GST Analytical Hackathon 2024

Developing a Predictive Model in GST

Team Name	:CodeOholicks
Team Members	1. Muktha Ghosh 2. N.Varshitha 3. D.Manusha 4. G.srinitha 5. B. Anusri
Guidance	: Mr. M.Tulsi Ram
Contractual Delivery	: Online
Actual Delivery Date	: 12 October 2024
Status	: Submitted

ABSTRACT

The target for this project is to generate an accurate model of prediction, which has the capability to categorize a variable as required based on input data. The dataset possesses different features; some are missing, whereas others have outliers. When dealing with such difficulties, some of the methods used are data imputation, outlier removal, and scaling. Then, we apply oversampling with SMOTE, especially when class imbalances are present to balance the dataset. We design and apply several types of machine learning models such as Logistic Regression, Artificial Neural Networks (ANN) as well as Multi-layer Perceptron, and compute the performance of every model by using accuracy, F1-score, and classification report. The main objective of the experiment is to compare various approaches and select the one based on which model does better on the evaluation metric.

1.INTRODUCTION

1.1 Motivation

Predictive models are applied in almost all types of business, such as healthcare, finance, and marketing. During the big-data era, developing a model that predicts the outcome precisely is highly important. This project has been inspired by real-world problems where advanced machine learning algorithms must be applied for a given dataset focused on developing a reliable and generalized prediction model.

The problem in hand is to develop a machine learning model that would be very effective in predicting the target variable on unseen data. The dataset provided contains numerous features and the target values, but the data doesn't have a perfect distribution, mainly due to missing values and possible outliers.This, therefore requires steps of appropriate preprocessing, which include dealing with missing data and class imbalance problems as well as outlier detection. In addition, careful selection and tuning of suitable algorithms are also necessary to establish a solid predictive model that generalizes well to unseen data.

1.2 Problem Statement

The challenge here will be to develop a model for predicting the target variable on new, unseen data given the various features in this dataset along with their corresponding target values. The data are imbalanced, have missing values, and even potential outliers, so effective preprocessing of the data and selection of appropriate algorithms are fundamentally crucial.

1.3 Objective

The objective of the project is :

- To Preprocess the data by handling missing values, outliers, and imbalanced classes.
- To apply machine learning algorithms like logistic regression, neural network
- To evaluate the performance of the model using evaluation metrics like precision, recall, accuracy and f1 score.
- To select the best model for generalization to unseen data

2.BACKGROUND WORK

2.1 Existing System

Regularly, decision trees, linear regression, and support vector machines are applied to traditional machine learning models that have commonly been used for typical classification problems. However, these models have a significant number of disadvantages, most notably when it comes to dealing with imbalanced datasets, missing data, or complex relationships between features.

2.2 Disadvantages of the Existing System

- Imbalance in Data: Classical models tend to favor the majority class and do not predict the minority class well.
- Outliers: Models like linear regression are quite sensitive to outliers, thereby affecting predictions while lowering the true accuracy of a model.
- Missing Values: Incomplete data sometimes leads to a wrong inference or even to model failure.
- Nonlinearity is limited: Conventional models fail to represent many nonlinear relationships very well.

2.3 Proposed system

To overcome the limitations of the traditional models, this project suggests the following:

Data Preprocessing: Imputation for missing values and IQR-based techniques for removing outliers.

SMOTE: Synthetic oversampling of majority class in the minority class dataset to create opportunity for learning on the minority class data by the models.

These models will make use of more advanced models, like ANN and MLP, that are less constrained by the linear relationship of data. Dropout and L2 regularization are used to avoid overfitting.

3. ANALYSIS :

3.1 Software Requirements

3.2 Algorithms used:

1. **Linear regression:**

Linear Regression fits a linear equation to observed data to model the relationship between input features (which are independent variables) and a continuous target variable (dependent variable). It is a primary machine learning algorithm. Here, Linear Regression acts as a baseline model so it provides a simple, interpretable solution. However this can capture only linear relationships between the inputs and output - in many real-world applications this might not be enough and there are plenty of complexity and non-linearity in the relationship. Also, independence and normally distributed nature of features are required assumptions which rarely met by real data.

