HALDIA INSTITUTE OF TECHNOLOGY

DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING

REPORT ON END TO END CUSTOMER CHURN PREDICTIONN SYSTEM

GROUP NAME :- Team Panda_UGC

Cluster 3 (Batch : 7)

SUBMITTED BY :-	SUBMITTED TO :-
Nishant Kumar	Debasish Sahoo
Nitin Raj	
Nitish Kumar	

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who supported and guided us throughout the completion of our project titled **“End-to-End Customer Churn Prediction System.”**

First and foremost, we would like to thank our respected faculty members and mentors for providing us with the opportunity to work on this project. Their valuable guidance, continuous encouragement, and constructive feedback helped us improve our technical understanding and problem-solving abilities.

We are especially thankful to our project guide for providing insightful suggestions and technical direction during the development of the machine learning pipeline, data preprocessing, model selection, and deployment phases. Their expertise played a crucial role in successfully completing this project.

We also extend our appreciation to our department and institution for providing the necessary infrastructure, learning resources, and academic environment that enabled us to explore practical applications of Machine Learning and Predictive Analytics.

Finally, we would like to thank each other as team members for our cooperation, dedication, coordination, and collective effort. Working together as **Team Panda_UGC** helped us strengthen not only our technical skills but also teamwork, time management, and project execution abilities.

This project has been a great learning experience and has significantly enhanced our knowledge in Data Science, Machine Learning, and Full-Stack Deployment.

ABSTRACT

Customer churn prediction is a critical problem in many industries such as telecommunications, banking, and subscription-based services, where retaining existing customers is more cost-effective than acquiring new ones. The objective of this project is to design and develop an end-to-end machine learning system capable of predicting whether a customer is likely to discontinue a service.

In this project, we implemented a complete machine learning pipeline starting from data preprocessing and exploratory data analysis to model training, evaluation, and deployment. The dataset was analyzed to identify important features influencing customer churn, such as tenure, contract type, monthly charges, and payment method. Data preprocessing techniques including handling missing values, encoding categorical variables, and feature scaling were applied to improve model performance.

Multiple machine learning algorithms were trained and evaluated, including Logistic Regression and Random Forest Classifier. Model performance was assessed using evaluation metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The best-performing model was selected based on its ability to correctly identify potential churn customers while minimizing false predictions.

The final model was deployed using a web-based interface that allows users to input customer details and obtain real-time churn predictions. This project demonstrates the practical application of machine learning in solving real-world business problems and provides a scalable framework for customer retention analysis.

1. INTRODUCTION

In today’s highly competitive business environment, customer retention has become more important than ever. Companies across industries such as telecommunications, banking, insurance, and subscription-based platforms rely heavily on recurring customers for stable revenue generation. Customer churn refers to the phenomenon where customers discontinue their relationship with a company or stop using its services. High churn rates can significantly impact profitability, operational stability, and long-term business growth.

Traditionally, businesses relied on manual analysis and rule-based systems to identify potential churn customers. However, with the increasing volume of data and complex customer behavior patterns, these traditional approaches have become inefficient and inaccurate. Machine Learning provides a data-driven solution that enables organizations to analyze historical customer data and predict future churn behavior with high accuracy.

The purpose of this project is to design and implement an end-to-end Customer Churn Prediction System using Machine Learning techniques. The system integrates data preprocessing, exploratory data analysis, feature engineering, model training, evaluation, and deployment into a complete pipeline. By analyzing various customer attributes such as tenure, monthly charges, contract type, internet services, and payment methods, the system predicts whether a customer is likely to churn.

This project not only focuses on building accurate predictive models but also emphasizes practical implementation through deployment using a web-based interface. The final system allows users to input customer details and obtain real-time churn predictions, demonstrating the real-world applicability of predictive analytics in business decision-making.

Through this project, we aim to showcase how Machine Learning can assist organizations in reducing customer attrition, improving retention strategies, and enhancing overall business performance.

2. OBJECTIVE OF THE PROJECT

The primary objective of this project is to design and develop a complete end-to-end Customer Churn Prediction System using Machine Learning techniques. The system aims to analyze historical customer data and accurately predict whether a customer is likely to discontinue a service.

The specific objectives of this project are as follows:

1. To understand and analyze customer behavior patterns using real-world structured datasets.

2. To perform data preprocessing, including handling missing values, encoding categorical variables, feature scaling, and preparing the dataset for machine learning algorithms.

