## **Customer Churn Analysis Using Machine Learning on Telecom Data**

**Introduction:**

Customer churn is a critical issue for businesses, especially in the telecommunications industry, where customer acquisition costs are high, and retaining customers is more profitable. This project focuses on predicting whether a customer will churn or not using machine learning techniques. The telecom dataset contains information on various customer attributes, such as their tenure, services subscribed to, and payment history.

**Objective:**

The objective of this project is to build an efficient machine learning model to predict customer churn, enabling telecom companies to identify at-risk customers and take necessary steps to retain them.

**Technologies and Libraries Used**

The following tools and libraries were used to build and evaluate the machine learning models:

- Programming Language: Python

- Libraries:

- Pandas: For data manipulation and analysis.

- NumPy: For handling arrays and numerical computations.

- Matplotlib and Seaborn: For data visualization.

- Scikit-learn: For implementing machine learning algorithms.

- Google Colab: For interactive coding and analysis.

**Dataset Description:**

The dataset used for this analysis contains the following attributes:

- Customer Information: Gender, age, tenure with the company, contract type, and more.

- Service Information: Whether the customer has internet service, phone service, streaming services, etc.

- Billing Information: Monthly charges, total charges, payment method, etc.

- Target Variable: A column indicating whether the customer has churned (`Yes`/`No`).

**Data Preprocessing:**

Before applying machine learning models, it was essential to preprocess the data. The preprocessing steps included:

1. Handling Missing Values: Missing values were imputed based on the median for numerical columns and mode for categorical columns.

2. Encoding Categorical Variables: Categorical variables such as gender, payment method, and internet service were encoded using one-hot encoding.

3. Feature Scaling: Feature scaling was applied using StandardScaler to normalize continuous features like monthly charges and tenure

**Exploratory Data Analysis (EDA):**

Key Insights

1. Churn Distribution: About 26% of the customers in the dataset had churned, highlighting the importance of churn prediction.

2. Correlation Analysis: A heatmap of correlations between features showed that total charges, tenure, and contract type were highly correlated with churn.

3. Visualization: Pair plots and box plots were used to understand the distribution of key variables and their relationship with churn.

**Machine Learning Models:**

Multiple machine learning models were trained and evaluated to predict customer churn:

**1. K-Nearest Neighbors (KNN)**

- Training: The KNN model was trained using different values of `k` to find the best model.

- Performance:

- Training Accuracy: 79.88%

- Test Accuracy: 79.56%

knn.score(X\_train, Y\_train) Training accuracy

accuracy\_score(Y\_test, pred\_knn) Test accuracy

**2. Logistic Regression**

- Training: Logistic regression was trained to predict churn based on customer data.

- Performance: Accuracy scores were calculated to measure its performance.

3. Decision Tree

- Training: A decision tree classifier was implemented and hyperparameters like tree depth were tuned to avoid overfitting.

- Evaluation: Accuracy and confusion matrices were used to measure performance.

**4. Support Vector Machine (SVM)**

- Training: SVM with different kernels was evaluated to find the optimal hyperparameters using grid search.

- Performance: The model performed well with a tuned kernel.

**5. Random Forest Classifier**

- Training: Random forest classifier, an ensemble model, was used to improve prediction accuracy by combining predictions from multiple decision trees.

- Performance: This model provided high accuracy and robustness in predictions.

**Evaluation Metrics:**

For each model, the following evaluation metrics were used:

1. Accuracy: The proportion of correctly predicted churned and non-churned customers.

2. Confusion Matrix: The confusion matrix provided insight into the true positives, true negatives, false positives, and false negatives.

Example:

from sklearn.metrics import confusion\_matrix

sns.heatmap(confusion\_matrix(Y\_test, pred\_knn), annot=True, fmt='g')

plt.show()

3. Precision, Recall, and F1-Score: These metrics were calculated to evaluate the trade-off between false positives and false negatives.

from sklearn.metrics import precision\_score, recall\_score, f1\_score

precision\_score(Y\_test, pred\_knn)

recall\_score(Y\_test, pred\_knn)

f1\_score(Y\_test, pred\_knn)

**Model Comparison:**

After training and evaluating multiple models, the following were the key takeaways:

- KNN and Random Forest performed comparably, achieving the highest accuracy in predicting churn.

- Random Forest had better generalization due to the ensemble approach, which reduces overfitting.

- Logistic Regression and SVM showed reasonable performance but were less accurate than Random Forest and KNN.

**Conclusion:**

This customer churn analysis project successfully used machine learning techniques to predict which customers are most likely to churn in a telecom company. With KNN and Random Forest emerging as the top models, the business can use these models to proactively identify at-risk customers and take measures to retain them.

The project highlighted the importance of feature engineering and preprocessing to ensure the quality of the predictions. Future improvements could include fine-tuning the models further, using advanced techniques like Gradient Boosting or Neural Networks, and incorporating additional data to enhance model accuracy.

**Future Scope:**

- Model Improvements: Explore advanced models such as XGBoost or Neural Networks to improve accuracy.

- Feature Engineering: Introduce new features such as customer sentiment analysis from interactions to enhance prediction capabilities.

- Deployment: Deploy the model into production using cloud services like AWS or Google Cloud.