Report on Telco Customer Churn Prediction

1. Introduction

This project aims to predict customer churn in a telecommunications company using various machine learning models. The dataset includes customer demographics, account information, and service usage details. The goal is to identify factors influencing churn and build predictive models to classify customers as likely to churn or not.

Customer churn refers to the phenomenon where customers stop using a company's service. Predicting churn is important for companies because retaining customers is usually less expensive than acquiring new ones. The dataset we used includes information about customers such as demographics, account details, and service usage.

Machine Learning Models

Logistic Regression: A statistical model that predicts the probability of a binary outcome (e.g., churn or no churn) based on one or more predictor variables. It uses a logistic function to model the probability.

Support Vector Machine (SVM): A model that finds the best boundary (or hyperplane) that separates the classes (e.g., churn vs. no churn) in the feature space. It is particularly effective in high-dimensional spaces.

K-Nearest Neighbors (KNN): A model that classifies a data point based on how its neighbors are classified. It looks at the 'k' closest data points and assigns the most common class among them to the new data point.

Decision Tree: A model that splits the data into branches to make decisions based on the values of the features. Each branch represents a possible decision, leading to a final classification.

Naive Bayes: A probabilistic model based on Bayes' theorem. It assumes that the features are independent given the class, making it simple and fast for classification tasks.

Random Forest: An ensemble model that uses multiple decision trees to make a prediction. It aggregates the results of individual trees to improve accuracy and control over-fitting.

Evaluation Metrics

Accuracy Score: The ratio of correctly predicted instances to the total instances. It measures the overall correctness of the model.

Confusion Matrix: A table used to evaluate the performance of a classification model. It shows the number of true positives, true negatives, false positives, and false negatives.

True Positives (TP): The number of instances correctly predicted as positive.

True Negatives (TN): The number of instances correctly predicted as negative.

False Positives (FP): The number of instances incorrectly predicted as positive.

False Negatives (FN): The number of instances incorrectly predicted as negative.

Precision: The ratio of true positive predictions to the total predicted positives. It indicates how many of the predicted positive cases are actually positive.

Precision = True positives/ (True positives + False positives)

Recall (Sensitivity): The ratio of true positive predictions to the total actual positives. It measures the ability of the model to identify all positive instances.

Recall = True Positive (TP) / True Positive (TP) + False Negative (FN)

F1 Score: The harmonic mean of precision and recall. It provides a single metric that balances both the precision and recall.

2*(Precision*Recall)/(Precision+Recall)

2. Data Preparation and Exploratory Data Analysis (EDA)

Data Loading and Initial Exploration

The dataset consists of 7043 entries and 21 columns, including customer demographics, account details, and usage metrics. Initial exploration revealed that there were no missing values, but the TotalCharges column required conversion to a numerical type as it was read as an object.

Dropping Irrelevant Columns

The customerID column was dropped as it did not contribute to the predictive modeling process.

Encoding Categorical Features

Categorical features such as gender, partner status, and various service usage indicators were encoded using label encoding to convert them into numerical values suitable for machine learning algorithms.

Data Visualization

Histograms were plotted for each feature to understand their distributions. This helped in identifying any skewness or anomalies in the data.

3. Feature Engineering

Standard scaling was applied to the dataset to normalize the features, ensuring that all features contributed equally to the model training process. No additional feature engineering steps were performed.

4. Model Building and Evaluation

Train-Test Split

The dataset was split into training and testing sets, with 70% of the data used for training and 30% for testing. This split ensured that the model's performance could be evaluated on unseen data.

Models and Evaluation Metrics

Multiple models were trained and evaluated using accuracy, confusion matrix, precision, recall, and F1 score.

Perceptron: Achieved a training accuracy of 71.60% and testing accuracy of 70.94%.

Logistic Regression: Performed well with a training accuracy of 80.43% and testing accuracy of 80.79%.

Support Vector Machine (SVM): Showed strong performance with a training accuracy of 81.97% and testing accuracy of 79.79%.

K-Nearest Neighbors (KNN): Achieved a training accuracy of 82.68% and testing accuracy of 76.48%.

Decision Tree: With a maximum depth of 5, it attained a training accuracy of 79.74% and testing accuracy of 78.56%.

Naive Bayes: Demonstrated a training accuracy of 75.38% and testing accuracy of 75.58%.

