PHASE-3

Project Title: “Predicting customer churn using machine learning to uncover hidden patterns”.

Student Name: DHIVYA S

Register Number: 421323205058

Institution: KRISHNASAMY COLLEGE OF ENGINEERING AND TECHNOLOGY

Department: IT (INFORMATION TECHNOLOGY)

Date of Submission: 12-05-2025

Github Link: https://github.com/Rengoku0143/PROJECT2.git

1. Problem Statement

- Define customer churn clearly and explain why it's a critical problem for businesses (e.g., revenue loss, increased acquisition costs).
- Highlight the challenges in predicting churn, such as imbalanced datasets, identifying key drivers, and the dynamic nature of customer behavior.
- State the primary goal of the project: to develop a machine learning model that accurately predicts which customers are likely to churn and to identify the underlying patterns and factors contributing to this churn.
- Explain the potential business value of accurate churn prediction, such as enabling proactive retention strategies, improving customer satisfaction, and optimizing marketing efforts..

2. Abstract

- Provide a concise summary of the project.
- Mention the use of machine learning techniques to analyze customer data and predict churn.
- Briefly describe the data sources and the types of features considered (e.g., demographic, behavioral, transactional).
- Outline the methodology, including data preprocessing, model selection, training, and evaluation.
- Briefly state the expected outcomes, such as the performance of the churn prediction model and the key insights gained about churn drivers.
- Mention any deployment plans or tools used (e.g., creating a dashboard or API).

3. System Requirements

Hardware:

- Minimum RAM (e.g., 4 GB, 8 GB recommended)
- Processor specifications (e.g., Intel i5 or equivalent)
- Disk space requirements for data and models.

Software:

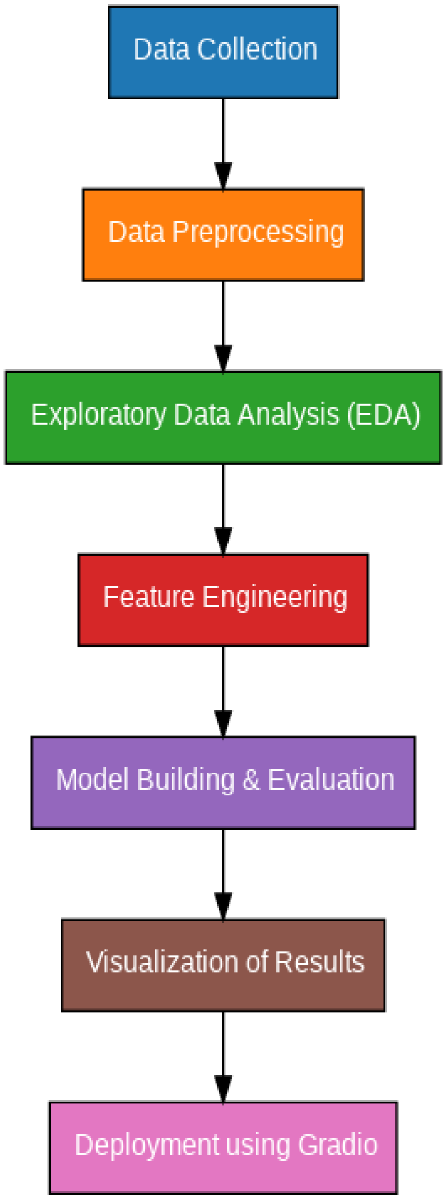
- Operating System (e.g., Windows, macOS, Linux)
- Python version (e.g., 3.8+)
- Key Python libraries: pandas, numpy, scikit-learn, matplotlib, seaborn, potentially others like XGBoost, LightGBM, TensorFlow, or PyTorch.
- IDE or environment (e.g., Jupyter Notebook, VS Code, Google Colab).

4. Objectives

- Develop a highly accurate and reliable machine learning model for predicting customer churn.
- Identify the most significant features and patterns that correlate with customer churn.
- Evaluate and compare the performance of various classification algorithms on the churn prediction task.
- Gain actionable insights into the reasons behind customer churn to inform business strategies and interventions.
- Potentially visualize the churn predictions and contributing factors for better understanding by stakeholders.
- If applicable, deploy the model in a user-friendly format or integrate it with existing business systems..