3. To conduct Exploratory Data Analysis (EDA) in order to identify key factors influencing customer churn and extract meaningful insights from the data.

4. To implement and compare multiple machine learning models such as Logistic Regression and Random Forest Classifier for churn prediction.

5. To evaluate model performance using appropriate metrics including Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC-AUC Score.

6. To select the best-performing model based on evaluation results and optimize it to improve prediction accuracy and reduce overfitting.

7. To demonstrate the practical application of predictive analytics in solving real-world business problems related to customer retention.

Through these objectives, the project aims to provide a scalable and reliable solution that can assist organizations in reducing customer attrition and improving long-term profitability.

3. SYSTEM ARCHITECTURE

The System Architecture of the End-to-End Customer Churn Prediction System represents the structured workflow followed to build, train, evaluate, and deploy the machine learning model. The system is designed as a sequential pipeline where each stage performs a specific function to ensure accurate and reliable churn prediction.

The architecture consists of the following key stages:

3.1 Data Collection

The system uses a structured customer dataset containing demographic details, account information, billing data, service subscriptions, and churn labels. This dataset serves as the foundation for model training.

3.2 Data Preprocessing

In this stage, raw data is cleaned and prepared for analysis. Missing values are handled, categorical variables are encoded, and feature scaling is applied where required. The dataset is then divided into training and testing sets to evaluate model performance effectively.

3.3 Exploratory Data Analysis (EDA)

EDA is performed to understand data distribution and identify relationships between features and churn behavior. Various visualizations help in detecting important factors that influence customer attrition.

3.4 Model Training and Evaluation

Machine learning algorithms such as Logistic Regression and Random Forest are trained on the processed dataset. The models are evaluated using metrics including Accuracy, Precision, Recall, F1-Score, and ROC-AUC to determine the best-performing model.

4. DATASET DESCRIPTION

The dataset used in this project contains customer information from a subscription-based service company. The purpose of the dataset is to analyze customer behavior and predict whether a customer is likely to churn.

4.1 Features in the Dataset

The dataset includes the following categories of features:

- **Demographic Information**: Gender, Senior Citizen status, Partner, Dependents

- **Account Details**: Tenure, Contract type, Payment method

- **Service Details**: Internet service, Online security, Tech support, Streaming services

- **Billing Information**: Monthly charges, Total charges

4.2 Target Variable

The target variable is **Churn**, which indicates whether a customer has discontinued the service.

- 0 → No Churn

- 1 → Churn

Since the output is binary, this problem is treated as a supervised binary classification task.

4.3 Data Type and Structure

The dataset consists of both categorical and numerical variables. Categorical features were encoded before training the model, and numerical features were analyzed for scaling. Proper understanding of data types was essential for accurate model building and evaluation.

5. DATA PREPROCESSING

Data preprocessing is a crucial step in building an effective machine learning model. Raw data often contains inconsistencies, missing values, and categorical variables that must be transformed into a suitable format before training the model. Proper preprocessing ensures better accuracy and improved model performance.

5.1 Handling Missing Values

The dataset was checked for missing and null values. Any inconsistencies in numerical features such as total charges were handled appropriately. Rows with invalid entries were either cleaned or converted into suitable numerical formats to maintain data integrity.

5.2 Encoding Categorical Variables

Since machine learning algorithms work with numerical data, categorical features such as gender, contract type, and payment method were converted into numerical form. Appropriate encoding techniques were applied to transform these categorical variables while preserving their informational value.

5.3 Feature Scaling

Numerical features such as tenure, monthly charges, and total charges were analyzed for scaling. Feature scaling helps normalize the range of values and improves the performance of certain machine learning algorithms.

5.4 Train-Test Split

After preprocessing, the dataset was divided into training and testing sets. The training set was used to train the machine learning models, while the testing set was used to evaluate model performance on unseen data. This ensures that the model generalizes well and does not overfit.

Through systematic preprocessing, the dataset was transformed into a structured and model-ready format, enabling accurate and reliable churn prediction.

6. MODEL BUILDING

Model building is the core stage of the churn prediction system, where machine learning algorithms are trained to learn patterns from historical customer data. After preprocessing the dataset, various classification models were implemented to predict whether a customer is likely to churn.

