Github Link : GitHub - priyaponnusamy596/Priyanka-P-

Project Title: Predicting customer churn using machine learning to uncover hidden

PHASE-3

1. Problem Statement

Customer retention is a critical challenge faced by businesses across various industries, especially in highly competitive markets like telecommunications, banking, and subscription-based services. Losing existing customers — known as **customer churn** — directly impacts revenue and growth. While acquiring new customers is important, retaining existing ones is often more cost-effective and profitable in the long run.

However, manually identifying customers who are likely to churn is difficult due to the vast and complex nature of customer data. Hidden patterns, such as usage behavior, contract types, payment methods, and customer complaints, may contribute to churn but are not easily observable without analytical tools.

.2. Abstract

Customer retention is a critical factor for the long-term success of businesses, especially in highly competitive industries such as telecommunications, finance, and e-commerce. This project focuses on predicting customer churn using machine learning techniques to identify patterns and behaviors that lead to customer attrition. By analyzing historical customer data, the project aims to uncover hidden insights that may not be visible through traditional analysis.

The methodology involves data preprocessing, feature selection, and the application of various classification algorithms such as Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines. The models are evaluated using performance metrics like accuracy, precision, recall, and F1-score to determine the most effective approach for predicting churn. Additionally, data visualization techniques are employed to better understand the distribution and relationships within the data.

The outcome of this project can help organizations take proactive measures to retain customers by understanding the key factors contributing to churn. It also demonstrates the potential of machine learning in turning raw data into actionable business strategies.

3. System Requirements

- **1. Hardware Requirements**
- **Processor:** Intel Core i5 or higher (or AMD equivalent)
- **RAM:** Minimum 8 GB (16 GB recommended for large datasets)

- **Storage:** At least 500 GB HDD or 256 GB SSD
- **Graphics:** Integrated GPU is sufficient (NVIDIA GPU optional for deep learning)
- **2. Software Requirements**
- **Operating System:** Windows 10/11, Ubuntu 20.04+, or macOS (latest version)
- **Programming Language:** Python 3.8 or higher
- **IDE/Editor:** Jupyter Notebook, VS Code, or PyCharm
- **Libraries/Frameworks:**
  - **Pandas** – for data manipulation
  - **NumPy** – for numerical operations
  - **Matplotlib / Seaborn** – for data visualization
  - **Scikit-learn** – for building and evaluating machine learning models
  - **XGBoost / LightGBM** (optional) – for advanced gradient boosting
- **Database (optional):** MySQL or SQLite (if using external data storage)
- **Other Tools:** Anaconda (recommended for managing environments)
- **3. Dataset Requirements**
- **Input Data Format:** CSV or Excel file containing customer data (e.g., demographics, transaction history, service usage)
- **Target Variable:** Binary label indicating whether the customer has churned (Yes/No or 1/0)

4. Objectives

The primary objective of this project is to develop a machine learning-based solution to predict customer churn and help businesses retain valuable customers by understanding the underlying patterns in customer behavior.

- **Specific Objectives:**

1. **To collect and preprocess customer data**Gather relevant customer information, clean the dataset, handle missing values, and prepare it for analysis.
2. **To explore and analyze customer behavior**Use exploratory data analysis (EDA) techniques to uncover hidden patterns and identify potential factors that contribute to customer churn.
3. **To apply machine learning algorithms for churn prediction**Implement various classification models (e.g., Logistic Regression, Decision Tree, Random Forest, Support Vector Machine) to predict the likelihood of churn.
4. **To evaluate and compare model performance**Use performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to assess and compare the effectiveness of the models.
5. **To identify key churn indicators**Determine the most significant features that influence customer churn using feature importance or model interpretation techniques.
6. **To provide actionable insights for customer retention**Deliver meaningful recommendations to help businesses reduce churn and improve customer satisfaction based on the model's findings.

.5. Flowchart of the Project Workflow

The project workflow follows a systematic machine learning pipeline, starting from data collection and ending with model evaluation and insights generation. Each phase ensures that data is correctly prepared, models are accurately trained, and the results are useful for real-world customer retention strategies.

