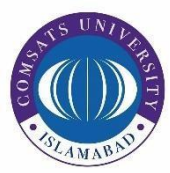

PROGRAMMING FOR ARTIFICIAL INTELLIGENCE

ML PROJECT

Customer Churn Predictor

SUBMITTED TO: Sir Aksam Iftikhar

SUBMITTED BY: Attia Inayat (SP23-BAI-012)

SUBMITTED DATE: 12 Dec,2024.

Customer Churn Predictor Report

Problem and Dataset Description

Customer Churn refers to a phenomena where the customers stop using the company services. This causes the loss to the company. So, the company wants to know that which customers are about to churn so they can give them discounts or offers so they do not churn. This Customer Churn Dataset consist of the features of customers .like(id, gender, senior citizen, monthly charges,total charges) and the target column either the customer has churn or not. So we will make a model where it will make predictions that the customer is about to churn or not and what are the key factors due to which they are leaving the platform.

Methodology

Preprocessing Steps

1. Dropped the duplicated and Null value rows.

2. Label Encoding: Applied Label Encoding on the target column.

3. Standardization and Normalization: Continuous features were scaled using StandardScaler to ensure uniformity and MinMaxScaler was applied to put them in a specific range..

4. Handling Imbalanced Data: SMOTEENN was applied to balance the dataset.

5. PCA was applied to reduce the high dimensionality.

Algorithms Applied

1. SVC: It finds the hyperplane to classify the target column by transforming the data into high dimensionality.

2. Random Forest: An ensemble learning method for improved accuracy, it works by training many weak decision trees and then make a final prediction by taking the majority class vote

3. XGBoost: A gradient boosting algorithm known for its high performance.In this trees work in sequential manner by taking the input of previous tree and reducing the error.

Optimization Techniques

- GridSearchCV: Exhaustive search over specified hyperparameter values, and it runs on all the possible combinations of parameters to find the best among of them.

- RandomizedSearchCV: Random search over hyperparameter space for faster results.

It works better in large dataset.

---

3. Results

Metrics:

- Evaluation metrics include Accuracy, Precision, Recall, F1-Score, and AUC-ROC.

- Performance comparison across algorithms was visualized using bar charts.

Visualizations

- Bar Chart: A comparative bar chart showcasing Accuracy, F1-Score, Precision, and Recall for each algorithm.

- Confusion Matrices: Heatmaps for SVC, Random Forest, and XGBoost to visualize classification performance.

---

4. Analysis

Insights

1. XGBoost delivered the best overall performance with the highest accuracy,precision, recall, F1-Score and AUC-ROC making it the most suitable algorithm for this dataset.

2. Random Forest showed normal results but required fine-tuning for optimal performance.

3. SVC performed well but its F1 score needed to be improved.

Challenges Faced

1. Imbalanced Data: SMOTEENN successfully addressed the imbalance, but further exploration of other techniques like cost-sensitive learning could be beneficial.

2. Hyperparameter Tuning: GridSearchCV was computationally intensive; RandomizedSearchCV provided faster results with comparable performance.

3. Feature Importance: Identifying the most impactful features was essential to understanding model behavior and optimizing performance.

Conclusion

The study demonstrated that XGBoost is the most effective algorithm for predicting customer churn dataset. Future work could explore neural networks or advanced feature engineering techniques to further enhance performance.

Algorithm	Accuracy Score	Precision Score	recall Score	F1 Score	roc_auc Score	Best Hyperparameter	Execution Time	Remarks
XGBoost	0.951560	0.943114	0.967742	0.955269	0.950361	{'gamma': 0.2, 'learning_rate': 0.05, 'max_dep...	23s	Performed best with pca and smoteenn
Random Forest	0.823110	0.636704	0.529595	0.578231	0.719932	{'bootstrap': True, 'criterion': 'entropy', 'm...	1s	Didn't capture the model pattern well
SVC	0.721113	0.440678	0.809969	0.570801	0.752348	{'C': 10, 'class_weight': 'balanced', 'gamma':...	14s	Just performed normal