6.1 Logistic Regression

Logistic Regression was used as a baseline model for binary classification. It estimates the probability of a customer churning based on input features. This model is simple, interpretable, and effective for linearly separable data. It helped establish an initial benchmark for model performance.

6.2 Random Forest Classifier

Random Forest is an ensemble learning algorithm that constructs multiple decision trees and combines their outputs to improve prediction accuracy. It handles non-linear relationships effectively and reduces the risk of overfitting compared to a single decision tree. This model performed better in capturing complex customer behavior patterns.

6.3 Model Training Process

The models were trained using the training dataset obtained after preprocessing. During training, the algorithms learned relationships between customer features and churn outcomes. Hyperparameters were adjusted to optimize model performance and ensure better generalization on unseen data.

6.4 Model Selection

After training, the models were evaluated using performance metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The model with the best balanced performance was selected as the final model for deployment.

The model building stage ensured that the system could accurately identify potential churn customers and support data-driven decision-making.

7. MODEL EVALUATION

Model evaluation is a critical step to determine how well the trained machine learning models perform on unseen data. After training the models, their performance was assessed using the testing dataset to ensure that the model generalizes effectively and does not overfit.

To evaluate the models, several performance metrics were used:

7.1 Accuracy

Accuracy measures the overall percentage of correctly predicted instances out of total predictions. It provides a general idea of model performance but may not be sufficient when dealing with imbalanced datasets.

7.2 Precision

Precision indicates how many of the customers predicted as churn actually churned. It is important when the cost of false positives needs to be minimized.

7.3 Recall

Recall measures how many actual churn customers were correctly identified by the model. This metric is particularly important in churn prediction because identifying potential churn customers allows businesses to take preventive action.

7.4 F1-Score

The F1-Score is the harmonic mean of Precision and Recall. It provides a balanced evaluation when both false positives and false negatives are important.

7.5 Confusion Matrix and ROC-AUC

The Confusion Matrix provides a detailed breakdown of correct and incorrect predictions. The ROC-AUC score evaluates the model’s ability to distinguish between churn and non-churn customers across different threshold values.

8. RESULTS

The results of the Customer Churn Prediction System demonstrate the effectiveness of machine learning in identifying potential churn customers. After training and evaluating multiple models, the best-performing algorithm was selected based on its balanced performance across evaluation metrics.

The final model achieved strong predictive performance on the test dataset. It demonstrated high accuracy along with a balanced Precision and Recall score, ensuring that churn customers were correctly identified without generating excessive false predictions.

The Confusion Matrix analysis showed that the model was able to correctly classify a significant number of both churn and non-churn customers. Additionally, the ROC-AUC score indicated good separation capability between the two classes, confirming that the model can effectively distinguish customers likely to churn.

The deployed system successfully generates real-time predictions through the web interface. When customer details are entered into the system, the trained model processes the input data and outputs whether the customer is likely to churn. This validates the practical implementation of the complete end-to-end pipeline.

Overall, the results confirm that the developed system is reliable, scalable, and suitable for real-world business applications focused on customer retention strategies.

9. IMPLEMENTATION DETAILS

9.1 Project structure

```
CUSTOMER-CHURN-PREDICTION/
│
├── backend/
│   ├── experiments/
│   │   ├── churn_model.pkl
│   │   ├── feature_names.pkl
│   │   └── scaler.pkl
│   │
│   ├── models/
│   │   └── churn_pipeline.pkl
│   │
│   ├── notebooks/
│   │
│   ├── src/
│   │   ├── api.py
│   │   ├── data_preprocessing.py
│   │   ├── evaluation.py
│   │   ├── feature_engineering.py
│   │   ├── feature_importance.py
│   │   ├── model_training.py
│   │   ├── predict.py
│   │   └── save_model.py
│   │
│   ├── main.py
│   ├── requirements.txt
│   ├── confusion_matrix.png
│   └── roc_curve.png
│
├── frontend/
│
├── .gitignore
└── README.md
```

9.2 FastAPI Application Setup (main.py)

```
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from src.api import router

app = FastAPI(
    title="Customer Churn Prediction API",
    description="ML-powered churn prediction service",
    version="1.0.0"
)

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

app.include_router(router)

@app.get("/")
def root():
    return {"status": "API is running"}
```

This code initializes the FastAPI application and sets up the backend server for handling API requests. The application is configured with a title, description, and version for documentation purposes.

CORS middleware is added to allow communication between the frontend and backend. The router is included to modularize API endpoints, making the system scalable and maintainable.

