**Customer Churn Prediction: Designing an Innovative Approach**

**Introduction**

In the rapidly evolving business landscape, understanding and predicting customer churn is essential for maintaining a competitive edge. Customer churn refers to the phenomenon where customers cease their relationship with a company. Predicting customer churn can significantly impact business strategies, allowing for proactive measures to retain customers. This document outlines the data collection and innovative design strategies for predicting customer churn. High customer churn rates can result in substantial financial losses, making it crucial for businesses to identify potential churners and implement retention strategies. The project aims to explore innovative ways to improve customer churn prediction, ultimately helping businesses reduce churn and enhance customer satisfaction.

**Objectives**

1. **Data Collection and Preparation:** Collect relevant customer data and prepare it for analysis. This may include demographic information, purchase history, customer service interactions, and any other pertinent data.
2. **Traditional Churn Prediction Models:** Implement and evaluate traditional customer churn prediction models, such as logistic regression, decision trees, and random forests, to establish a baseline for performance.
3. **Innovative Design:** Propose and develop innovative features or models that can enhance the accuracy and effectiveness of churn prediction. This may involve incorporating external data sources, exploring novel data preprocessing techniques, or experimenting with state-of-the-art machine learning algorithms.
4. **Model Evaluation:** Assess the performance of both traditional and innovative churn prediction models using appropriate metrics like accuracy, precision, recall, F1-score, and ROC AUC. Compare the results to determine the impact of the innovative design.
5. **Recommendation System:** Develop a recommendation system to suggest personalized strategies for customer retention based on the predictions. This could involve targeted marketing campaigns, loyalty programs, or other customer engagement initiatives.
6. **User-Friendly Interface:** Create a user-friendly interface or dashboard to enable businesses to interact with the churn prediction and recommendation system easily.

**Section 1: Data Collection**

**1.1 Data Sources:**

Identify diverse sources for collecting customer data, including but not limited to:

* Customer demographics and profile information
* Purchase history
* Usage patterns
* Customer service interactions
* Feedback and reviews
* Social media engagement
* Website interactions

**1.2 Data Preprocessing:**

* Clean the collected data by handling missing values, outliers, and inconsistencies.
* Perform feature engineering to extract meaningful insights from the raw data.
* Conduct Exploratory Data Analysis (EDA) to gain a comprehensive understanding of the data.

**1.3 Data Integration and Storage:**

* Integrate data from various sources to create a comprehensive dataset.
* Store the integrated data in a structured and easily accessible format for model development.

**Section 2: Designing an Innovative Approach**

**2.1 Advanced Machine Learning Models:**

Utilize advanced machine learning models to predict customer churn, such as:

* Random Forest
* Gradient Boosting Machines (GBM)
* Neural Networks
* Support Vector Machines (SVM)
* XGBoost

**2.2 Ensemble Learning:**

Combine multiple models using ensemble techniques like:

* Stacking
* Bagging
* Boosting

**2.3 Hyperparameter Optimization:**

Optimize model performance through hyperparameter tuning using techniques like:

* Grid Search
* Random Search
* Bayesian Optimization

**2.4 Feature Selection:**

Employ feature selection methods to identify the most relevant features for prediction, such as:

* Recursive Feature Elimination (RFE)
* LASSO regression
* Feature Importance from models

**2.5 Explainable AI:**

Integrate explainable AI techniques to interpret model predictions and provide insights into the factors influencing churn prediction, enhancing model transparency and trustworthiness.

**2.6 Database:**

**Link:**[**https://www.kaggle.com/datasets/blastchar/telco-customer-churn**](https://www.kaggle.com/datasets/blastchar/telco-customer-churn)

**2.7 Program:**

import numpy as np

import pandas as pd

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

df = pd.read\_csv('/content/WA\_Fn-UseC\_-Telco-Customer-Churn.csv')

df.head()

df.columns = df.columns.str.lower().str.replace(' ', '\_')

categorical\_columns = list(df.dtypes[df.dtypes == 'object'].index)

for c in categorical\_columns:

df[c] = df[c].str.lower().str.replace(' ', '\_')

df.head().T

df.totalcharges

df.totalcharges = pd.to\_numeric(df.totalcharges, errors='coerce')

df.totalcharges = df.totalcharges.fillna(0)

df.churn = (df.churn == 'yes').astype(int)

df['churn']

from sklearn.model\_selection import train\_test\_split

df\_full\_train, df\_test = train\_test\_split(df, test\_size=0.2, random\_state=1)

df\_train, df\_val = train\_test\_split(df\_full\_train, test\_size=0.25, random\_state=1)

df\_test.head()

len(df\_train), len(df\_val), len(df\_test)

df\_train = df\_train.reset\_index(drop=True)

df\_val = df\_val.reset\_index(drop=True)

df\_test = df\_test.reset\_index(drop=True)