2. **Neural network:**

It is one of the sophisticated models of machine learning closely akin to the human brain in regard to finding patterns in data. An ANN consists of many layers of interconnected neurones: an input layer, one or more hidden layers, and an output layer. Every neurone receives inputs and then sends the result through an activation function, which decides the output. There are quite a number of fully connected layers in ANN design used for this project whereby every neurone in one layer is linked with every other neurons in the next layer. This will capture complex non-linear inter-correlations between characteristics and target variables. The neural network therefore learns several levels of abstraction that help make it much more plausible than the linear regression network.

4. DESIGN

4.1 Design approach

The design approach undertaken is a step-by-step solution for the problem at hand, predictive modeling:

- **Data Preprocessing:** The missing values are filled with mean imputation. The outliers are capped using the Interquartile Range method.
- **Feature Engineering:** SMOTE oversampling technique is adopted for the minority class. Numerical features are scaled using StandardScaler to make all models optimal.
- **Model Development:** Engage Logistic Regression as baseline, ANN and MLP on higher level of recognition of complex patterns.
- **Evaluation Metrics:** Accuracy and F1-score with an eye on how well it generalizes to the unseen test data.

4.2 Block Diagram

Blocks involved while designing workflow are:

Data Preprocessing: take care of missing values, identification and treatment of outliers, Scaling.

Feature Engineering: Application of SMOTE in balancing the classes.

Model Construction: Logistic Regression, ANN, and MLP model.

Model Evaluation: With the metrics of accuracy and F1-score.

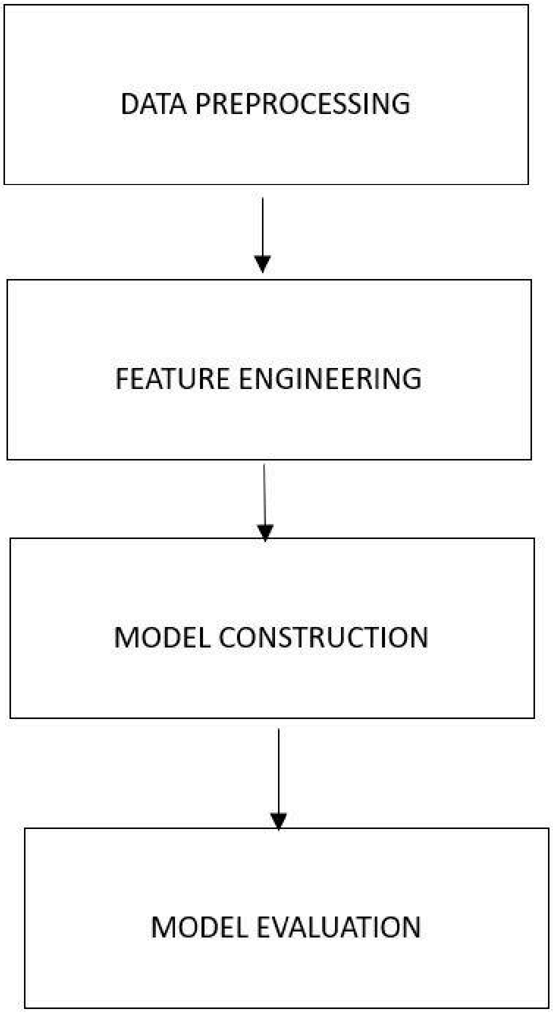


Fig- 4.2 Block Diagram

5. IMPLEMENTATION DETAILS

1. Data Preprocessing:

- Handling Missing Values:
    - Use fillna() to fill missing values with the mean of the respective columns.
- Outlier Detection and Treatment:
    - Use the Interquartile Range (IQR) method to identify and cap outliers.
- Scaling Features:
    - Apply StandardScaler() to normalize numerical features for better model performance.