Random Forest: After hyperparameter tuning, it achieved a training accuracy of 81.32% and testing accuracy of 79.70%.

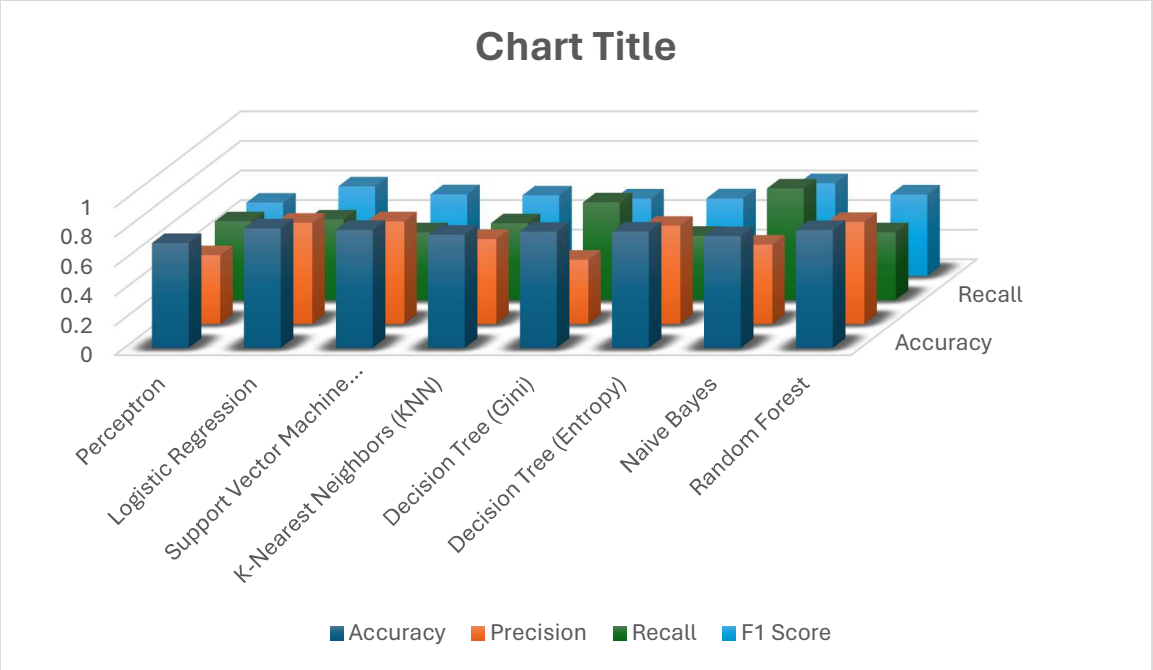
5. Challenges Faced

Model Selection: Balancing model complexity and performance was challenging, especially with more complex models that were prone to overfitting.

6. Table

Model	Accuracy	Precision	Recall	F1 Score
Perceptron	0.709	0.465	0.532	0.496
Logistic Regression	0.808	0684	0.544	0.606
SVM	0.798	0.694	0.458	0.552
KNN	0.765	0.574	0.513	0.545
Decision Tree(Gini)	0.786	0.434	0.660	0.524
Decision Tree(Entropy)	0.787	0.664	0.434	0.524
Naïve Bayes	0.756	0.536	0.754	0.627
Random Forest	0.797	0.690	0.458	0.551

7. Graph



Conclusion

The logistic regression model demonstrated superior performance, effectively balancing accuracy, precision, recall, and F1 score. This makes it the most suitable model for predicting customer churn in our dataset. The key metrics for logistic regression were an accuracy of 80.8%, precision of 68.4%, recall of 54.4%, and an F1 score of 60.6%. These results indicate that logistic regression not only correctly identifies the majority of churn cases but also maintains a reasonable balance between false positives and false negatives.

Other models like the Support Vector Machine (SVM) and Random Forest also showed promising results, with accuracies of 79.8% and 79.7% respectively. While SVM had a relatively high

precision (69.4%), its recall was lower (45.8%), indicating it missed a significant number of actual churn cases. The Random Forest model, despite undergoing hyperparameter tuning, achieved a precision of 69.0% and recall of 45.8%, suggesting potential improvements in recall could further enhance its utility.