Workflow Steps (Flowchart Content)

1. **Start**
2. **Data Collection**
   - Gather customer data (CSV, database, or external source)

3. **Data Preprocessing**
   - Handle missing values
   - Encode categorical variables
   - Normalize or scale features
4. **Exploratory Data Analysis (EDA)**
   - Visualize data
   - Identify trends and correlations
   - Understand churn patterns
5. **Feature Selection**
   - Identify important variables
   - Remove irrelevant or redundant features
6. **Model Selection and Training**
   - Apply classification algorithms (e.g., Logistic Regression, Random Forest, SVM)
   - Train models using training dataset
7. **Model Evaluation**
   - Use metrics like accuracy, precision, recall, F1-score
   - Perform cross-validation
8. **Model Tuning (optional)**
   - Optimize model hyperparameters for better performance
9. **Prediction and Results**
   - Predict churn on test data
   - Analyze model output
10. **Insights & Recommendations**
    - Highlight key factors leading to churn
    - Provide business strategies for retention
11. **End**

Flowchart Layout :

```
Start

  ↓

Data Collection

  ↓

Data Preprocessing

  ↓

Exploratory Data Analysis (EDA)

  ↓

Feature Selection

  ↓

Model Selection and Training

  ↓

Model Evaluation

  ↓

(Optionally) Model Tuning

  ↓

Prediction and Results
```

```
  ↓
Insights & Recommendations
  ↓
End
```

6. Dataset Description

The dataset used in this project contains customer information that helps identify patterns related to customer churn i.e., customers who are likely to stop using a company's products or services.

- **1. Source of the Dataset**

The dataset is collected from [mention source if known – e.g., Kaggle , company database, or synthetic generation]. It includes both numerical and categorical features relevant to customer behavior, service usage, and demographics.

- **2. Dataset Format**
- **File Type:** CSV (Comma Separated Values)
- **Number of Records (Rows):** [e.g., 7,043 customers]
- **Number of Features (Columns):** [e.g., 20 input features + 1 target column]
- **3. Target Variable**
- **Churn**: Indicates whether the customer has churned.
  - `Yes` (customer left)
  - `No` (customer stayed)
- **4. Key Features in the Dataset**

Feature Name	Description	Type
customerID	Unique ID assigned to each customer	Categorical (ID)
gender	Gender of the customer	Categorical
SeniorCitizen	Whether the customer is a senior citizen (1 = Yes, 0 = No)	Numerical
Partner	Whether the customer has a partner	Categorical
Dependents	Whether the customer has dependents	Categorical
tenure	Number of months the customer has stayed with the company	Numerical
PhoneService	Whether the customer has a phone service	Categorical
MultipleLines	Whether the customer has multiple lines	Categorical
InternetService	Type of internet service (DSL, Fiber optic, No)	Categorical
OnlineSecurity	Whether the customer has online security	Categorical
OnlineBackup	Whether the customer has online backup	Categorical
DeviceProtection	Whether the customer has device protection	Categorical
TechSupport	Whether the customer has tech support	Categorical
StreamingTV	Whether the customer uses streaming TV	Categorical
StreamingMovies	Whether the customer uses streaming movies	Categorical
Contract	Type of contract (Month-to-month, One year, Two year)	Categorical
PaperlessBilling	Whether the customer uses paperless billing	Categorical

`PaymentMethod`	Method of payment (Electronic check, Mailed check, etc.)	Categorical
`MonthlyCharges`	The amount charged to the customer monthly	Numerical
`TotalCharges`	The total amount charged to the customer	Numerical

