kenny067 /
Churn_Customers

Code Issues Pull requests Actions Projects Wiki Security Insights Setting

0 stars 0 forks 1 watching Branches Activity
Tags

Public repository

1 Branch 0 Tags Go to file t Go to file Add file Code

kenny067 dashboard 8e47566 · 3 minutes ago

images	adding images	1 hour ago
README.md	Updated README	1 hour ago
churn presentation.pdf	commiting pptx	3 hours ago
churn_customer.ipynb	Update churn_customer.ipynb	1 hour ago
churn_dataset.csv	commiting the notebook and the d...	yesterday
dashboard.twbx	dashboard	3 minutes ago
~dashboard__3768.twbr	dashboard	3 minutes ago

CHURN_CUSTOMERS

BUSINESS UNDERSTANDING

Syria Tel, a telecommunication company, wants to predict customer churn. Identifying customers who are likely to stop using their services soon. Churn represents a major financial challenge, as acquiring new customers is often more expensive than retaining existing ones. By analyzing customer data, Syria Tel can devolop targeted strategies to improve customer retention and reduce revenue loss

Project Overview

In this project we aim to build a classifier to predict whether a customer will ('soon') stop doing business with Syria Tel, a Telecommunications company. This project analyzes customer churn for Syria Tel, a telecommunications company. The goal is to identify patterns and factors contributing to customer churn and develop predictive models to assist in proactive customer retention strategies.

Objective

- Are churned customers more likely to have high or low usage
- The corelation between churn and other variables

DATA UNDERSTANDING

The dataset has 21 variables with a record of 3,333 records

Key variables include customer service calls, international plan subscription, total usage, and tenure.

The target variable is churn, indicating whether a customer has left the service.

Import the libraries

- **Data manipulation** : pandas, numpy
- **Visualization** : matplotlib, seaborn
- **Machine learning** : sklearn for Decision Tree, evaluation, and preprocessing
- **Handling Imbalanced Data** : imblearn.SMOTE for oversampling minority classes

```
    # Import libraries
    import pandas as pd
    import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import  LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import precision_recall_curve
from imblearn.over_sampling import SMOTE
```

