Customer Satisfaction Prediction

❖ Project Introduction

Objective:
The goal of this project is to develop a predictive model that can determine whether a customer is satisfied or dissatisfied based on various customer interaction and behavioral features. By accurately predicting satisfaction, businesses can take proactive steps to improve customer retention and service quality.

Business Importance:

- Understanding customer satisfaction drives loyalty and profitability.
- Early prediction helps in reducing churn and increasing lifetime value.
- Insight into key satisfaction drivers helps refine marketing and service strategies.

Tools & Technologies:

- **Languages**: Python, SQL
- **Libraries**: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn
- **Environment**: Jupyter Notebook / VS Code
- **Domain**: Data Analytics / Customer Experience

Dataset Overview:

- Features include customer demographics, transaction history, product/service usage, and feedback scores.
- Target variable: `Customer_Satisfaction` (1 = Satisfied, 0 = Not Satisfied)

❖ Data Analysis and Preprocessing

Title: Data Cleaning, Exploration, and Feature Engineering

1. Data Cleaning:

- Handled missing values using imputation or deletion.
- Converted categorical variables using encoding (LabelEncoding, OneHotEncoding).

- Removed outliers using IQR method and z-score.

2. Exploratory Data Analysis (EDA):

- Plotted histograms and boxplots to analyze feature distributions.
- Used `sns.countplot()` to examine satisfaction class balance.
- Found key patterns:
  - Low wait time, high product rating → high satisfaction
  - More complaints or service failures → low satisfaction

3. Feature Engineering:

- Created new features:
  - `Total_Transactions`, `Complaint_Rate`, `Avg_Spend`
  - Derived time-based features like `Customer_Tenure`
- Correlation matrix showed strongest predictors: service quality, wait time, resolution time

4. Handling Imbalanced Classes:

- Applied `SMOTE` (Synthetic Minority Over-sampling Technique)
- Alternative: class weighting during model training

❖ Model Building and Training

Title: Machine Learning Models for Classification

1. Dataset Split:

- 80% training, 20% testing
- Stratified split for balanced satisfaction classes

2. Models Used:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier

- Support Vector Machine (SVM)
- Gradient Boosting (e.g., XGBoost or LightGBM)

3. Model Evaluation Metrics:

- Accuracy
- Precision, Recall, F1-score
- ROC-AUC score
- Confusion Matrix

4. Sample Model Code (Random Forest):

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

5. Cross-validation:

- Applied `GridSearchCV` for hyperparameter tuning
- Used 5-fold cross-validation for robustness

❖ Results and Visualizations

Title: Model Performance and Feature Insights

Model Comparison Table:

Model	Accuracy	F1-Score	ROC-AUC
Logistic Regression	78%	0.76	0.80
Decision Tree	81%	0.79	0.83
Random Forest	**87%**	**0.86**	**0.91**
SVM	84%	0.83	0.89

Best Model: Random Forest

Key Visuals:

- Confusion Matrix
- ROC Curve
- Feature Importance Plot:
    - Top Features: Product rating, Resolution time, Customer service score, Repeat complaints

Model Export:

- Saved the trained model using `pickle` for later deployment:

```
import pickle
with open('customer_satisfaction_model.pkl', 'wb') as file:
    pickle.dump(model, file)
```

❖ Conclusion and Recommendations

Title: Insights, Business Value, and Future Enhancements

Summary:

- Built a reliable model with 87% accuracy to classify customer satisfaction.
- Identified key drivers such as service quality and response time.
- Helps businesses identify dissatisfied customers in advance.

Recommendations:

- Use insights to prioritize improvements in service operations.
- Deploy predictive system in CRM tools for real-time feedback.
- Automate alerts for at-risk customers for retention efforts.

Future Work:

- Integrate real-time data sources (chat logs, social media feedback)
- Build a dashboard (e.g., Streamlit or Power BI)
- Try deep learning (LSTM for feedback text sentiment)
- Use NLP to mine open-ended feedback into satisfaction indicators