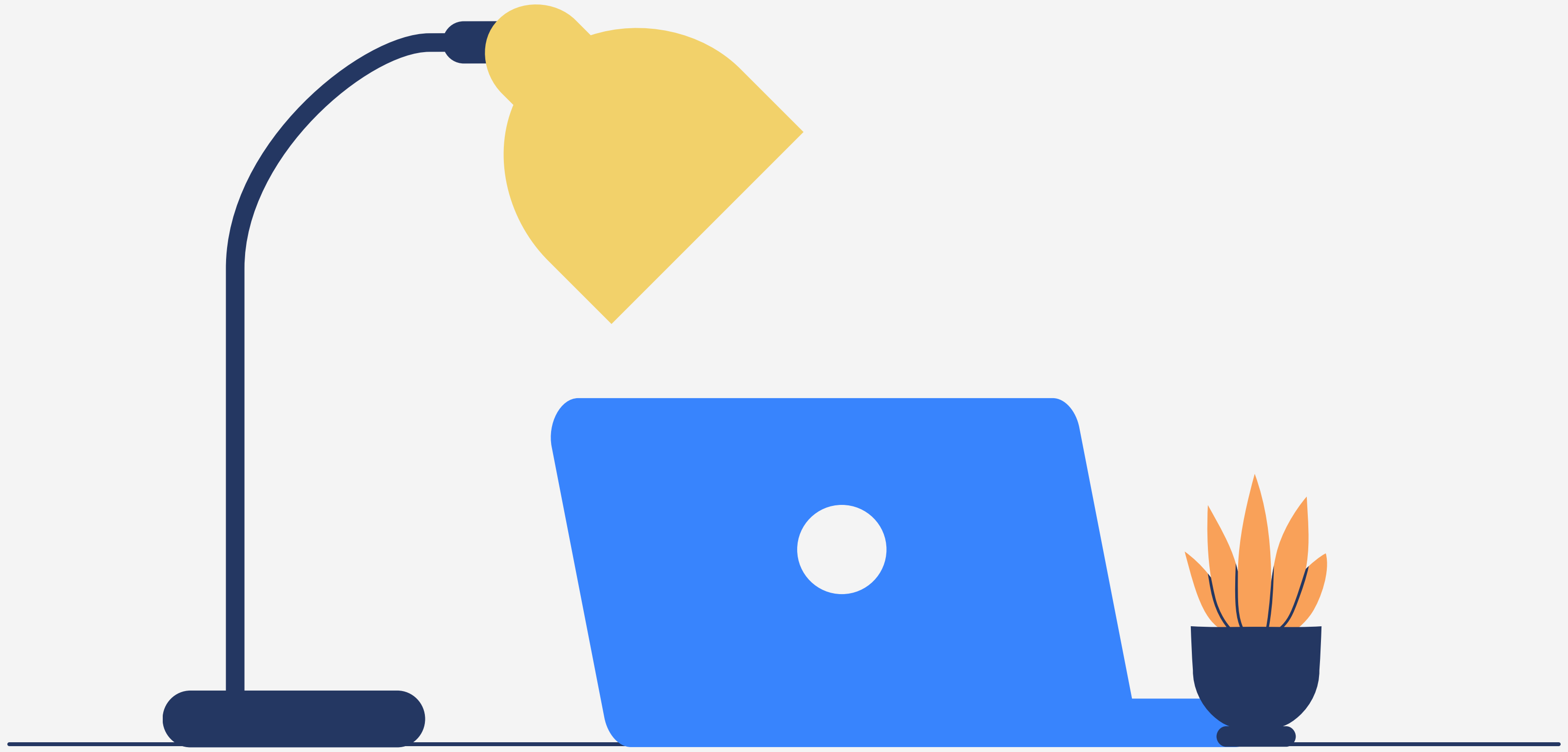
Team Name- Ved

Team Leader Name- Purvi Verma

Team Leader Email Address-
20ce01050@iitbbs.ac.in

Brief Description of the Problem at hand:

The best approach for solving the problem will depend on the specific dataset and the requirements of the application.



