Table Classification from Financial Statements

DATE: 08-06-2024
D18 Batch

PRESENTED BY SANTHOSH R

Problem Statement

This project focuses on categorizing tables extracted from financial statements into specific types: Income Statements, Balance Sheets, Cash Flows, Notes, and Others.

DATASET DESCRIPTION:

The dataset comprises HTML files organized into different folders representing specific categories:

- Balance Sheets: Includes tables related to balance sheets.
- Cash Flow: Contains tables related to cash flow statements.
- Income Statement: Contains tables related to income statements.
- Notes: Contains miscellaneous tables related to financials.
- Others: Includes tables that do not fall into the above categories.

REQUIRED LIBRARIES:

The necessary modules and libraries are imported for data reading, preprocessing, visualizing, and for model evaluation respectively. pandas numpy, scikit-learn, imbalanced-learn matplotlib, nltk beautifulsoup4

DATA PREPROCESSING:

- Text Cleaning: This included removing non-alphabetic characters and extra spaces from the text, as well as converting it to lowercase.
- Stemming: The Porter Stemmer was used to apply stemming, reducing words to their root forms.

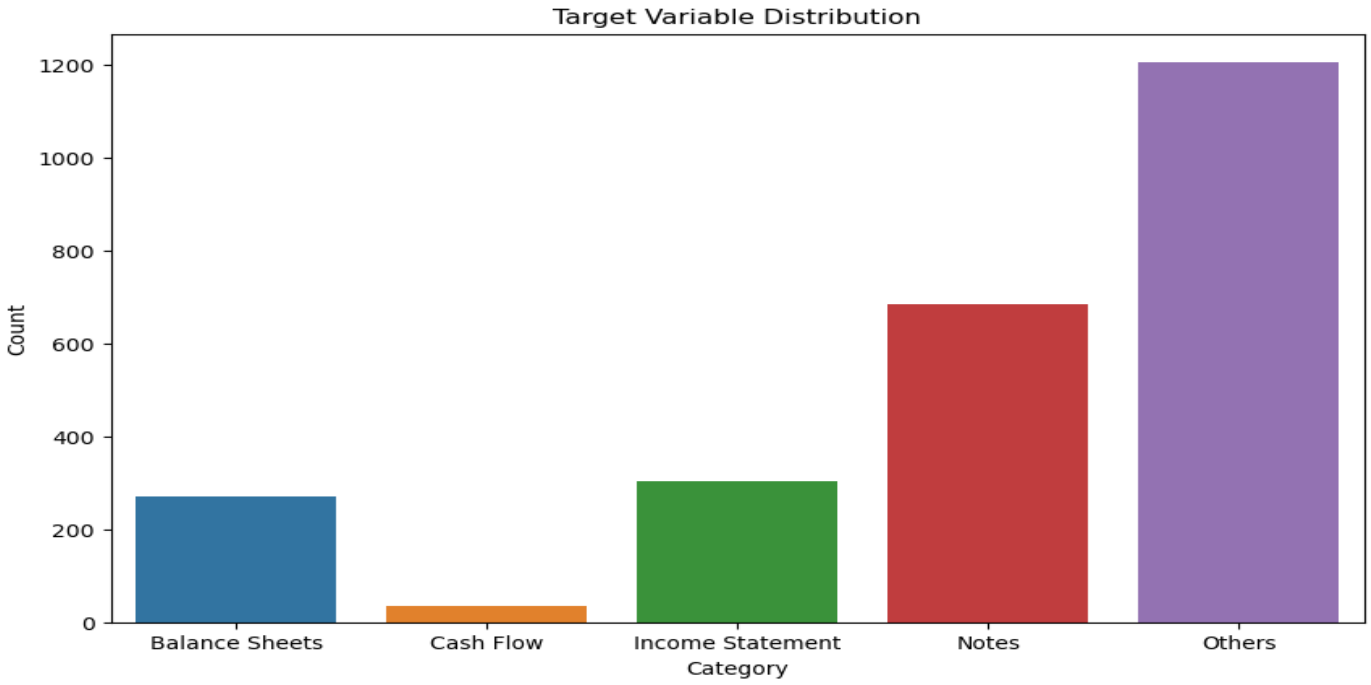
- Label Encoder: The target variable (categories) was encoded using a Label Encoder to convert them into numerical values.