5. Flowchart of the Project Workflow

The overall project workflow for predicting customer churn was systematically structured into key stages. Data Acquisition gathers relevant customer data. Data Preprocessing cleans and encodes the data. Exploratory Data Analysis (EDA) uncovers patterns and trends. Feature Engineering creates meaningful input variables. Model Building selects and trains machine learning algorithms. Model Evaluation assesses performance. Deployment considers real-world application strategies. Testing and Interpretation validates the model's effectiveness and understands churn drivers



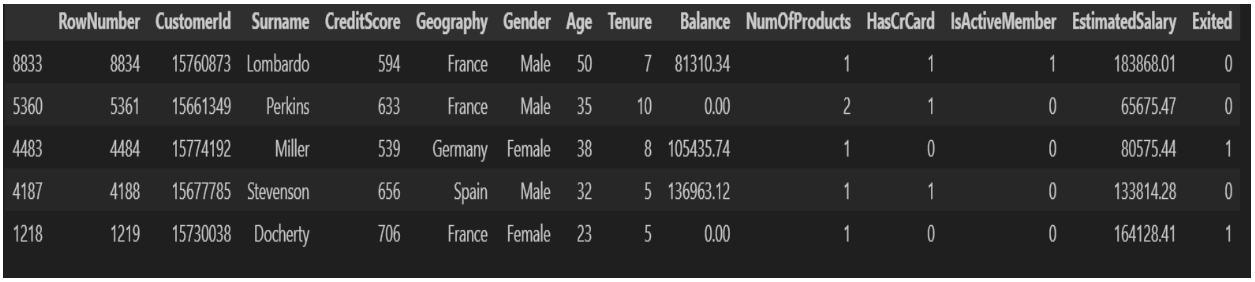
6. Dataset Description

- **Source:** Potentially a customer churn dataset from the UCI Machine Learning Repository (https://archive.ics.uci.edu/datasets).

- **Type:** Public dataset.

- **Size:** We expect the dataset to contain information on a **number of customers** (the exact count will depend on the specific dataset chosen) and have **various features** describing each customer.
- **Nature:** Structured tabular data.
- **Attributes:** We anticipate the dataset will include details such as:
    - **Demographics:** Age, gender, location, etc.
    - **Account Information:** Tenure, contract type, payment method, etc.
    - **Usage:** Service usage patterns, spending, etc.
    - **Interactions:** Support calls, complaints, etc.
    - **Churn Status:** Whether the customer has churned or not (our target variable).

Sample dataset (df.head())

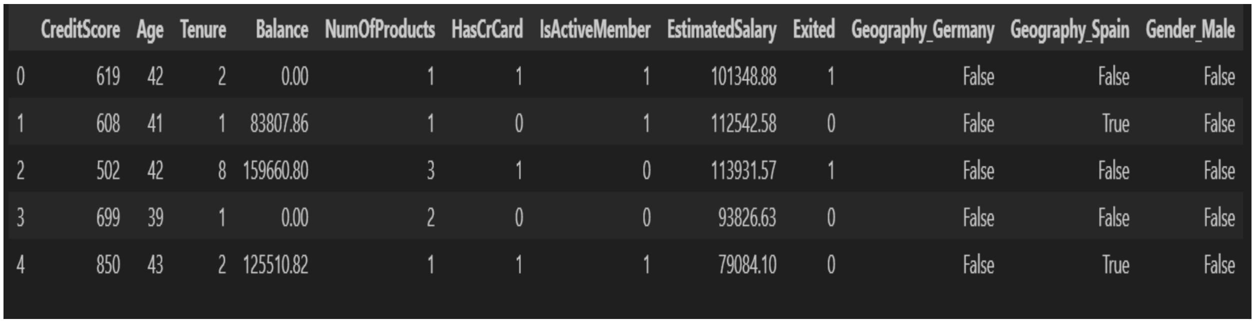


	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
8833	8834	15760873	Lombardo	594	France	Male	50	7	81310.34	1	1	1	183868.01	0
5360	5361	15661349	Perkins	633	France	Male	35	10	0.00	2	1	0	65675.47	0
4483	4484	15774192	Miller	539	Germany	Female	38	8	105435.74	1	0	0	80575.44	1
4187	4188	15677785	Stevenson	656	Spain	Male	32	5	136963.12	1	1	0	133814.28	0
1218	1219	15730038	Docherty	706	France	Female	23	5	0.00	1	0	0	164128.41	1

7. Data Preprocessing

- **Missing Values:** Explain the methods used to identify and handle missing data (e.g., imputation techniques, removal).
- **Duplicates:** Describe how duplicate records were identified and processed.