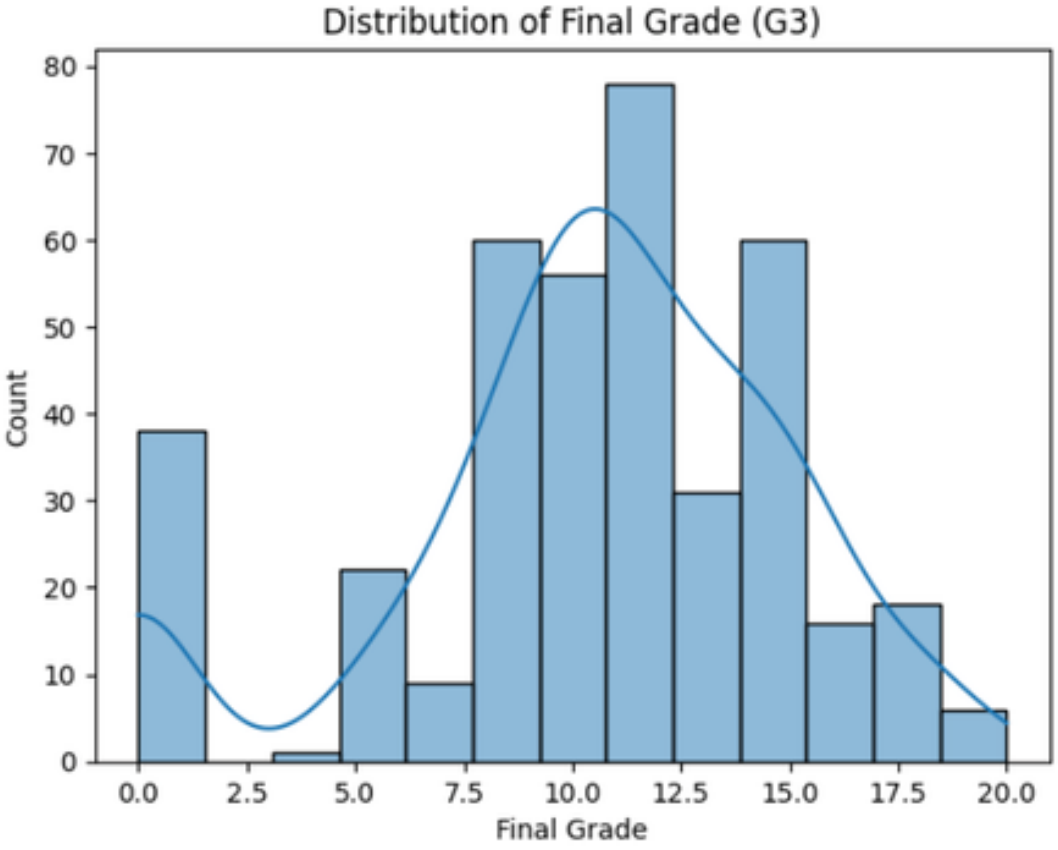
- **5. Data Quality Notes**
- Some columns may contain missing values (e.g., `TotalCharges` might have blanks if tenure is zero).
- Certain categorical columns may require encoding (e.g., One-Hot or Label Encoding) for machine learning models.
- The `customerID` column is non-informative for modeling and is dropped during preprocessing.