- The problem at hand is to classify text data into different taxonomies. The taxonomy is a hierarchical classification system that is used to categorize text data. The goal is to build a model that can accurately classify text data into the correct taxonomy.
- The problem is challenging because the text data can be very diverse. The text data can contain a variety of different words, phrases, and grammatical structures. This makes it difficult to build a model that can accurately classify all of the text data.
- However, the problem is also solvable. There are a number of machine learning algorithms that can be used to classify text data. These algorithms can learn the patterns in the text data and use these patterns to make predictions.

Solution proposed and description

The proposed solution is to use a systematic and data-driven approach to identify the columns that are most related to the "TAXONOMY" and "IS_HCP" column.

The CountVectorizer model is used to perform the text vectorization.
Encoding the target column: The target column ('TAXONOMY') is encoded using a LabelEncoder model. This ensures that the target column is in a format that the model can understand.

Model RandomForestClassifier is used for column "TAXONOMY"
Model Support Vector Machines is used for IS_HCP

Approach

01

The approach used in this project was to first identify the columns in the dataframe that are related to the "TAXONOMY" column. This was done by calculating the Cramer's V statistic for each column. The threshold for correlation was set to 0.5, which means that only columns with a correlation coefficient of 0.5 or greater were considered to be related to "TAXONOMY".

02

The code identified the following columns as being related: BIDREQUESTIP, USERPLATFORMUID, USERZIPCODE, and URL. However, I also manually detected that the column KEYWORDS should be included.

03

To determine which other variables to include, I used a list of column names to print out the number of unique values in each column. I found that the number of unique values in some of the columns was quite high. This means that the training model could encounter a different value for these columns that it has not seen before, which could cause the model to show errors. Therefore, the only remaining column that should be included is KEYWORDS, which will be encoded for the target.

Feature engineering

Feature engineering is the process of transforming raw data into features that are more suitable for machine learning algorithms. In this case, I performed the following feature engineering steps:

01 Text vectorization for text columns: The text vectorization step converts the text columns into a numerical representation that can be used by the model. The CountVectorizer model is used to perform the text vectorization.

02 Encoding the target column ('TAXONOMY'): The target column ('TAXONOMY') is encoded using a LabelEncoder model. This ensures that the target column is in a format that the model can understand.

03 Model Training and Evaluation: The code trains a Random Forest Classifier model (RandomForestClassifier()) and a Support Vector Classifier model (SVC()) on the preprocessed and encoded data. It then evaluates the models' performance by calculating the accuracy score on the test data. This step helps assess how well the models generalize to unseen data.

Tools

The following tools were used in this project:

NumPy

NumPy was used to perform mathematical operations.

Pandas

Pandas was used to manipulate the data.

Scikit-learn

Scikit-learn was used to train the model.

Why this approach works

1. Model Training and Evaluation: The code trains two different models, a Random Forest Classifier and a Support Vector Classifier, on the preprocessed and encoded data. These models are well-known and widely used in classification tasks. Random Forests are an ensemble learning method that combines multiple decision trees to make predictions, while Support Vector Machines (SVM) construct hyperplanes to separate different classes. Both models have shown good performance in various classification problems.
2. Evaluation Metrics: The code evaluates the models' performance using the accuracy score, which measures the proportion of correctly predicted labels. Accuracy is a commonly used metric for classification tasks when the classes are balanced. It provides an overall measure of how well the models are able to predict the correct class labels.

Accuracy Result

The accuracy results obtained from training and evaluating the models are as follows:

1. Random Forest Classifier:
   Accuracy on Test Data: 0.821748288572933
2. Support Vector Classifier (SVC):
   Accuracy on Test Data: 0.9787168685272951

Video link

https://drive.google.com/file/d/1CQNh92APqxoKYWqXKruH62omelLe1pWT/view?usp=sharing

Code link

https://colab.research.google.com/drive/1kpQInskQJN1wctRXAu0v3T07NYrf6Gqy?authuser=1#scrollTo=810xhCs-D0Q5

Result

https://docs.google.com/spreadsheets/d/1xdGBjHH90YfxycaL9U2ON31HhHEI2hp_h_fldhpjng8/edit#gid=684281203

Thank You