- **Outliers:** Discuss the approach to detect and treat outliers in relevant numerical features.
- **Encoding:** Detail the techniques used to convert categorical variables into numerical formats suitable for machine learning models (e.g., One-Hot Encoding, Label Encoding). Explain the rationale behind the chosen methods.
- **Handling Imbalanced Data:** If the churned vs. non-churned classes are significantly imbalanced, explain the techniques used to address this (e.g., oversampling, undersampling, SMOTE, using class weights in algorithms).
- **Scaling:** Describe if and how numerical features were scaled (e.g., StandardScaler, MinMaxScaler) and the reasons for scaling (e.g., sensitivity of some algorithms to feature scales).



	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germany	Geography_Spain	Gender_Male
0	619	42	2	0.00	1	1	1	101348.88	1	False	False	False
1	608	41	1	83807.86	1	0	1	112542.58	0	False	True	False
2	502	42	8	159660.80	3	1	0	113931.57	1	False	False	False
3	699	39	1	0.00	2	0	0	93826.63	0	False	False	False
4	850	43	2	125510.82	1	1	1	79084.10	0	False	True	False

8. Exploratory Data Analysis (EDA)

- **Univariate Analysis:**
    - Histograms to visualize the distribution of key features like tenure, monthly charges, and call volume.

- Boxplots to examine the distribution of numerical features across churned and non-churned customers (e.g., compare monthly charges for churned vs. non-churned).

➢ **Bivariate/Multivariate Analysis:**

Correlation heatmap:

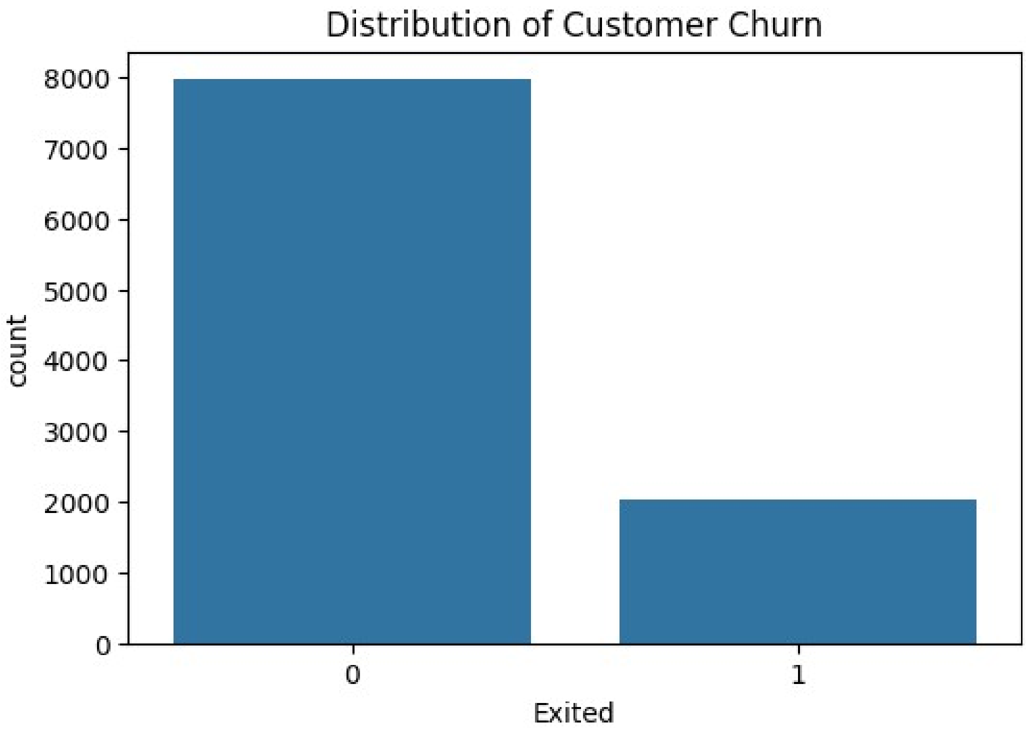
- Identify features that show strong positive or negative correlations with churn.