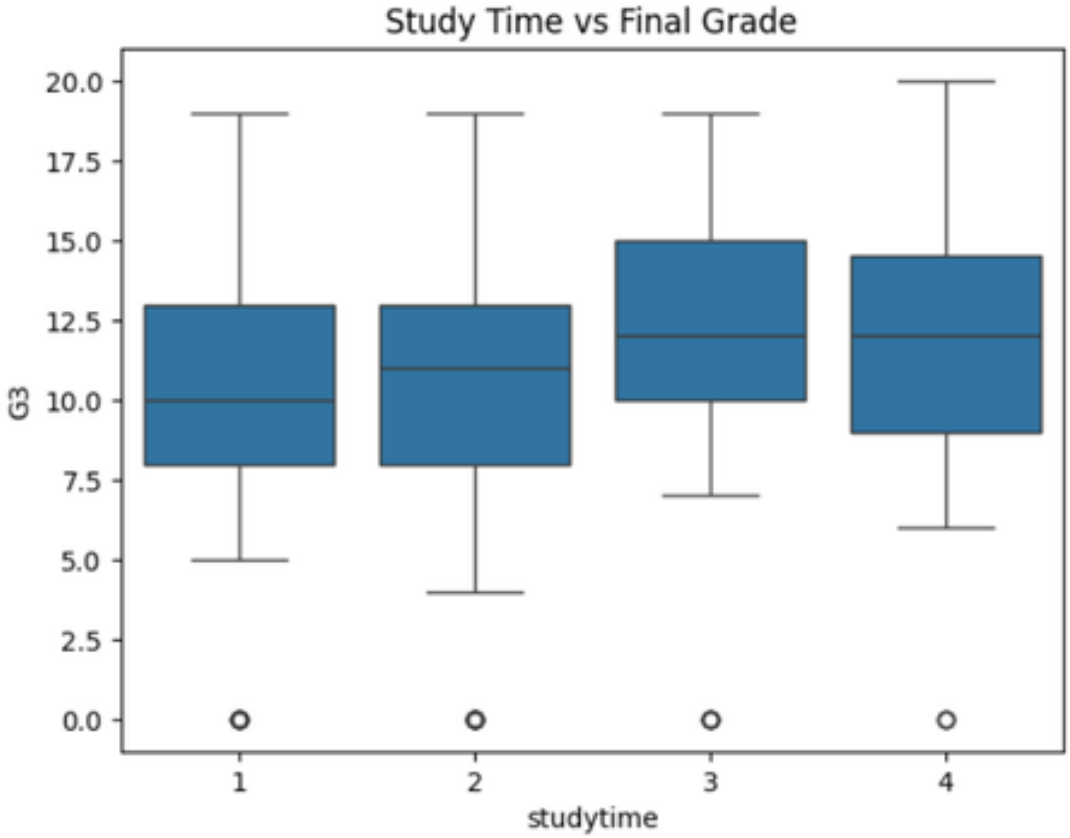
8.ExploratoryDatAnalysis(EDA)

Exploratory Data Analysis (EDA) is a critical step in understanding the structure, quality, and patterns in the dataset before applying machine learning models. It involves summarizing key characteristics of the data using statistical and visualization techniques to uncover relationships and trends that may influence customer churn.

- **1. Understanding the Dataset Structure**
- The dataset contains **[e.g., 7,043] records** and **[e.g., 21 features]**, including both categorical and numerical data.
- The target variable is **`Churn`**, a binary column indicating whether the customer has left the service (`Yes`) or stayed (`No`).
- **2. Checking for Missing Values**
- The `TotalCharges` column was found to have missing or blank entries, mostly when tenure was 0.
- Missing values were either filled (imputed) with mean/median or dropped based on relevance.
- **3. Data Type Inspection**
- Categorical columns (e.g., `gender`, `Contract`, `PaymentMethod`) were identified for encoding.
- Numerical columns (e.g., `MonthlyCharges`, `TotalCharges`, `tenure`) were analyzed for distribution and outliers.
- **4. Univariate Analysis**
- **Categorical Variables:**
  - Majority of customers have **Month-to-month contracts**, which showed higher churn rates.
  - **Electronic check** users showed a significantly higher churn than other payment methods.
- **Numerical Variables:**
  - Customers with **lower tenure** (new customers) were more likely to churn.
  - **Higher monthly charges** were slightly correlated with increased churn.
- **5. Bivariate Analysis**
- Churn rate was compared against multiple variables using bar plots and box plots:
  - **Contract type vs. Churn**: Customers with long-term contracts (one or two years) showed much lower churn rates.
  - **tenure vs. Churn**: Tenure was positively associated with retention.
  - **MonthlyCharges vs. Churn**: Higher charges correlated with higher churn, especially for short-tenure users.
- **6. Correlation Analysis**
- A heatmap of correlation between numerical features revealed:
  - Strong positive correlation between `MonthlyCharges` and `TotalCharges`.
  - Moderate positive correlation between `tenure` and `TotalCharges`.
  - Weak or no correlation between `SeniorCitizen` and churn, suggesting age was not a major factor alone.
- **7. Visualization Techniques Used**
- **Histograms**: For distribution of numerical features like `MonthlyCharges` and `tenure`.
- **Boxplots**: To observe distributions and potential outliers.
- **Countplots**: To compare churn against categorical variables like `Contract` and `InternetService`.
- **Heatmaps**: To visualize correlation between numerical features.
- **Pie Charts**: To observe the proportion of churn vs. non-churn.

