Build a Customer Churn Prediction Model using Ensemble Techniques

Business Objective

Ensemble methods aim to combine the predictions of several base estimators built with a given learning algorithm to improve generalizability/robustness over a single estimator. Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models. The most famous ensemble methods are boosting and bagging.

Bagging is a homogeneous weak learners' model that learns from each other independently in parallel and combines them for determining the model average. The Random Forest model uses Bagging.

Boosting is also a homogeneous weak learners' model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm. The AdaBoost and Gradient Boosting use Boosting techniques.

In our case study, we will be working on a churn dataset. Churned Customers are those who have decided to end their relationship with their existing company.
XYZ is a service-providing company that provides customers with a one-year subscription plan for their product. The company wants to know if the customers will renew the subscription for the coming year or not.

We have already seen how the logistics regression model works on this dataset in the first project of this series; Churn Analysis for Streaming App using Logistic Regression
We have also implemented the decision tree algorithm in our second project; Build a Customer Churn Prediction Model using Decision Trees
It is advised to check these two projects first before starting with ensemble techniques.

Data Description

This data provides information about a video streaming service company, where they want to predict if the customer will churn or not. The CSV consists of around 2000 rows and 16 columns

Aim

To build ensemble models like Random Forest, Adaboost, and Gradient boosting models on the given dataset to determine whether the customer will churn or not.

Tech stack

- Language - Python
- Libraries - NumPy, pandas, matplotlib, sklearn, pickle, imblearn, lime

Approach

1. Importing the required libraries and reading the dataset.
2. Feature Engineering
   - Dropping of unwanted columns
3. Model Building
   - Performing train test split
   - Random Forest Model
   - AdaBoost Model
   - Gradient Boosting Model
4. Model Validation (predictions)
   - Recall score
   - Precision score
   - F1-score
   - ROC and AUC
5. Feature Importance
   - Create a function to find important features
   - Plot the features
6. LIME implementation
   - Define a function for implementing the LIME technique over the dataset.

Modular code overview

```
input
  |_data_regression.csv

src
  |_Engine.py
  |_ML_Pipeline
            |_evaluate_metrics.py
            |_lime.py
            |_ml_model.py
            |_utils.py

lib
  |_[DEMO]_Ensemble_Learning.ipynb

output
  |_LIME_reports folder
  |_models folder
  |_ROC_curves folder
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input
2. src
3. output
4. lib

1. Input folder **-** It contains all the data that we have for analysis. There is one csv file in our case,
    - Data_regression

2. Src folder - This is the most important folder of the project. This folder contains all the modularized code for all the above steps in a modularized manner. This folder consists of:
    - Engine.py
    - ML_Pipeline
      The ML_pipeline is a folder that contains all the functions put into different python files, which are appropriately named. These python functions are then called inside the engine.py file.

3. Output folder **–** The output folder contains three subfolders.
    - LIME_reports - contains the LIME reports generated for all three algorithms.
    - Models - contains the models generated for all three algorithms.
    - ROC_curves - contains the ROC curves generated for all three algorithms.

4. Lib folder **-** This is a reference folder. It contains the original ipython notebook that we saw in the videos.

Project Takeaways

1. Introduction to ensemble techniques
2. Understanding the working of Random Forest, AdaBoost, and Gradient boosting algorithms.
3. Using python libraries such as matplotlib for data interpretation and advanced visualizations.
4. Data inspection and cleaning
5. Using sklearn library to build the Random Forest, AdaBoost, and Gradient boosting models.
6. Splitting Dataset into Train and Test using sklearn.
7. Making predictions using the trained model.
8. Gaining confidence in the model using metrics such as ROC, AUC, recall, precision, and f1 score
9. Handling the unbalanced data using SMOTE method.
10. Performing feature importance.
11. Evaluating the ROC curve results across multiple models
12. Evaluating the different models w.r.t the feature importance results generated.
13. Understanding the concept of LIME in machine learning
14. Implementing the LIME technique on the dataset