y\_train = df\_train.churn.values

y\_val = df\_val.churn.values

y\_test = df\_test.churn.values

del df\_train['churn']

del df\_val['churn']

del df\_test['churn']

df\_full\_train = df\_full\_train.reset\_index(drop=True)

df\_full\_train.isnull().sum()

df\_full\_train.churn.value\_counts(normalize=True)

df\_full\_train.churn.mean()

numerical = ['tenure', 'monthlycharges', 'totalcharges']

categorical = [

'gender',

'seniorcitizen',

'partner',

'dependents',

'phoneservice',

'multiplelines',

'internetservice',

'onlinesecurity',

'onlinebackup',

'deviceprotection',

'techsupport',

'streamingtv',

'streamingmovies',

'contract',

'paperlessbilling',

'paymentmethod',

]

df\_full\_train[categorical].nunique()

df\_full\_train.head()

churn\_male = df\_full\_train[df\_full\_train.gender == 'male'].churn.mean()

churn\_male

churn\_female = df\_full\_train[df\_full\_train.gender == 'female'].churn.mean()

churn\_female

global\_churn = df\_full\_train.churn.mean()

global\_churn

df\_full\_train.partner.value\_counts()

churn\_partner = df\_full\_train[df\_full\_train.partner == 'yes'].churn.mean()

churn\_partner

churn\_no\_partner = df\_full\_train[df\_full\_train.partner == 'no'].churn.mean()

churn\_no\_partner

churn\_no\_partner / global\_churn

churn\_partner / global\_churn

SELECT gender, AVG(churn), AVG(churn) - global\_churn AS diff, AVG(churn) / global\_churn AS risk FROM data GROUP BY gender;

from IPython.display import display

for c in categorical:

print(c)

df\_group = df\_full\_train.groupby(c).churn.agg(['mean', 'count'])

df\_group['diff'] = df\_group['mean'] - global\_churn

df\_group['risk'] = df\_group['mean'] / global\_churn

display(df\_group)

from sklearn.metrics import mutual\_info\_score

mutual\_info\_score(df\_full\_train.churn, df\_full\_train.contract)

mutual\_info\_score(df\_full\_train.gender, df\_full\_train.churn)

mutual\_info\_score(df\_full\_train.contract, df\_full\_train.churn)

mutual\_info\_score(df\_full\_train.partner, df\_full\_train.churn)

def mutual\_info\_churn\_score(series):

return mutual\_info\_score(series, df\_full\_train.churn)

mi = df\_full\_train[categorical].apply(mutual\_info\_churn\_score)

mi.sort\_values(ascending=False)

df\_full\_train[numerical].corrwith(df\_full\_train.churn)

df\_full\_train[df\_full\_train.tenure >= 2].churn.mean()

df\_full\_train[(df\_full\_train.tenure > 2) & (df\_full\_train.tenure <= 12)].churn.mean()

df\_full\_train[df\_full\_train.tenure > 12].churn.mean()

from sklearn.feature\_extraction import DictVectorizer

df\_train[['gender','contract','tenure']].iloc[:10].to\_dict(orient='records')

dicts = df\_train[['gender','contract','tenure']].iloc[:100].to\_dict(orient='records')

dv = DictVectorizer(sparse=False)

dv.fit(dicts)

dv.get\_feature\_names\_out()

dv.transform(dicts)

dv = DictVectorizer(sparse=False)

train\_dict = df\_train[categorical + numerical].to\_dict(orient='records')

print(type(train\_dict))

X\_train = dv.fit\_transform(train\_dict)

X\_train

dv.get\_feature\_names\_out()

val\_dict = df\_val[categorical + numerical].to\_dict(orient='records')

X\_val = dv.transform(val\_dict)

X\_val

def sigmoid(z):

return 1 / (1 + np.exp(-z))

z = np.linspace(-7, 7, 51)

z

sigmoid(10000)

import matplotlib.pyplot as plt

plt.plot(z, sigmoid(z))

def linear\_regression(xi):

result = w0

for j in range(len(w)):

result = result + xi[j] \* w[j]

return result

def logistic\_regression(xi):

score = w0

for j in range(len(w)):

score = score + xi[j] \* w[j]

result = sigmoid(score)

return result

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression(solver='lbfgs')

model.fit(X\_train, y\_train)

model.intercept\_[0]

model.coef\_[0]

model.predict\_proba(X\_val)

**Google collab link:** <https://colab.research.google.com/drive/1tWo2W3CUZZ0mTTDAd-Wf9fusoWng-e31?usp=sharing>

**Conclusion**

Designing an innovative approach to customer churn prediction involves a strategic blend of advanced machine learning techniques, efficient data collection, preprocessing, model optimization, and continuous monitoring. By employing these methods, businesses can enhance customer retention strategies and drive sustainable growth.