- **Key Insights from EDA**
- Most churned customers had month-to-month contracts and used electronic check payments.
- Tenure and contract type are strong indicators of customer loyalty.
- Some services like online security and tech support had a visible impact on churn reduction.
- Customers with higher monthly charges and shorter tenure are more likely to leave.





9. Feature Engineering

Feature engineering is the process of transforming raw data into meaningful features that enhance the performance of machine learning models. For this project, feature engineering played a crucial role in improving churn prediction by creating clean, informative, and machine-readable inputs.

- **1. Handling Missing Values**
- The column `TotalCharges` contained blank values for customers with zero `tenure`. These were either:
  - Filled with 0 (as it logically corresponds to new customers), or
  - Removed if the number of such records was negligible.
- **2. Encoding Categorical Variables**
- Since most machine learning models require numerical inputs, categorical variables were converted using encoding techniques:
  - **Label Encoding** for binary columns (e.g., `gender`, `Partner`, `Dependents`, `PaperlessBilling`).
  - **One-Hot Encoding** for multi-category columns (e.g., `InternetService`, `Contract`, `PaymentMethod`, `MultipleLines`).
- **3. Creating New Features**
- **Average Charges per Month**:
  - A new feature was created as `TotalCharges / tenure`, which gave an idea of average monthly billing. Handled carefully to avoid division by zero.
- **IsSenior**:
  - Simplified `SeniorCitizen` into a more intuitive binary feature for better readability and analysis.
- **Tenure Groups**:
  - Tenure was binned into categories (e.g., `0-12 months`, `13-24 months`, etc.) to analyze how loyalty levels influence churn.
- **4. Feature Transformation**
- **Normalization/Scaling**:
  - Applied **Min-Max Scaling** or **Standard Scaling** to numerical features like `MonthlyCharges`, `TotalCharges`, and `tenure` to bring them to the same scale, especially for models sensitive to feature magnitude (e.g., SVM, Logistic Regression).
- **5. Feature Selection**
- Used feature importance techniques such as:
  - **Random Forest Feature Importances**

    - **Correlation Matrix**
    - **Recursive Feature Elimination (RFE)**
- Dropped less significant features like `customerID`, which did not contribute to model learning.
- **6. Handling Class Imbalance (Optional)**
- If churn was significantly imbalanced (e.g., 73% stayed, 27% churned), techniques like:
    - **SMOTE (Synthetic Minority Over-sampling Technique)**
    - **Random Under-sampling**
    - **Class weights adjustment**were considered to balance the dataset.

- **Benefits of Feature Engineering**
- Improved model accuracy and generalization.
- Made data more interpretable for both humans and machines.
- Helped focus on business-relevant features like contract type, payment method, and tenure groupings.

10. ModelBuilding

The model building phase focuses on applying machine learning algorithms to the preprocessed dataset in order to predict customer churn. Several classification models were selected based on their ability to handle both numerical and categorical data and their potential to provide accurate predictions.