Visualizations (examples):

- Tenure vs. Churn: Explore if customers with shorter tenures are more likely to churn.
- Monthly Charges vs. Churn: Investigate the relationship between monthly charges and churn.
- Service Usage vs. Churn: Analyze how different service usage patterns relate to churn.

➢ **Key Insights:**

- Identify features that are strong indicators of churn.
- Summarize the relationships between different factors and churn probability.
- Note any interesting patterns or anomalies observed in the data.



9. Feature Engineering

➢ **New Features:**

- **Usage Ratio:** We can create a new feature that represents the ratio of a customer's current usage to their average usage over a longer period. This might help identify customers whose usage patterns are significantly changing, potentially indicating dissatisfaction or a switch to a competitor.
- **Interaction Frequency:** A feature representing the frequency of customer interactions (e.g., support calls, website visits) in a given period. A sudden increase or decrease in interaction frequency could be a sign of potential churn.
- **Tenure Group:** Instead of raw tenure, we can create categorical tenure groups (e.g., new customer, mid-tenure, long-term customer). This can help capture non-linear relationships between tenure and churn.
- **Spending Change:** Calculate the percentage change in a customer's monthly spending over a recent period. A significant drop in spending could be an early indicator of churn.

- **Number of Services/Products:** A count of the total number of services or products a customer subscribes to. Customers with fewer subscriptions might be easier to lose.
- **Days Since Last Interaction:** The number of days since the customer's last interaction with the company (e.g., support call, website login). Longer periods of inactivity could indicate disengagement.

➢ **Feature Selection:**

- We will identify and drop features that have extremely low variance, as these features provide little to no predictive information for the model.
- We will also analyze the correlation between different features and remove redundant features that are highly correlated with each other. This helps prevent multicollinearity, which can negatively impact the performance and interpretability of some models.

➢ **Impact:**

- The goal of feature engineering and selection is to improve the model's ability to accurately predict customer churn by reducing noise and highlighting the most relevant information.
- By creating new, potentially more informative features and selecting the most impactful ones, we aim to build a more robust and reliable churn prediction model that focuses on the factors most directly related to customers leaving.

10. Model Building

➢ Try **multiple models (baseline and advanced)**:

- Baseline Model: Logistic Regression
- Advanced Models:
  - Random Forest
  - Gradient Boosting Machine (e.g., XGBoost, LightGBM)
  - Support Vector Machine (SVM)

- **Explain why those models were chosen**:

- Logistic Regression: A good baseline for binary classification, easy to interpret.
- Random Forest: Handles non-linearity well, robust to overfitting.
- Gradient Boosting Machine: Often provides high accuracy, captures complex relationships.
- SVM: Effective in high-dimensional spaces, can handle non-linear boundaries with kernels.

11. Model Evaluation

Show evaluation metrics:

- Accuracy
- Precision
- Recall
- F1-score

ROC AUC

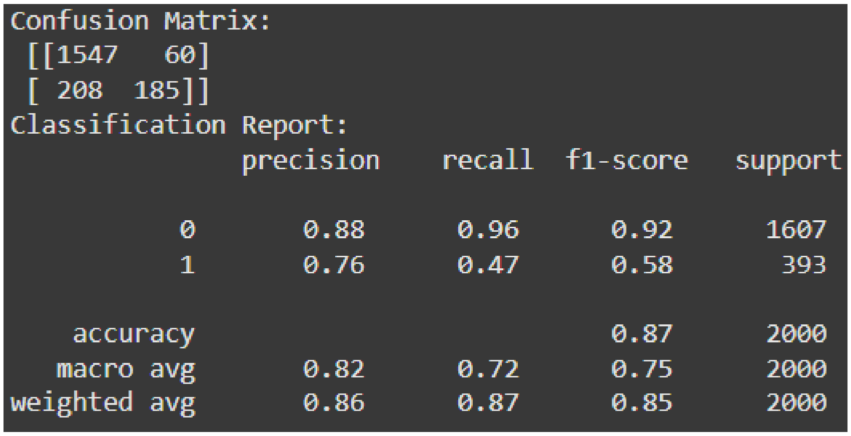
- (If applicable) RMSE (This is more for regression, but include if relevant to a part of your analysis)

