Github Link:https://github.com/vicky8925/predicting-customer-churn-using-machine-learning-to-uncover-hidden-patterns.git

Project Title: Predicting Customer Churn Using Machine Learning to Uncover Hidden Pattterns

1. Problem Statement:

- **Churners**: Customers who are likely to discontinue their relationship with the company.
- **Non-churners**: Customers who are likely to remain with the company.

Importance of the Problem

Addressing customer churn is vital for several reasons:

- **Revenue Retention**: Retaining existing customers is often more cost-effective than acquiring new ones.
- **Customer Lifetime Value**: Long-term customers contribute more to the company's profitability.
- **Competitive Advantage**: Understanding churn can provide insights into customer satisfaction and areas for improvement.

2. Project Objectives:

- **Data Collection and Preprocessing**
  - **Objective:** Gather comprehensive customer data, including demographics, usage patterns, transaction history, and customer service interactions.
  - **Actions:** Clean the data by handling missing values, outliers, and inconsistencies.
  - **Outcome:** A well-prepared dataset ready for analysis and modeling.
- **Feature Engineering**
  - **Objective:** Identify and create relevant features that can enhance the predictive power of the model.
  - **Actions:** Generate new variables, such as total charges or average monthly usage, and encode categorical variables appropriately.
  - **Outcome:** A set of features that effectively represent customer behavior and characteristics.Data AI Revolution
  -

- **Model Development and Training**
  - **Objective:** Build and train multiple machine learning models to predict customer churn.
  - **Actions:** Implement algorithms like Logistic Regression, Decision Trees, Random Forest, and Gradient Boosting.

- **Outcome:** A trained model capable of making predictions based on historical data.
-