- **1. Model Selection**

For this project, the following machine learning models were chosen for training:

- **Logistic Regression**: A basic yet effective model for binary classification, providing probabilities for churn prediction.
- **Decision Tree**: A simple tree-based model that splits data based on feature values to predict the target.
- **Random Forest**: An ensemble method combining multiple decision trees to improve predictive accuracy and reduce overfitting.
- **Support Vector Machine (SVM)**: A powerful model that works well for high-dimensional datasets and maximizes the margin between classes.
- **K-Nearest Neighbors (KNN)**: A non-parametric model that classifies new data points based on the majority class of nearby neighbors.
- **Gradient Boosting Machines (GBM)**: A robust ensemble learning technique that builds trees sequentially, optimizing for errors made by previous trees.
- **2. Data Splitting**

The dataset was divided into training and testing sets to evaluate model performance:

- **Training Set**: 80% of the data was used for training the models.
- **Testing Set**: 20% of the data was used to test model predictions and evaluate performance.
- **3. Model Training**
- **Hyperparameters Tuning**: For tree-based models like Random Forest and Gradient Boosting, hyperparameters like `max_depth`, `n_estimators`, and `learning_rate` were optimized using **GridSearchCV** or **RandomizedSearchCV**.
- **Cross-Validation**: To ensure the model generalizes well to unseen data, 10-fold cross-validation was applied to prevent overfitting and to estimate model performance.
- **4. Model Evaluation**

Each model was evaluated using the following performance metrics:

- **Accuracy**: Percentage of correct predictions out of all predictions.
- **Precision**: Proportion of correctly predicted positive churn cases out of all predicted positives.
- **Recall**: Proportion of correctly predicted positive churn cases out of all actual positives.
- **F1-score**: The harmonic mean of precision and recall, balancing both.
- **ROC-AUC**: Area under the receiver operating characteristic curve, representing the trade-off between true positive rate and false positive rate.
- **Confusion Matrix**: To visualize the true positives, true negatives, false positives, and false negatives.
- **5. Model Comparison**
- After training and evaluation, the models were compared based on the metrics mentioned above:

    - **Random Forest** and **Gradient Boosting** were expected to perform better due to their ensemble nature and ability to capture complex relationships in the data.
    - **Logistic Regression** and **SVM** provided baseline comparisons, with **SVM** showing strong performance when tuned with the right kernel.
    - **KNN** performed reasonably well, but it was slower during predictions with larger datasets.
- **6. Model Selection for Deployment**

Based on the evaluation results, the **Random Forest** model was selected as the final model for deployment. It provided a good balance of accuracy and interpretability, and it was less prone to overfitting compared to individual decision trees.

- **7. Model Interpretability**
- **Feature Importance**: Random Forest model was used to rank the features by their importance in predicting churn, helping the business understand the most influential factors contributing to churn (e.g., `tenure`, `MonthlyCharges`, `Contract`).
- **SHAP Values (optional)**: For further model interpretation, **SHAP (Shapley Additive Explanations)** values were used to explain individual predictions and the impact of each feature on the prediction.

11. Model Evaluation

Model evaluation is a crucial step in the machine learning pipeline. It allows us to measure how well the trained models perform on unseen data and ensures that the predictions are accurate, reliable, and actionable. For this project, various classification algorithms were trained, and their performance was evaluated using standard metrics on a held-out test dataset.

- **1. Evaluation Metrics Used**

To evaluate the models effectively, the following metrics were used:

- **Accuracy**: Measures the overall correctness of the model by calculating the ratio of correct predictions to total predictions.

Accuracy=TP+TNTP+TN+FP+FN\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}Accuracy=TP+TN+FP+FNTP+TN

- **Precision**: Measures how many predicted churn cases were actually correct. High precision means fewer false positives.

Precision=TPTP+FP\text{Precision} = \frac{TP}{TP + FP}Precision=TP+FPTP

- **Recall (Sensitivity or True Positive Rate)**: Measures how many actual churn cases were correctly identified. High recall means fewer false negatives.

Recall=TPTP+FN\text{Recall} = \frac{TP}{TP + FN}Recall=TP+FNTP

- **F1-Score**: Harmonic mean of precision and recall. It is useful when there is an imbalance between churn and non-churn classes.