Visuals:

- Confusion matrix (for each model)
- ROC curve (for each model, or a combined plot)

Error analysis or model comparison table:

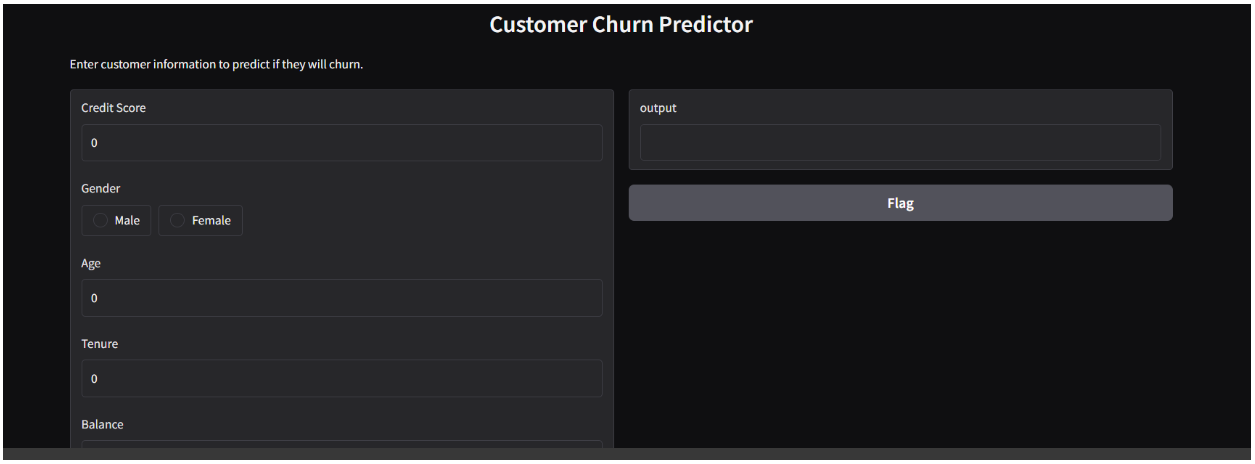
- Create a table comparing the metrics of all the models.
- Analyze where each model performs well or struggles (e.g., "Model X has high precision but lower recall").



```
Confusion Matrix:
 [[1547   60]
 [ 208  185]]
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.96      0.92      1607
           1       0.76      0.47      0.58       393

    accuracy                           0.87      2000
   macro avg       0.82      0.72      0.75      2000
weighted avg       0.86      0.87      0.85      2000
```

12. Deployment

- **Deployment Method**: Gradio Interface
- **Public Link**: https://d5e84948ec2a05a2ff.gradio.live/
- **UI Screenshot**:



Customer Churn Predictor

Enter customer information to predict if they will churn.

Credit Score: 0

Gender: ○ Male ○ Female

Age: 0

Tenure: 0

Balance

output

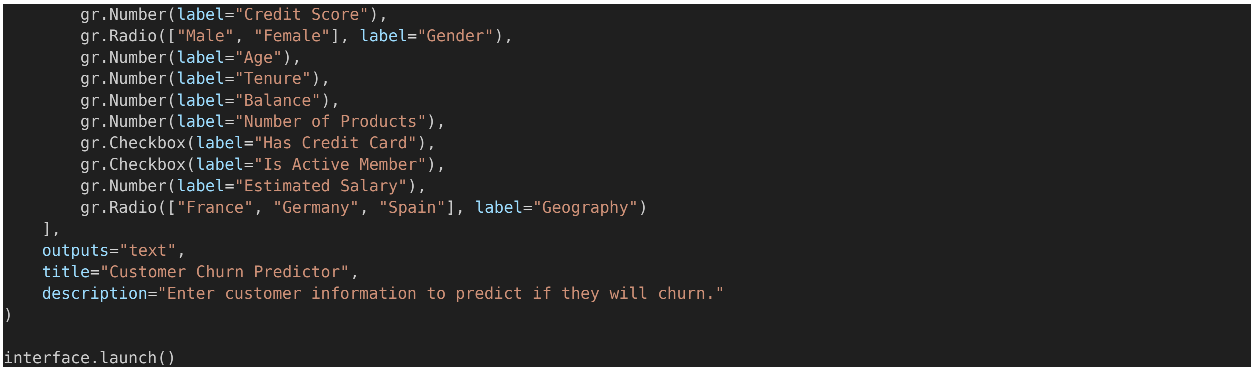
- **Sample Prediction**:

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

  Predicted G3 = 15.5

13. Source Code

```python
# Churn Prediction Data Preprocessing and Model Training
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import gradio as gr

# Load dataset
df = pd.read_csv('Churn_Modelling.csv')

# Drop irrelevant columns
df.drop(['RowNumber', 'CustomerId', 'Surname'], axis=1, inplace=True)

# Encode 'Gender' column
df['Gender'] = df['Gender'].map({'Female': 1, 'Male': 0})

# One-hot encode 'Geography' column
df = pd.get_dummies(df, columns=['Geography'], drop_first=True)

# Split into features and target
X = df.drop('Exited', axis=1)
y = df['Exited']

# Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

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

# Predict and evaluate
y_pred = model.predict(X_test)
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

# Define prediction function
def predict_churn(credit_score, gender, age, tenure, balance, num_products, has_cr_card, is_active, salary, geography):
    input_data = {
        'CreditScore': credit_score,
        'Gender': 1 if gender == 'Female' else 0,
        'Age': age,
        'Tenure': tenure,
        'Balance': balance,
        'NumOfProducts': num_products,
        'HasCrCard': has_cr_card,
        'IsActiveMember': is_active,
        'EstimatedSalary': salary,
        'Geography_Germany': 1 if geography == 'Germany' else 0,
        'Geography_Spain': 1 if geography == 'Spain' else 0
    }
    input_df = pd.DataFrame([input_data])
    for col in X.columns:
        if col not in input_df.columns:
            input_df[col] = 0
    input_df = input_df[X.columns]
    input_scaled = scaler.transform(input_df)
    prediction = model.predict(input_scaled)
    return "Churn" if prediction[0] == 1 else "No Churn"

# Define Gradio interface
interface = gr.Interface(
    fn=predict_churn,
    inputs=[
```



```
        gr.Number(label="Credit Score"),
        gr.Radio(["Male", "Female"], label="Gender"),
        gr.Number(label="Age"),
        gr.Number(label="Tenure"),
        gr.Number(label="Balance"),
        gr.Number(label="Number of Products"),
        gr.Checkbox(label="Has Credit Card"),
        gr.Checkbox(label="Is Active Member"),
        gr.Number(label="Estimated Salary"),
        gr.Radio(["France", "Germany", "Spain"], label="Geography")
    ],
    outputs="text",
    title="Customer Churn Predictor",
    description="Enter customer information to predict if they will churn."
)

interface.launch()
```

14. Future Scope

- "In the future, we aim to incorporate real-time data streaming to provide more dynamic churn predictions."
- "We plan to develop personalized retention strategies based on the churn prediction probabilities, integrating with a CRM system to automate targeted interventions."
- "Further research could explore the use of deep learning techniques, such as recurrent neural networks, to capture temporal patterns in customer behavior."

13. Team Members and Roles

- **DHIVYA S:**

  Acquires the dataset, handles missing values, removes duplicates, cleans and formats the data

- **SANDHIYA R:**

  Performs detailed EDA, creates visualizations, extracts key insights from data trends.

- **POOJA V:**

Creates new features, handles encoding/scaling, selects the most impactful features.

- **NANDHINI S:**

  Builds machine learning models, tunes hyperparameters, evaluates performance using various metrics.

- **DHANAVARSHINI B:**

  Deploys the model using Streamlit or Gradio, documents the full project, and manages the GitHub repo.