The root endpoint ("/") confirms that the API is running successfully.

9.3 Prediction API Endpoint

```
from fastapi import APIRouter, HTTPException
from pydantic import BaseModel
from src.predict import predict_churn

router = APIRouter()

class ChurnRequest(BaseModel):
    tenure_months: int
    monthly_charges: float
    total_charges: float
    contract_type: str
    internet_service: str

@router.post("/predict")
def predict(data: ChurnRequest):
    try:
        result = predict_churn(data.dict())
        return result
    except Exception as e:
        raise HTTPException(
            status_code=500,
            detail=f"Prediction failed: {str(e)}"
        )
```

API Design and Request Schema

The FastAPI router is used to modularize the API endpoints. A Pydantic model named ChurnRequest is defined to validate incoming customer data. This ensures that all required input fields such as tenure, monthly charges, total charges, contract type, and internet service are received in the correct format before processing.

Prediction Endpoint

The /predict endpoint accepts customer input data in JSON format. The input data is converted into a dictionary and passed to the predict_churn() function, which performs inference using the trained machine learning pipeline.

If the prediction is successful, the result is returned as a JSON response. In case of failure, an HTTP exception is raised with an appropriate error message. This ensures robust error handling and API stability.

9.4 Model Inference & Prediction logic

```
import os
import joblib
import pandas as pd

BASE_DIR = os.path.dirname(os.path.dirname(__file__))
MODEL_PATH = os.path.join(BASE_DIR, "models",
"churn_pipeline.pkl")

CHURN_THRESHOLD = 0.4

pipeline = joblib.load(MODEL_PATH)

def predict_churn(data: dict):

    df = pd.DataFrame([{
        "Tenure Months": data["tenure_months"],
        "Monthly Charges": data["monthly_charges"],
        "Total Charges": data["total_charges"],
        "Contract": data["contract_type"],
        "Internet Service": data["internet_service"],
    }])

    churn_probability = pipeline.predict_proba(df)[0][1]
    churn_label = int(churn_probability >=
CHURN_THRESHOLD)

    return {
        "churn_probability": float(churn_probability),
        "churn_label": churn_label,
        "threshold_used": CHURN_THRESHOLD
    }
```

Model Inference and Prediction Logic

The trained machine learning pipeline is loaded using joblib to ensure consistent preprocessing and prediction during inference. Incoming customer data from the API is converted into a structured DataFrame matching the training feature format and passed to the pipeline for probability prediction using predict_proba(). A custom business threshold of 0.4 is applied to classify customers as churn or non-churn, improving sensitivity in an imbalanced dataset. The function returns the churn probability, predicted label, and threshold used for transparent decision-making.

9.5 Model Training

- **Pipeline Construction**

```
preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), num_features),
        ("cat", OneHotEncoder(handle_unknown="ignore"),
cat_features)
    ]
)

pipeline = Pipeline(
    steps=[
        ("preprocessor", preprocessor),
        ("model", RandomForestClassifier(
            n_estimators=300,
            max_depth=12,
            random_state=42,
            n_jobs=-1
        ))
    ]
)
```