2. Feature Engineering:

- SMOTE for Class Imbalance:
    - Use SMOTE() to oversample the minority class in the dataset and balance the class distribution.
- Scaling:
    - Use StandardScaler() after data preprocessing to scale features.

3. Model Building:

- Baseline Model (Logistic Regression):
    - Implement LogisticRegression() from scikit-learn for the baseline model.
- Artificial Neural Networks (ANN) / Multi-Layer Perceptron (MLP):
    - Use TensorFlow/Keras to build ANN and MLP models with hidden layers, activation functions, and regularization techniques like dropout and L2.
- Training Models:
    - Train models using fit() and apply early stopping to avoid overfitting.
    - For ANN/MLP, implement dropout and L2 regularization techniques.

4. Evaluation of Models:

- Metrics:
    - Evaluate model performance using accuracy, precision, recall, F1-score, and classification reports.
- Generalization:
    - Test on unseen data to assess how well the models generalize.

5. Hyperparameter Tuning (Future Work):

- Experiment with hyperparameter tuning for ANN and MLP models to optimize their performance.
- Explore other algorithms like Gradient Boosting Machines and ensemble learning techniques.

5.1 Key Functions

i. Data preprocessing functions:

1. fillna(): The missing values in the data are replaced by mean values of each column. It will also be used to make outliers and edge cases clipped.
2. StandardScaler(): It is being used in the case of feature scaling.

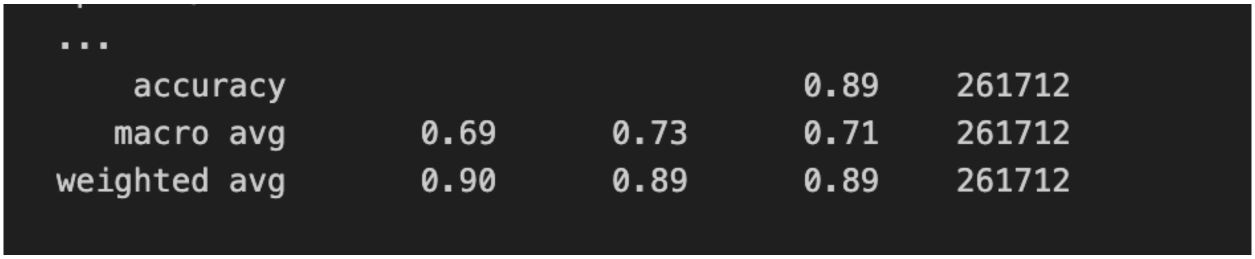
ii. SMOTE Implementation: A fit_resample() function can be used for the oversampling purpose of a minor class.

iii. Model Building Functions:

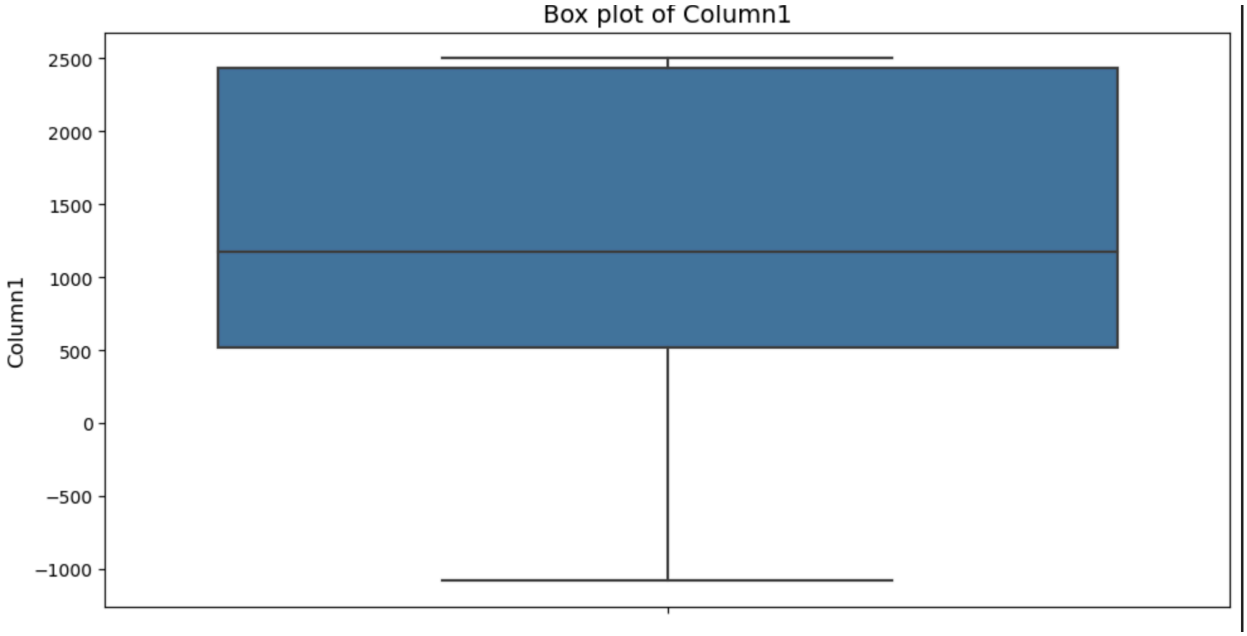
1. Logistic Regression: LogisticRegression() from the scikit-learn library is used.
2. ANN/MLP: Layers from tensorflow.keras could be used to develop deep learning models.