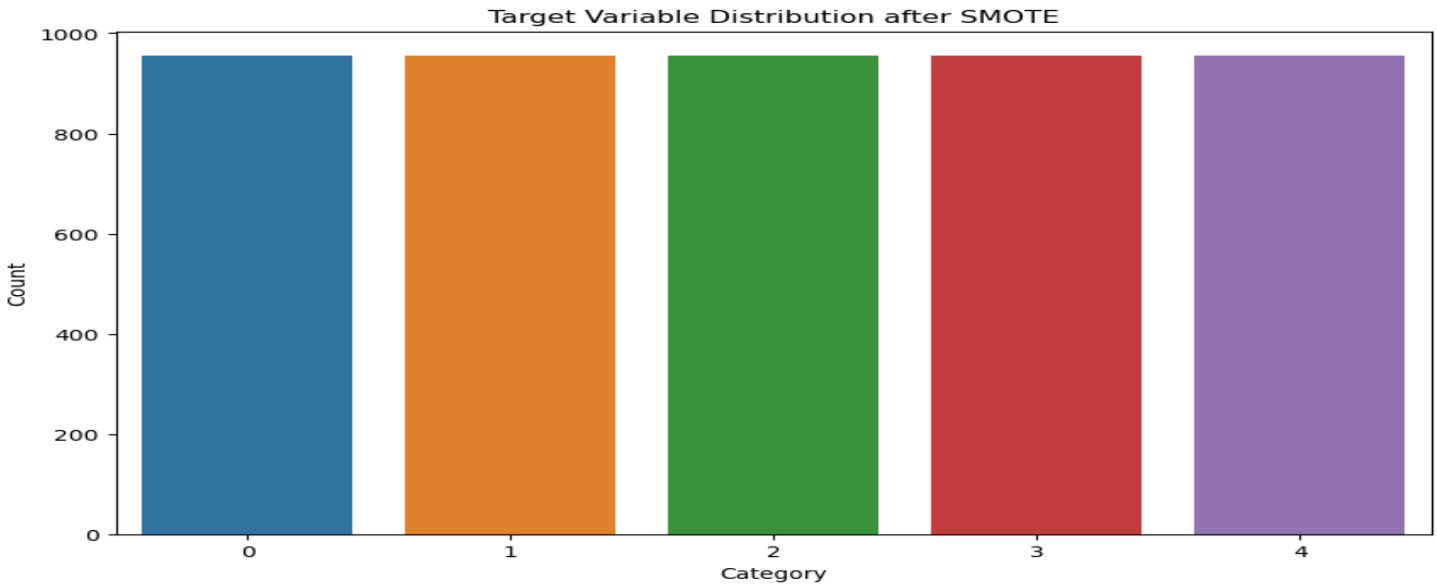
DATA EXTRACTION:

The files were grouped into directories, each corresponding to a distinct class. The extraction procedure included:

- Reading Files: Extracting text content from files within each directory.
- Labelling Data: Assigning category labels according to the directory names.
- Merging Data: Combining all documents into a unified DataFrame along with their respective labels.

IMBALANCED DATA: To tackle class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. This technique rebalances the dataset by creating synthetic samples for the minority classes, thus addressing the class imbalance issue.





DATA VECTORIZATION AND SPLITTING:

The data was divided into training and testing sets. Subsequently, the text data was transformed into numerical vectors using Count Vectorizer.

MODELS USED:

- Support Vector Machine (SVM): SVM is known for its effectiveness in high-dimensional spaces, making it a popular choice for text classification.

- Multinomial Naive Bayes: This classifier is well-suited for text classification tasks due to its simplicity and efficiency.

- Random Forest: An ensemble method that combines multiple decision trees for more accurate and stable predictions.

- KNN (K-Nearest Neighbors): This instance-based learning algorithm classifies data points based on their proximity to neighboring points.