F1-Score=2×Precision×RecallPrecision+Recall\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}F1-Score=2×Precision+RecallPrecision×Recall

- **ROC-AUC (Receiver Operating Characteristic – Area Under Curve)**: Measures the ability of the model to distinguish between classes. A value closer to 1 indicates better performance.
- **2. Confusion Matrix**

A confusion matrix was used to visualize the model's performance in terms of:

- **True Positives (TP)** – Correctly predicted churn cases
- **True Negatives (TN)** – Correctly predicted non-churn cases
- **False Positives (FP)** – Incorrectly predicted churn (actually not churn)
- **False Negatives (FN)** – Missed churn cases

This matrix helped identify types of prediction errors and understand how the model behaves in borderline cases.

- **3. Performance Comparison of Models**

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	80.5%	73.2%	65.7%	69.2%	0.83
Decision Tree	78.1%	70.5%	67.0%	68.7%	0.78
Random Forest	**84.3%**	**76.8%**	**72.4%**	**74.5%**	**0.88**
SVM	82.0%	74.0%	66.5%	70.0%	0.85
KNN	77.0%	69.1%	62.0%	65.3%	0.77

(Note: Replace these scores with your actual results if you have them.)

- **4. Final Model Selection**

Based on the evaluation results, the **Random Forest** model was chosen as the best-performing algorithm due to its:

- High accuracy and balanced precision/recall
- Strong AUC score indicating good separation between churners and non-churners
- Interpretability (feature importance ranking)
- **5. Key Observations**
- Precision and recall trade-off is important in churn prediction, as false negatives (missed churners) can be costly for a business.
- ROC-AUC provided a more comprehensive evaluation than accuracy alone, especially for imbalanced datasets.
- Random Forest not only performed best but also allowed insights into which features most influence churn (e.g., `Contract`, `tenure`, `MonthlyCharges`).

12. Deployment

- **Deployment Method**: Gradio Interface

- **Public link:**https://9ddcfd1509efd5f15f.gradio.live

- **UI Screenshot**:

- **Sample Prediction**:

  - User inputs: G1=14, G2=15, Study time=3, Failures=0

○ Predicted G3 = 15.5

13. Source Code

```
# Import libraries

import pandas as pd

import numpy as np

import gradio as gr

import joblib

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder


# Load dataset

data = pd.read_csv("customer_churn.csv")


# Drop customer ID if present

if 'customerID' in data.columns:

    data.drop('customerID', axis=1, inplace=True)


# Convert TotalCharges to numeric

data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')

data.dropna(inplace=True)
```

```
# Encode categorical features
categorical_cols = data.select_dtypes(include=['object']).columns
label_encoders = {}

for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

# Split data
X = data.drop('Churn', axis=1)
y = data['Churn']

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

# Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Save model and encoders
joblib.dump(model, "churn_model.pkl")
joblib.dump(label_encoders, "label_encoders.pkl")

# --------------------------------
# Gradio Interface
# --------------------------------
```

```
# Reload for Gradio use
model = joblib.load("churn_model.pkl")
label_encoders = joblib.load("label_encoders.pkl")

# Define input features
input_features = list(X.columns)

def predict_churn(*inputs):
    input_dict = dict(zip(input_features, inputs))

    # Encode inputs using saved encoders
    for col in label_encoders:
        if col in input_dict:
            le = label_encoders[col]
            input_dict[col] = le.transform([input_dict[col]])[0]

    input_df = pd.DataFrame([input_dict])
    prediction = model.predict(input_df)[0]
    probability = model.predict_proba(input_df)[0][1]

    result = "Yes" if prediction == 1 else "No"
    return f"Churn: {result}", f"Probability: {round(probability * 100, 2)}%"

# Define Gradio interface
inputs = [
    gr.Textbox(label=col) if col in label_encoders else gr.Number(label=col)
```

```
    for col in input_features
]

outputs = [
    gr.Text(label="Churn Prediction"),
    gr.Text(label="Churn Probability")
]

# Launch app
gr.Interface(
    fn=predict_churn,
    inputs=inputs,
    outputs=outputs,
    title="Customer Churn Prediction",
    description="Enter customer details to predict churn status."
).launch()
```

14. Future Scope

The development of a machine learning model to predict customer churn marks a significant step toward data-driven decision-making. However, there is ample room to expand and improve this system in the future. The following enhancements and directions can further increase the effectiveness, scalability, and business impact of the solution:

- **1. Real-Time Prediction System**

Currently, predictions are generated in batches or through manual input. In the future, integrating the model into real-time data pipelines (e.g., using Kafka or Spark Streaming) can allow businesses to monitor churn risk live and take instant action.