Load the dataset

- The dataset is loaded using pd.read_csv().
- It is inspected using .head(), .tail(), .dtypes(), .columns(), .info(), .describe() to understand the data structure.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
#   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
0   state                   3333 non-null   object
1   account length          3333 non-null   int64
2   area code               3333 non-null   int64
3   phone number            3333 non-null   object
4   international plan      3333 non-null   object
5   voice mail plan         3333 non-null   object
6   number vmail messages   3333 non-null   int64
7   total day minutes       3333 non-null   float64
8   total day calls         3333 non-null   int64
9   total day charge        3333 non-null   float64
10  total eve minutes       3333 non-null   float64
11  total eve calls         3333 non-null   int64
12  total eve charge        3333 non-null   float64
13  total night minutes     3333 non-null   float64
14  total night calls       3333 non-null   int64
15  total night charge      3333 non-null   float64
16  total intl minutes      3333 non-null   float64
17  total intl calls        3333 non-null   int64
18  total intl charge       3333 non-null   float64
19  customer service calls  3333 non-null   int64
20  churn                   3333 non-null   bool
dtypes: bool(1), float64(8), int64(8), object(4)
memory usage: 524.2+ KB **
```

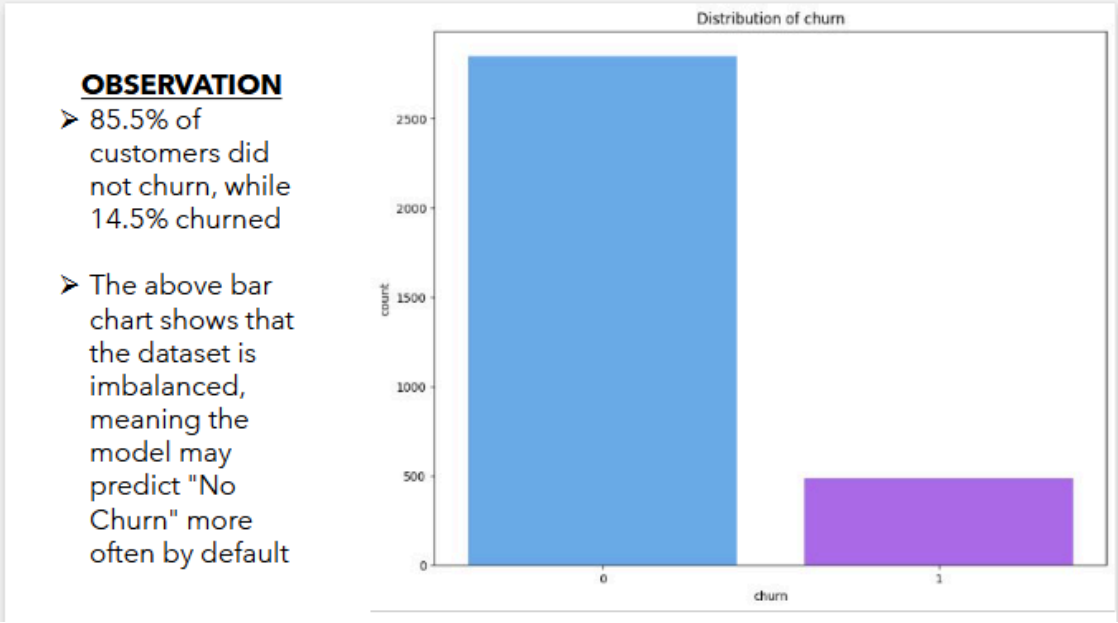
DATA PREPARATION

Data Cleaning

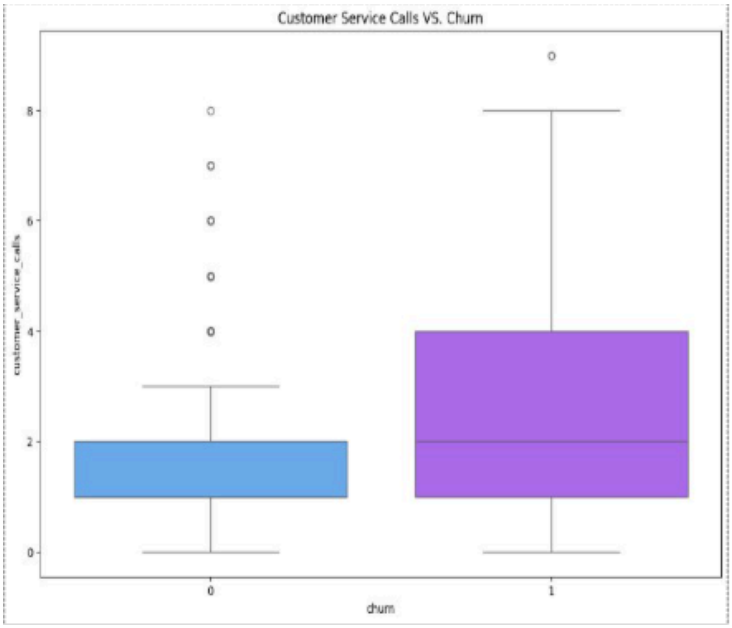
The dataset is clean with 0 missing values The column names were separated by space, so we had to replace the spaces with underscore

EDA(Explaratory Data Analysis)

- Churned Distribution : Visualized the proppotion of churned verses the non churned customer



- Customer Service Calls & Churn : Found that higher customer service interactions correlate with higher churn rates.



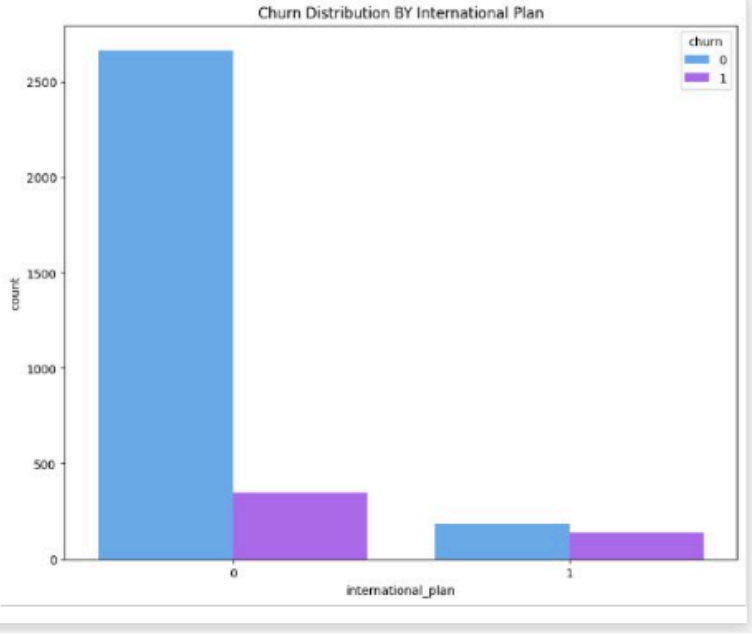
OBSERVATION

- A high number of customer service calls might indicate unresolved issues or dissatisfaction.
- The median number of customer service calls is higher for churn = 1, suggests that customers who call the customer service frequently are more likely to churn.

- International Plan & Churn : Customers with international plans showed varying churn tendancies