- Decision Tree: Decision Trees recursively partition the feature space to make predictions.

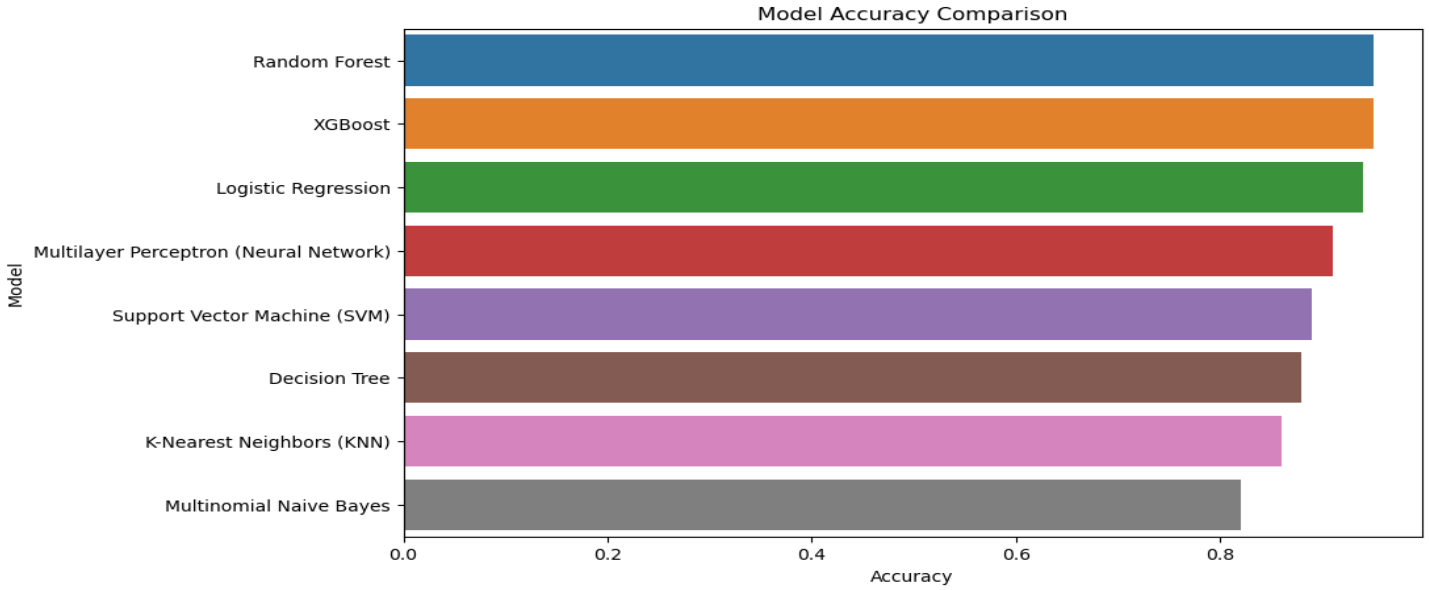
- Logistic Regression: A statistical model that uses a logistic function to model binary dependent variables.

- XGBoost: An optimized gradient-boosting algorithm known for its high efficiency and accuracy.

- Multi-Layer Perceptron (MLP): A type of neural network with input, hidden, and output layers, capable of modeling complex, non-linear relationships using backpropagation for training.

MODEL COMPARISON:

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
RANDOM FOREST	0.95	0.95	0.95	0.95
XGBOOST	0.95	0.95	0.95	0.95
LOGISTIC REGRESSION	0.94	0.94	0.94	0.94
MULTILAYER PERCEPTRON	0.91	0.92	0.91	0.91
SUPPORT VECTOR MACHINES	0.89	0.90	0.89	0.89
K-NEAREST NEIGHBORS	0.88	0.89	0.86	0.87
MULTINOMINAL NAÏVE BAYES	0.82	0.87	0.82	0.82
DECISION TREE	0.88	0.88	0.88	0.88



CONCLUSION:

According to the evaluation metrics, the Random Forest and XGBoost models attained the highest accuracy at 95%, with Logistic Regression closely following at 94%.