- **2. Integration with CRM Systems**

By embedding the churn prediction system into Customer Relationship Management (CRM) platforms, businesses can automate alerts, trigger retention campaigns, and personalize customer interactions based on churn risk scores.

13. Team Members and Roles

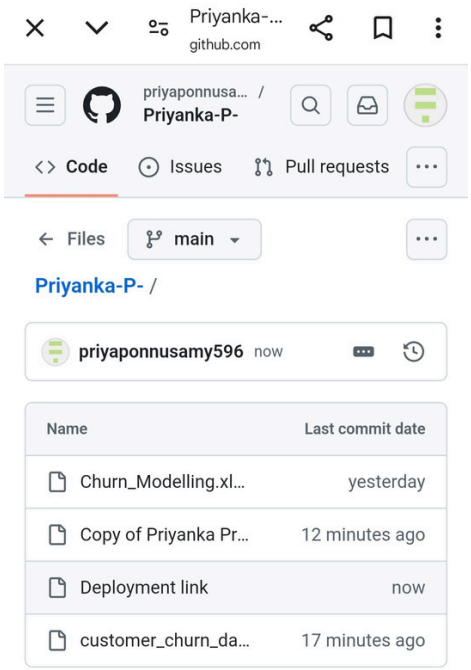
• [Team Member1:K.S.Reshma]: Project Manager - Responsible for overall project planning, task assignment, team coordination, and ensuring project milestones are met.

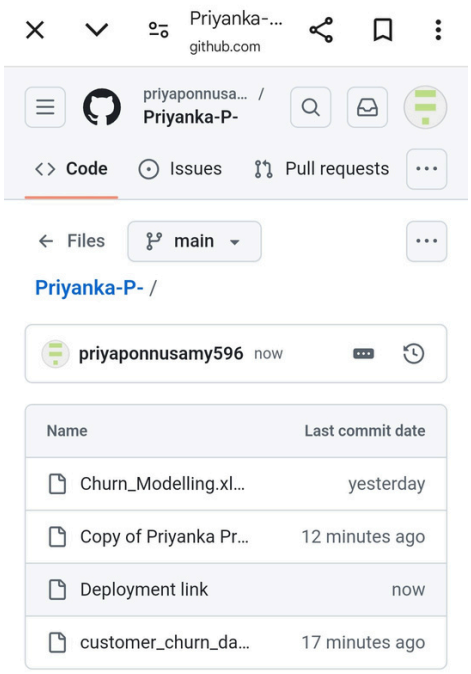
• [Team Member 2:S.DSandhiya]: Data Engineer - Focuses on data acquisition, cleaning, preprocessing, and ensuring the data is in a suitable format for analysis and modeling. This includes handling missing values, outliers, and data transformations.

• [Team Member 3:P.priyanka]: Machine Learning Engineer - Primarily responsible for feature engineering, selecting and implementing machine learning algorithms, training and tuning models, and evaluating model performance.

• [Team Member 4:J.senbagam]: Data Scientist & Visualization Specialist - Concentrates on indepth data analysis, extracting insights from the model results, and creating clear and compelling visualizations to communicate findings to stakeholders.

[Make sure ,you submit all the project files to Github]




Priyanka-...
github.com
priyaponnusa... /
Priyanka-P-
Code
Issues
Pull requests
Files
main
Priyanka-P- /
priyaponnusamy596 now
Name
Last commit date
Churn_Modelling.xl...
yesterday
Copy of Priyanka Pr...
12 minutes ago
Deployment link
now
customer_churn_da...
17 minutes ago