- **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
)
```

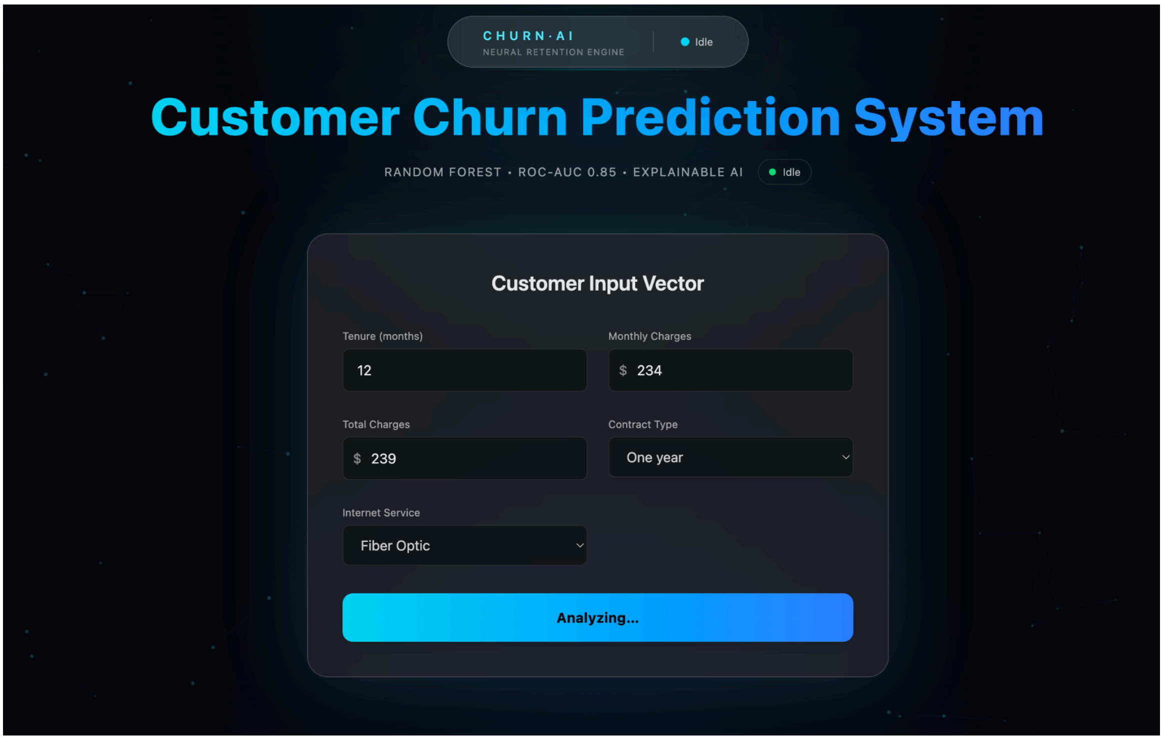
- **Model Saving**