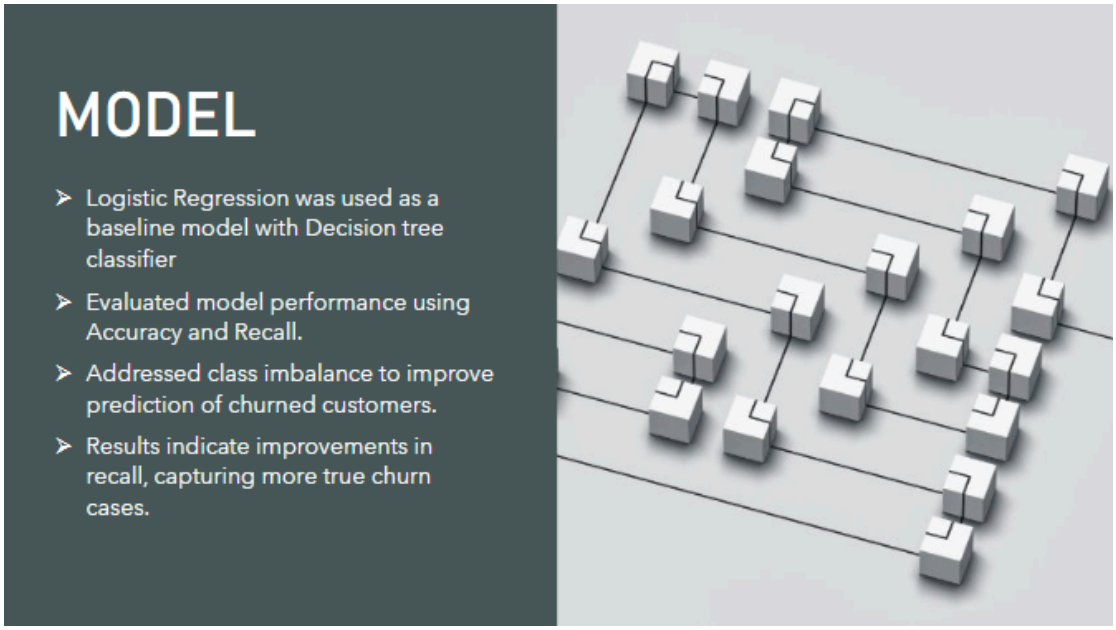
OBSERVATION

- The chart shows that customers with an international plan (international plan = 1) have a relatively higher proportion of churned users compared to those without an international plan (international plan = 0)



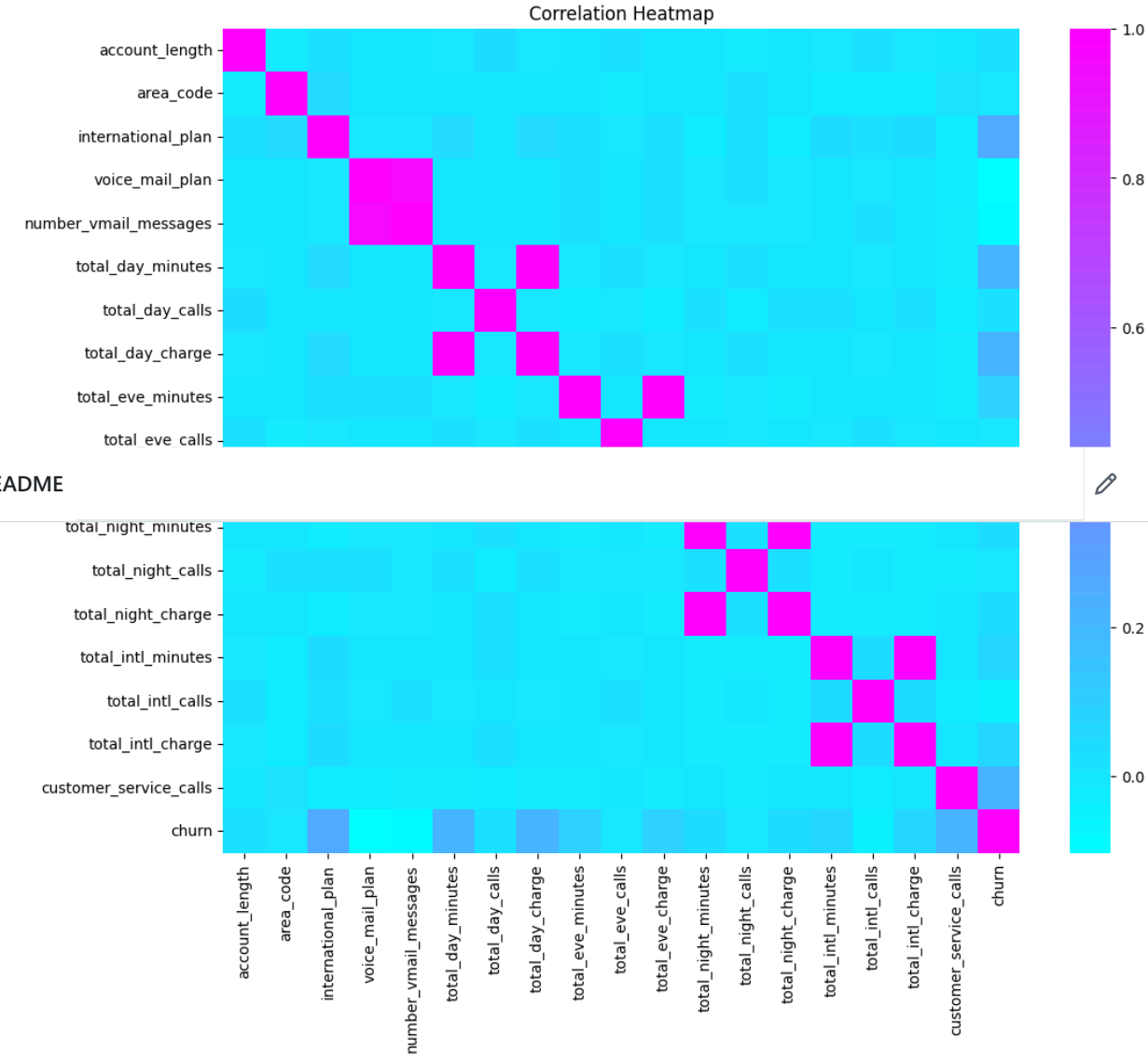
- Handled Class Imbalanced using SMOTE to ensure balanced model learning.
-

MODEL DEVELOPMENT



** Baseline Model** : Logistic Regression was the baseline model followed by the Decision Tree Classifier

- Check the correlation of the variable to see which variables has a higher correlation



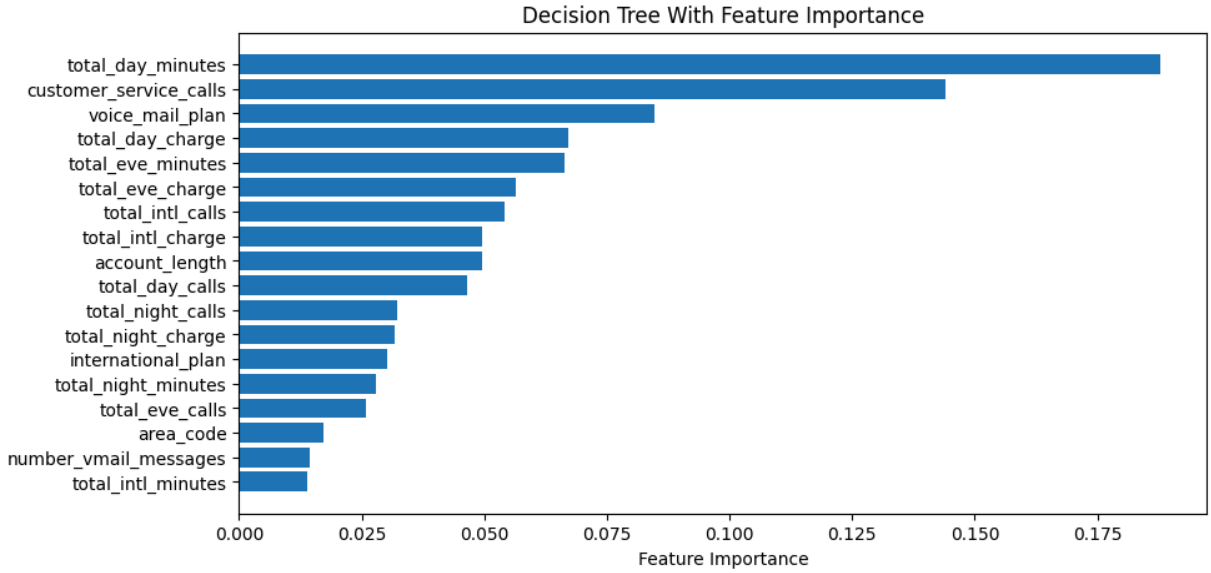
README

Logistic Regression

- Started by splitting the data into 80% traing and 20% testing
- Fixed the class imbalance using SMOTE and we able to attain a class balance of 2280
- Feature scalling using the ** standard Scaller** to improve the numerical stability and model perfomance
- Trained the baseline LR model
- Evaluated the perfomance of the baseline model where the model provides a better perfomance across both the churn class 1 and the non-churned class

Decision Tree Classifier