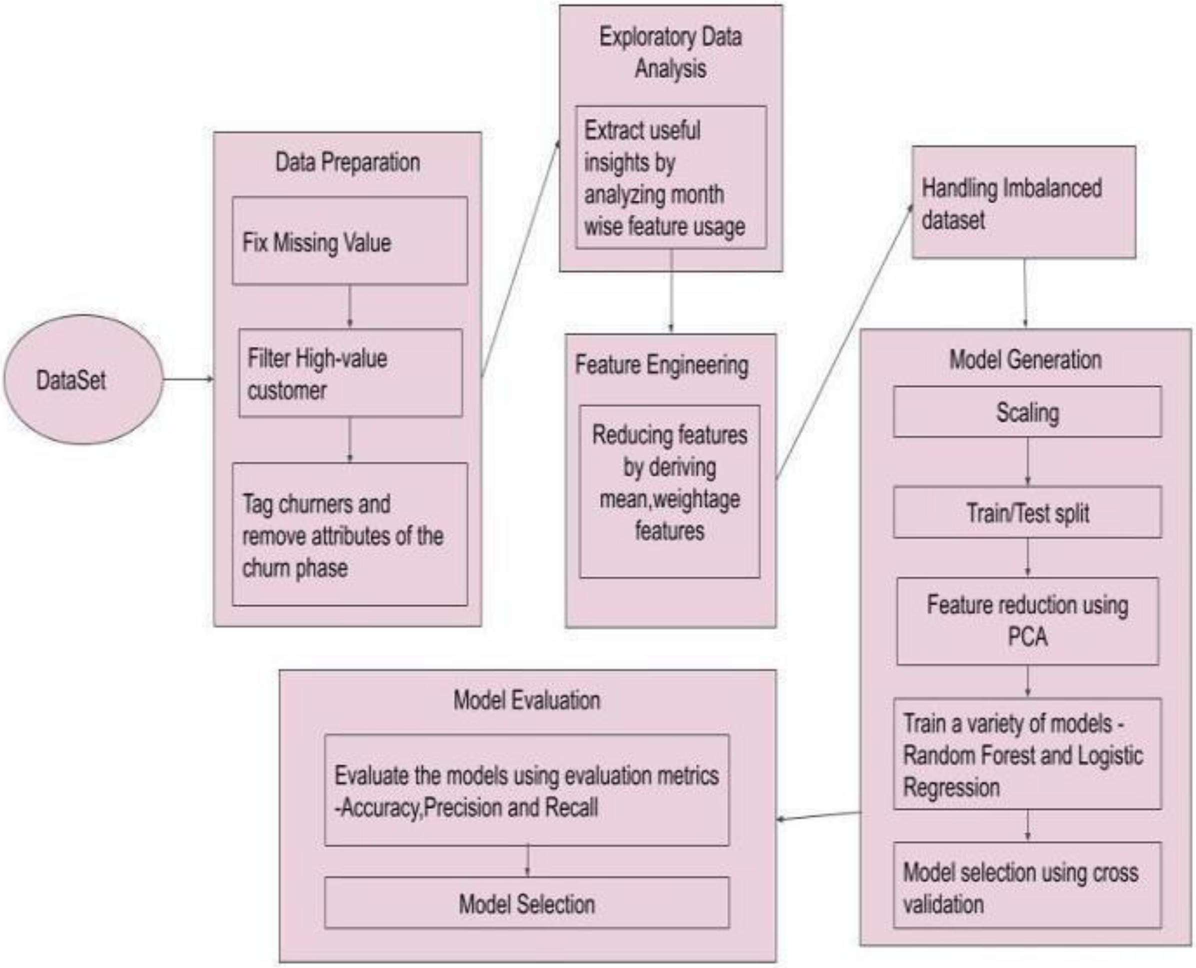
- **Model Evaluation**

  - **Objective:** Assess the performance of the developed models to ensure accuracy and reliability.
  - **Actions:** Use metrics such as accuracy, precision, recall, F1-score, and AUC-ROC curve to evaluate model performance.
  - **Outcome:** Identification of the best-performing model for deployment.ScienceDirect+4LeewayHertz - AI Development Company+4Coditude+4
- **Model Deployment and Monitoring**

  - **Objective:** Deploy the selected model into a production environment for real-time predictions.
  - **Actions:** Integrate the model with existing business systems and monitor its performance over time.
  - **Outcome:** A functional system that provides ongoing churn predictions to inform business decisions.
  -
- **Business Impact Analysis**
  - **Objective:** Evaluate the impact of churn prediction on business outcomes.
  - **Actions:** Analyze metrics such as customer retention rates, revenue growth, and customer lifetime value before and after implementing churn prediction strategies.
  - **Outcome:** Quantifiable evidence of the effectiveness of churn prediction in improving business performance. Coditude+1neptune.ai+1

By achieving these objectives, the project aims to provide businesses with actionable insights that can lead to improved customer retention, optimized resource allocation, and enhanced overall profitability.

3. Flowchart of the Project Workflow:


DataSet
Data Preparation
Fix Missing Value
Filter High-value customer
Tag churners and remove attributes of the churn phase
Exploratory Data Analysis
Extract useful insights by analyzing month wise feature usage
Feature Engineering
Reducing features by deriving mean,weightage features
Handling Imbalanced dataset
Model Generation
Scaling
Train/Test split
Feature reduction using PCA
Train a variety of models - Random Forest and Logistic Regression
Model selection using cross validation
Model Evaluation
Evaluate the models using evaluation metrics -Accuracy,Precision and Recall
Model Selection

4. Data Description:

Customer Demographics

- **CustomerID**: A unique identifier for each customer.
- **Gender**: The customer's gender (e.g., Male, Female).
- **Age**: The customer's age.
- **Tenure**: The number of months the customer has been with the company.
- **Geography**: The customer's location, which can indicate regional trends in churn.
- **CreditScore**: A numerical value representing the customer's creditworthiness.Enjoy Algorithms+1Home+1Home+1Enjoy Algorithms+1

Financial & Product Engagement

- **Balance**: The current balance in the customer's account.
- **NumOfProducts**: The total number of products or services the customer holds with the company.
- **HasCrCard**: Indicates whether the customer has a company-issued credit card.
- **IsActiveMember**: Indicates if the customer is currently an active member of the service.Home

5. Data Preprocessing:

1. Data Collection

- Gather data from various sources like CRM systems, transaction history, customer service logs, etc.
- Common features include: customer ID, demographic details, service usage, payment history, customer support interaction, and churn status (target variable).

2. Data Cleaning

- **Handle missing values**: Replace with mean/median/mode, drop rows/columns, or use imputation techniques.
- **Remove duplicates**: Check for and remove duplicate records to prevent bias.
- **Correct errors**: Fix inconsistencies like wrong data formats (e.g., date formats) or incorrect labels.

3. Feature Engineering

- **Create new features**: e.g., tenure groups, average usage per month, or interaction frequency.
- **Transform variables**: e.g., log transformation for skewed numerical data.

Encode temporal variables: Create features from dates such as "days since last interaction". Encoding Categorical Variables

- **Label encoding**: Converts categorical text data into numerical form (useful for ordinal variables).
- **One-hot encoding**: Creates binary columns for each category (good for nominal variables).
- **Target encoding**: Replaces category values with the mean of the target variable for that category.