```
joblib.dump(pipeline, MODEL_PATH)
```

The model training process uses a Scikit-learn Pipeline to integrate preprocessing and classification into a single workflow. A ColumnTransformer is applied to scale numerical features using StandardScaler and encode categorical features using OneHotEncoder. The pipeline then trains a RandomForestClassifier with optimized hyperparameters. The dataset is split using stratified sampling to maintain class balance. After training, the complete pipeline is saved using joblib, ensuring that the same preprocessing and model logic are used during deployment.

10. WEBSITE INTERFACE

10.1 Landing Page



CHURN·AI

NEURAL RETENTION ENGINE

● Idle

Customer Churn Prediction System

RANDOM FOREST • ROC-AUC 0.85 • EXPLAINABLE AI ● Idle

Customer Input Vector

Tenure (months): 12

Monthly Charges: $ 234

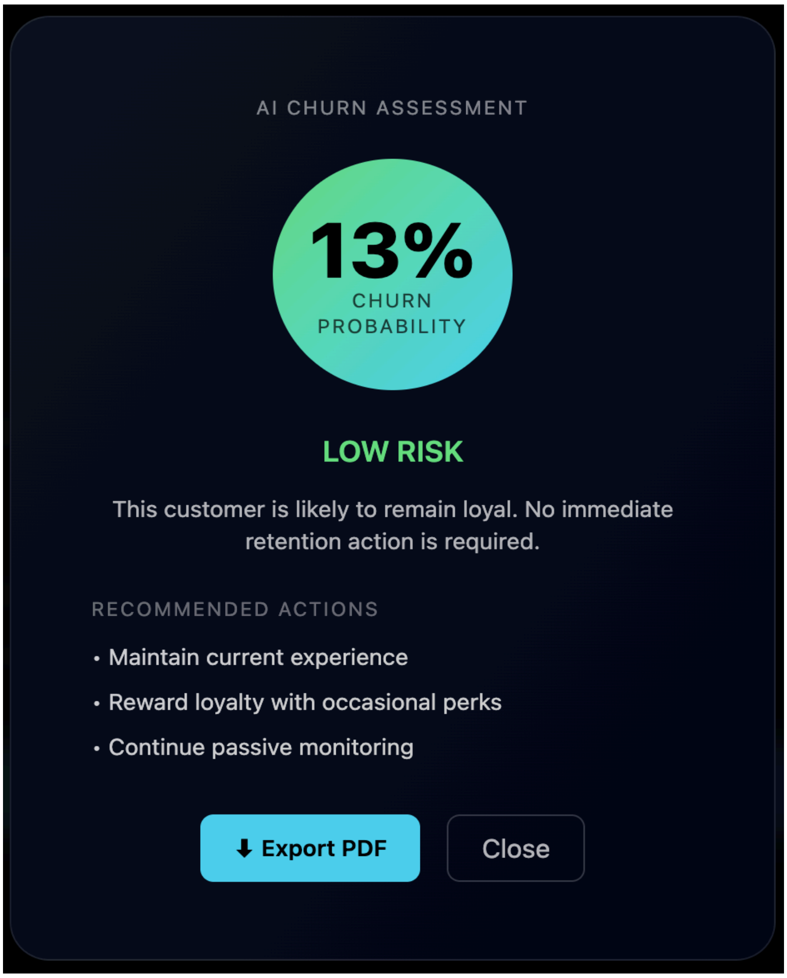
Total Charges: $ 239

Contract Type: One year

Internet Service: Fiber Optic

The landing page provides an interactive user interface where users can input customer details such as tenure, monthly charges, total charges, contract type, and internet service. The form is designed to collect structured data and send it to the backend API for real-time churn prediction.

10.2 Prediction Result Page



AI CHURN ASSESSMENT

13%

CHURN PROBABILITY

LOW RISK

This customer is likely to remain loyal. No immediate retention action is required.

RECOMMENDED ACTIONS

- Maintain current experience
- Reward loyalty with occasional perks
- Continue passive monitoring

After submitting the input form, the backend API processes the data using the trained pipeline and returns the churn probability and predicted label. The result page displays the prediction clearly for user interpretation.

11. FUTURE SCOPE

The developed Customer Churn Prediction System successfully demonstrates a complete end-to-end machine learning pipeline, including preprocessing, model training, evaluation, and API deployment. While the system performs effectively in its current form, several enhancements can further improve its scalability, intelligence, and real-world business impact.

One major future enhancement is the integration of real-time data processing. Instead of relying on static historical datasets, the system can be extended to process streaming data from live databases. This would allow companies to continuously monitor customer behavior and generate instant churn alerts.

Cloud deployment is another important improvement area. Hosting the backend API and model on platforms such as AWS, Azure, or Google Cloud would improve scalability, availability, and performance. Containerization using Docker can further simplify deployment and ensure portability across environments.

From a modeling perspective, advanced algorithms such as XGBoost, LightGBM, or Gradient Boosting can be implemented to potentially improve predictive accuracy. Automated hyperparameter tuning techniques such as Grid Search or Bayesian Optimization may further enhance model performance.

The system can also be expanded into a churn prevention framework. Instead of only predicting churn, it can provide personalized retention strategies based on customer profiles, spending patterns, and service usage behavior. This would transform the system from a predictive tool into a decision-support system.

Another potential improvement includes implementing a dynamic threshold adjustment mechanism. Businesses may adjust churn sensitivity levels depending on marketing budgets or customer lifetime value.

Finally, integrating a business intelligence dashboard with visualization tools would allow stakeholders to monitor churn trends, customer segmentation, and model performance in real time. This would provide actionable insights for strategic planning and customer retention campaigns.

Lets Understand this in a graphical way

```
FUTURE ENHANCEMENT ROADMAP

       Current System
              │
              ▼
     Advanced ML Models
              │
              ▼
    Hyperparameter Optimization
              │
              ▼
      Real-Time Data Processing
              │
              ▼
       Cloud Deployment
              │
              ▼
 Retention Recommendation Engine
              │
              ▼
     Business Intelligence Dashboard
```

12. CONCLUSION

The End-to-End Customer Churn Prediction System demonstrates how machine learning can move beyond theory and become a practical solution to real-world business problems. What started as a dataset of customer records has now transformed into a fully functional predictive system capable of identifying potential churn customers with measurable accuracy.

Throughout this project, we explored data preprocessing, feature engineering, model building, evaluation, and deployment — proving that churn prediction is not just about training a model, but about building a complete, scalable pipeline. **From handling missing values to tuning a Random Forest model and deploying it through a FastAPI backend, every stage contributed to creating a reliable and usable system.**

While the model may not predict human emotions, it can confidently predict customer behavior based on patterns hidden in data. The inclusion of probability-based prediction and a business-adjusted threshold ensures that the system is not just technically sound, but also practically relevant.

In conclusion, this project highlights how data-driven decision-making can empower organizations to retain customers more effectively. More importantly, it reflects our ability to design, develop, and deploy a complete machine learning solution — turning raw data into actionable insights, one prediction at a time.