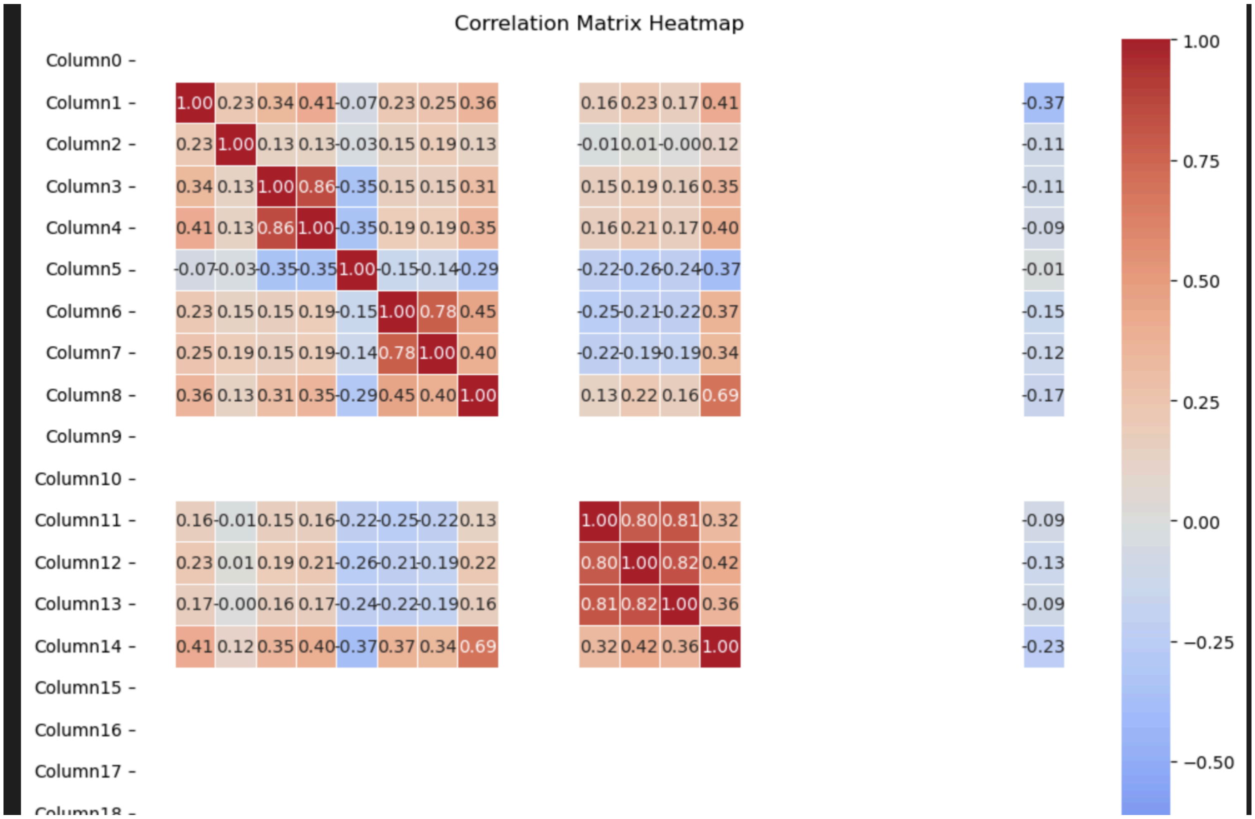
iv. Training Model: Models are trained using fit() function along with early stopping as well as lr schedulers if it is ANNs.