5. Feature Scaling

- **Standardization** (Z-score normalization): For algorithms like SVM, KNN, or logistic regression.
- **Normalization** (Min-Max scaling): Especially useful when data is not normally distributed.

6. Handling Imbalanced Data

- **Resampling techniques**:
  - **Oversampling**: e.g., SMOTE (Synthetic Minority Over-sampling Technique)
  - **Undersampling**: Reducing the majority class.
- **Use of class weights**: For models that support it (e.g., logistic regression, random forest).

7. Train-Test Split

- Divide the data into training and test (or validation) sets to evaluate model performance fairly.
- Common ratios: 70:30, 80:20, or use **K-Fold Cross-Validation** for robust evaluation.

8. Dimensionality Reduction (Optional)

- Use **PCA** (Principal Component Analysis) or **feature selection techniques** to reduce noise and improve performance.

6. Exploratory Data Analysis (EDA):

. Understand the Dataset

- **Objective**: Predict whether a customer will churn (i.e., stop using a service).
- **Target Variable**: Usually a binary column like `Churn` (Yes/No or 1/0).
- **Features**: Could include customer demographics, account information, usage patterns, service types, etc.

2. Data Summary

- Use `.info()`, `.describe()` to get an overview:
  - Data types
  - Missing values
  - Statistical summary of numerical features
  - Unique values for categorical features

3. Missing Value Analysis

- Identify features with missing data.
- Strategy: Drop, impute (mean, median, mode), or flag them with indicators.

Univariate Analysis

- **Numerical features**:

  ```
  python
  CopyEdit
  df['MonthlyCharges'].hist(bins=20)
  ```

- **Categorical features**:

  ```
  python
  CopyEdit
  df['Contract'].value_counts().plot(kind='bar')
  ```

6. Bivariate Analysis

- **Churn vs Categorical Variables**:

  ```
  python
  CopyEdit
  pd.crosstab(df['Contract'], df['Churn'],
  normalize='index').plot(kind='bar', stacked=True)
  ```

- **Churn vs Numerical Variables**:

  ```
  python
  CopyEdit
  sns.boxplot(x='Churn', y='MonthlyCharges', data=df)
  ```

7. Correlation Analysis

- For numerical variables:

```
python
CopyEdit
sns.heatmap(df.corr(), annot=True)
```

7. Feature Engineering :

Understanding the Domain

Before feature engineering, it's essential to understand:

- What defines churn in your business (e.g., account closure, inactivity, subscription cancellation).
- The nature of your product/service and customer lifecycle.

2. Data Sources

Churn prediction models typically pull data from:

- **Customer demographics**
- **Transactional data**
- **Service usage logs**
- **Customer service interactions**
- **Subscription history**

3. Types of Features for Churn Prediction

A. Demographic Features

- Age, gender, income level, location

Type of account (individual vs. corporate) Feature Transformation Techniques

- **Binning**: Group continuous variables (e.g., age into age groups)
- **Scaling/Normalization**: Especially for distance-based models (e.g., k-NN)
- **One-hot encoding**: For categorical variables
- **Target encoding**: For high-cardinality categorical features

5. Feature Selection Techniques

- **Univariate statistics** (Chi-square, ANOVA)
- **Model-based** (feature importance from Random Forest, XGBoost)
- **Recursive Feature Elimination (RFE)**
- **Correlation analysis**: To remove multicollinearity

6. Time-Based Features (For Sequential Data)

- **Time since last purchase/login**
- **Recency, Frequency, and Monetary (RFM) analysis**
- **Seasonal patterns**: Usage changes during holidays, weekends, etc.

7. Advanced Feature Engineering

- **Natural Language Processing**: Sentiment analysis from reviews or support tickets
- **Clustering**: Assigning cluster labels (e.g., usage pattern groups)
- **Embedding techniques**: For high-dimensional categorical data

8. Automation Tools

- **Featuretools** (Python library for automated feature engineering)
- **tsfresh** (for time series data)
- **DataRobot, H2O.ai** (AutoML platforms)

Summary Table:

-

8. Model Building:

Understanding the Problem

- **Goal:** Predict whether a customer will churn (i.e., stop using a service).
- **Type:** Supervised classification problem (binary classification: churn vs no churn).