- Checked the feature importance to understand which variables contributes the most to my model predictions.



- Evaluated the DT model accuracy and found that it had 100% Training accuracy and 82% Test accuracy, the gap indicates overfitting
- We needed to reduce overfitting and we ended up Training a Pruned DT and found that the Training accuracy had reduce to 87% and the Test accuracy to increase to 85%.
- This is gud since there isn't overfitting amd our model

The model is now detecting churn well(72% recall for class 1 and 88% recall for class 0) while keeping the overall accuracy of 86%.

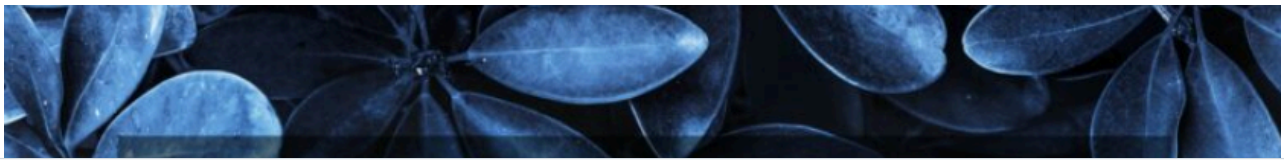
It is a good balance between detecting churners and avoiding too many false alarms.

Files in This Repository

- churn_customer.ipyn : Jupyter Notebook containg the data analysis and model training.
- churn presentation.pdf : Presentation summarizing the analysis with non-technical slides
- churn_dashboard : A tableau dashboard showing visual of the analysis

AUTHOR

- KENNEDY KARIUKI
- LINKEDIN :https://www.linkedin.com/in/kennedy067/
- GITHUB : https://github.com/kenny067/Churn_Customers/edit/main/README.md



Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%