6. RESULT ANALYSIS



```
...
     accuracy                           0.89    261712
    macro avg       0.69      0.73      0.71    261712
 weighted avg       0.90      0.89      0.89    261712
```




Correlation Matrix Heatmap
Column0
Column1 1.00 0.23 0.34 0.41 -0.07 0.23 0.25 0.36 0.16 0.23 0.17 0.41 -0.37
Column2 0.23 1.00 0.13 0.13 -0.03 0.15 0.19 0.13 -0.01 0.01 -0.00 0.12 -0.11
Column3 0.34 0.13 1.00 0.86 -0.35 0.15 0.15 0.31 0.15 0.19 0.16 0.35 -0.11
Column4 0.41 0.13 0.86 1.00 -0.35 0.19 0.19 0.35 0.16 0.21 0.17 0.40 -0.09
Column5 -0.07 -0.03 -0.35 -0.35 1.00 -0.15 -0.14 -0.29 -0.22 -0.26 -0.24 -0.37 -0.01
Column6 0.23 0.15 0.15 0.19 -0.15 1.00 0.78 0.45 -0.25 -0.21 -0.22 0.37 -0.15
Column7 0.25 0.19 0.15 0.19 -0.14 0.78 1.00 0.40 -0.22 -0.19 -0.19 0.34 -0.12
Column8 0.36 0.13 0.31 0.35 -0.29 0.45 0.40 1.00 0.13 0.22 0.16 0.69 -0.17
Column9
Column10
Column11 0.16 -0.01 0.15 0.16 -0.22 -0.25 -0.22 0.13 1.00 0.80 0.81 0.32 -0.09
Column12 0.23 0.01 0.19 0.21 -0.26 -0.21 -0.19 0.22 0.80 1.00 0.82 0.42 -0.13
Column13 0.17 -0.00 0.16 0.17 -0.24 -0.22 -0.19 0.16 0.81 0.82 1.00 0.36 -0.09
Column14 0.41 0.12 0.35 0.40 -0.37 0.37 0.34 0.69 0.32 0.42 0.36 1.00 -0.23
Column15
Column16
Column17
Column18
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50

7.CONCLUSION AND FUTURE WORK

7.1 Conclusion

The Artificial Neural Network model emerged as the best predictor among all the models. They were able to catch nonlinear relationships; however, the Logistic Regression model was the least of the models since, although it had efficiency in computation, the patterns found in the data could not be followed up in a very effective manner. SMOTE must have been used for handling class imbalance that assists the model with prediction for the minority classes to its very best.

In conclusion, implementing predictive modeling solutions for the Goods and Services Tax (GST) system significantly enhances tax compliance and operational efficiency. By leveraging advanced techniques such as checksums and artificial intelligence, the project aims to improve data accuracy and detect fraud. These proactive solutions allow tax authorities to identify compliance issues early, strengthening the integrity of the GST system. Additionally, the insights generated support informed policy-making, fostering a transparent tax environment.

7.2 Future Work:

Further directions for this project are hyperparameter tuning of the ANN and MLP models, experimenting with other algorithms like Gradient Boosting Machines, or even applying ensemble learning techniques. Other techniques like feature selection or principal component analysis (PCA) can be further done to reduce the dimensionality of the data.