Customer Churn Analysis and prediction

Version 1: Without Modeling (Business-Focused Summary)

1. Objective

- Analyze Telco customer data to understand churn patterns.
- Identify customers likely to leave.
- Help business plan retention campaigns for high-risk customers.

2. Data Overview

- Dataset: Cleaned Telco customer data (cleaned_telco_churn.csv).
- Features include customer demographics, subscription details, payment info, and service usage.
- Target variable: Customer churn (Yes/No).

3. Exploratory Data Analysis (EDA)

- Checked data distributions and relationships.
- Found churn rate around 22% in the dataset.
- Visualized key trends like churn rate by contract type, payment method, monthly charges.

4. Customer Segmentation

- Customers segmented based on risk of churning:
    - High Risk (~15-20%)
    - Medium Risk
    - Low Risk
- Focus is on high-risk customers for targeted retention.

5. Business Insights

- High monthly charges, short tenure, and certain services (like lack of online security or tech support) are key churn drivers.

- Customers on month-to-month contracts or using electronic check payments tend to churn more.
- These insights help design better offers and personalized retention campaigns.

6. Outcome

- Identified ~22% of customers likely to churn.
- Generated list of top 100 high-risk customers for immediate action.
- Created visualizations (pie charts, bar charts) to communicate churn risk and drivers.

Version 2: With Modeling (Technical/Data Science-Focused Summary)

1. Objective

- Build and evaluate machine learning models to predict customer churn.
- Understand feature importance to identify key churn drivers.
- Segment customers by churn risk probability.

2. Key Libraries Used

Library	**Purpose**
pandas	Data loading, cleaning, and manipulation
numpy	Numerical operations
matplotlib.pyplot	Data visualization
seaborn	Advanced statistical plots
sklearn.model_selection	Splitting data into train/test sets
sklearn.preprocessing	Encoding categorical data, scaling features
sklearn.linear_model	Logistic Regression model

Library	Purpose
sklearn.ensemble	Random Forest model
xgboost	Gradient boosting classifier
sklearn.metrics	Model evaluation (classification report, ROC curve, AUC)
joblib	Save/load models and scalers
os	File system operations (create folders)

---

3. Important Code Snippets

Data Loading and Preprocessing

python

CopyEdit

```
import pandas as pd
import numpy as np

df = pd.read_csv("cleaned_telco_churn.csv")
df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})

# Handle categorical variables
df_encoded = pd.get_dummies(df.drop('Churn', axis=1), drop_first=True)
X = df_encoded
y = df['Churn']
```

Train-Test Split with Stratification

python

CopyEdit

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.2, random_state=42, stratify=y)
```

Feature Scaling

python

CopyEdit

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Training Models

python

CopyEdit

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

# Logistic Regression
logreg = LogisticRegression(max_iter=1000)
logreg.fit(X_train_scaled, y_train)

# Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
```

```
# XGBoost
xgb = XGBClassifier(use_label_encoder=False, eval_metric='logloss')
xgb.fit(X_train, y_train)
```

Predictions and Evaluation

```
python
CopyEdit
from sklearn.metrics import classification_report, roc_auc_score, roc_curve
import matplotlib.pyplot as plt

y_pred_xgb = xgb.predict(X_test)
print(classification_report(y_test, y_pred_xgb))

# ROC Curve
y_proba = xgb.predict_proba(X_test)[:, 1]
fpr, tpr, _ = roc_curve(y_test, y_proba)
plt.plot(fpr, tpr, label=f"AUC = {roc_auc_score(y_test, y_proba):.2f}")
plt.legend()
plt.show()
```

Feature Importance Visualization

```
python
CopyEdit
import pandas as pd
import matplotlib.pyplot as plt

feat_importances = pd.Series(xgb.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
```

```
plt.title("Top 10 Important Features")
plt.show()
```

Save Model and Scaler

```
python
CopyEdit
import joblib
import os

os.makedirs("model", exist_ok=True)
joblib.dump(xgb, 'model/xgb_churn_model.pkl')
joblib.dump(scaler, 'model/scaler.pkl')
```

Predict and Extract High-Risk Customers

```
python
CopyEdit
import pandas as pd

results = pd.DataFrame({
  'customerID': id_test,
  'ActualChurn': y_test,
  'PredictedChurn': y_pred_xgb,
  'ChurnProbability': y_proba
})

high_risk_customers = results[results['PredictedChurn'] ==
1].sort_values(by='ChurnProbability', ascending=False)
high_risk_customers.head(100).to_csv('high_risk_customers_top100.csv', index=False)
```

---

4. Model Performance

Model	Accuracy	AUC
Logistic Regression	~0.80	0.82
Random Forest	~0.82	0.85
XGBoost	**~0.84**	**0.87**

---

5. Business Impact

- The model helps identify customers who are highly likely to churn.
- Enables targeted marketing and retention strategies.
- Reduces customer loss and increases revenue.
- Feature importance reveals actionable factors for service improvement.