2. Data Collection and Preprocessing

Typical dataset includes features like:

- **Customer demographics** (age, gender, location)
- **Service usage** (subscription type, usage minutes, data consumption)
- **Customer support interaction** (number of complaints, support tickets)
- **Payment information** (billing method, payment history)

Preprocessing Steps:

- Handle missing values
- Encode categorical variables (One-Hot, Label Encoding)
- Feature scaling (StandardScaler, MinMaxScaler)
- Address class imbalance (e.g., using SMOTE, oversampling, or class weights)

3. Exploratory Data Analysis (EDA)

- Visualize churn rates across different features
- Correlation analysis
- Identify important variables

4. Model Selection

Start with baseline models and move to more complex ones:

- **Logistic Regression** (baseline)

- **Decision Trees**
- **Random Forest**
- **Gradient Boosting (e.g., XGBoost, LightGBM)**
- **Support Vector Machines**
- **Neural Networks (if dataset is large)**

5. Model Training & Evaluation

Split data into:

- **Training Set**
- **Validation Set**
- **Test Set**

Use metrics like:

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

Apply **cross-validation** to reduce overfitting and assess generalizability.

6. Feature Importance & Model Interpretation

- Use model explainability tools (e.g., SHAP, LIME)
- Identify key drivers of churn
- Align insights with business actions

7. Deployment

- Save model using `joblib` or `pickle`
- Deploy with APIs (e.g., Flask, FastAPI)
- Monitor model performance over time

8. Monitoring and Updating

- Track churn prediction performance (via dashboards)
- Re-train model periodically with new data

9. Visualization of Results & Model Insights:

When working on a **Customer Churn Prediction** project using machine learning, the **visualization of results and model insights** is essential to:

- Understand how the model is performing.
- Gain business insights from the data and model.
- Communicate findings effectively to stakeholders.

Here's a breakdown of how to visualize results and model insights in this context:

1. Confusion Matrix

- Shows **true positives**, **true negatives**, **false positives**, and **false negatives**.
- Helps assess where the model is making errors in predicting churn vs. non-churn.

2. ROC Curve & AUC Score

- **ROC Curve** shows the trade-off between true positive rate and false positive rate.
- **AUC (Area Under Curve)** quantifies the overall ability of the model to discriminate between classes.

3. Precision-Recall Curve

- Especially helpful in **imbalanced datasets** (where churners are a minority).
- Focuses on how well the model identifies actual churners.

4. Feature Importance (Model Explainability)

- Visualize which features (e.g., tenure, usage, customer service calls) are most important for predicting churn
- SHAP (SHapley Additive exPlanations):
  - Local and global interpretability.
  - Shows individual prediction reasoning.

Churn Rate by Feature

- Bar plots of churn rate across customer segments (e.g., by contract type, monthly charges).
- Helps uncover **behavioral patterns** in churn.

6. Model Performance Metrics

- **Accuracy, Precision, Recall, F1-Score** – often summarized in a table or bar chart.
- Helpful for comparing multiple models.

7. Customer Segmentation (Clustering for Insights)

- Use t-SNE or PCA to reduce dimensions and visualize customer clusters.
- Label clusters by churn rates to discover high-risk groups.

10. Tools and Technologies Used:

Data Collection & Storage

- **Databases:** MySQL, PostgreSQL, MongoDB
- **Data Lakes / Warehouses:** Amazon S3, Google BigQuery, Snowflake
- **APIs:** REST APIs for fetching customer interaction logs, payment history, etc.

2. Data Preprocessing & Analysis

- **Python Libraries:**
  - `pandas` – data manipulation
  - `numpy` – numerical operations
  - `scikit-learn` – preprocessing tools (e.g., encoding, scaling)
  - `missingno` – missing data visualization
- **Notebook Environments:**
-
  - Jupyter Notebook / JupyterLab
  - Google Colab

3. Data Visualization & EDA

- `matplotlib`, `seaborn` – static visualizations
- `plotly`, `bokeh` – interactive visualizations
- `Tableau`, `Power BI` – business-friendly dashboards

4. Machine Learning Models

- **Libraries:**
  - `scikit-learn` – Logistic Regression, Decision Trees, Random Forests
  - `XGBoost`, `LightGBM` – advanced gradient boosting
  - `TensorFlow`, `Keras`, `PyTorch` – neural networks for deep learning
- **Model Selection & Tuning:**
  - `GridSearchCV`, `RandomizedSearchCV`
  - `Optuna`, `Hyperopt` for hyperparameter tuning

11. Team Members and Contributions :

K.ABIKA :[***Data cleaning, EDA]***

T.R.VIGNESH :[***Feature engineering]***

S.KIRUBANANDHAM :[***Model development]***

V.SIVA VETRIVEL:[***Documentation and reporting]***