

**Faculty of Engineering & Technology**

**Department of Computer Science and Applications**

**2023-24**

**Mini Project Report**

**on**

**“Customer Churn Prediction”**

**By**

**Pranay Varade,**

**Srivatsa Puranik,**

**Omkar Patne.**

**PRN**

**1132220534,**

**1132220542,**

**1132220552.**

**M. Sc. (Computer Science)**

**DR. VISHWANATH KARAD MIT WORLD PEACE UNIVERSITY, PUNE**  


**Faculty of Engineering & Technology**

**Department of Computer Science and Applications**

Certificate

This is to certify that**,** Mr. **Pranay Varade** of SYMSc. Computer Science Semester III has successfully completed mini project entitled “**Customer Churn Prediction**” in the subject “Lab on ML using Python & Mini Project” for the academic year 2023-24.

Prof. Navnath Shete

Project Guide

Date: \_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_

Examiners:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**Faculty of Engineering & Technology**

**Department of Computer Science and Applications**

Certificate

This is to certify that**,** Mr. **Srivatsa Puranik** of SYMSc. Computer Science Semester III has successfully completed mini project entitled “**Customer Churn Prediction**” in the subject “Lab on ML using Python & Mini Project” for the academic year 2023-24.

Prof. Navnath Shete

Project Guide

Date: \_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_

Examiners:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**Faculty of Engineering & Technology**

**Department of Computer Science and Applications**

Certificate

This is to certify that**,** Mr. **Omkar Patne** of SYMSc. Computer Science Semester III has successfully completed mini project entitled “**Customer Churn Prediction**” in the subject “Lab on ML using Python & Mini Project” for the academic year 2023-24.

Prof. Navnath Shete

Project Guide

Date: \_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_

Examiners:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Content** | **Page No.** |
| 1. | Front Page | 1 |
| 2. | Certificate | 2 |
| 3. | Abstract | 6 |
| 4. | Introduction | 7 |
| 5. | Literature Survey | 8 |
| 6. | Problem Statement | 11 |
| 7. | Methodology | 12 |
| 8. | Architecture | 13 |
| 9. | Graphs and Tables | 15 |
| 10. | Model Building | 20 |
| 11. | Accuracy Interpretation | 29 |
| 12. | Result, Discussion and Suggestions | 33 |
| 13. | Conclusion | 35 |
| 14. | Limitations and Future Scope | 36 |

**Abstract**

In the fiercely competitive telecommunications industry, customer retention stands as a paramount challenge. This project addresses this imperative by harnessing the power of machine learning to predict customer churn based on historical data. Through the exploration of usage patterns, demographic insights, and service-related information, a predictive model is meticulously crafted. The model undergoes comprehensive training and evaluation, employing key metrics such as accuracy and precision.

The chosen algorithms, including Logistic Regression, Support Vector Machine, Decision Tree, and Random Forest. Their adaptability to binary classification tasks and proven efficacy in churn prediction make them fitting candidates. Logistic Regression excels at binary classification, while Support Vector Machine's flexibility accommodates complex decision boundaries. Decision Tree and Random Forest bring ensemble learning to the forefront, enhancing predictive accuracy. By delving into the intricacies of each algorithm, this study aims to unveil their specific strengths in the context of customer churn, providing a foundation for more informed choices in predictive modeling for telecom operators. These algorithms undergo rigorous training and evaluation, utilizing essential metrics like accuracy and precision.

The model identifies patterns that signify potential churn, serving as a proactive tool for telecom operators. The findings offer nuanced insights into customer behavior, enabling operators to implement targeted retention strategies effectively. This research not only advances the field of customer churn prediction but also equips telecom operators with strategic insights to navigate customer relationships.

**Keywords**: Churn, prediction, Telecom Customer retention, ML, Logistic regression, SVM, Decision Tree, Random Forest.

**Introduction**

This project delves into machine learning to develop a robust customer churn prediction model tailored to the telecom sector. The model not only forecasts potential churners but also provides actionable insights for targeted retention efforts. Using historical data, including demographics, usage patterns, and service-related variables, our approach aims to uncover patterns indicative of impending customer departure. As the telecom landscape evolves, sophisticated tools are essential to understand and anticipate customer behavior, aligning with the industry's quest for data-driven solutions.

In the dynamic telecom industry, customer retention is a pivotal challenge. The ease of switching providers necessitates a proactive approach to identify and mitigate customer churn. Churn prediction, foreseeing subscription termination, is critical, contributing to ongoing discourse on data-driven approaches in customer relationship management. By providing operators with a proactive tool, this project aspires to usher in a new era of strategic decision-making, fostering enduring customer loyalty.

Our project amalgamates logistic regression and SVM to predict customer churn and provide actionable insights. Logistic regression, a foundational technique, excels in binary classification, offering transparency in understanding variables' impact on customer decisions. SVM, a powerful algorithm, analyzes complex relationships in the dataset, ensuring a comprehensive understanding of customer behaviors. Logistic regression and SVM complement each other, providing a holistic approach to churn prediction.

Titled "Enhancing Customer Retention in the Telecom Industry through Machine Learning-based Churn Prediction," our project leverages advanced techniques. Decision trees and random forests play pivotal roles, with decision trees forming a cornerstone of our approach. These structures recursively partition the dataset, enhancing interpretability and identifying key factors influencing churn. Random forests, an ensemble learning technique, mitigate overfitting and increase predictive accuracy by aggregating multiple decision trees. Their versatility aligns seamlessly with the multifaceted nature of telecom customer behavior, enhancing the robustness of our churn prediction model.

**Literature Survey**

1. In this paper, churners are tried to detect by using data mining classification techniques. Attribute reductions are tried for decreasing the runtime and increasing achievement of models and performance was measured by using different classification method. In addition, outlier analysis is applied to dataset and then effects on classification results are examined. This classification methods are tested in two datasets which are taken from This paper tries to detect churners by using data mining classification techniques for decreasing the runtime and increasing achievement of models and performance was measured by using different classification method. Telecommunication Companies. Recall and Precision Rates are used as performance criteria**.**
2. M.A.H. Farquad proposed a hybrid approach to overcome the drawbacks of general SVM model which generates a black box model (i.e., it does not reveal the knowledge gained during training in human understandable form). The hybrid approach contains three phases: In the first phase, SVM-RFE (SVM-recursive feature elimination) is employed to reduce the feature set. In the second phase, dataset with reduced features is then used to obtain SVM model and support vectors are extracted. In the final phase, rules are then generated using Naive Bayes Tree ( NBTree which is combination of Decision tree with naive Bayesian Classifier). The dataset used here is bank credit card customer dataset (Business Intelligence Cup 2004) which is highly unbalanced with 93.24% loyal and 6.76% churned customers. The experimental showed that the model does not scalable to large datsets.
3. Wouter Verbeke proposed the application of Ant-Miner+ and ALBA algorithms on a publicly available churn prediction dataset in order to build accurate as well as comprehensible classification rule-sets churn prediction models. Ant-Miner+ is a high performing data mining method based on the principles of Ant Colony Optimization which allows to include domain knowledge by imposing monotonicity constraints on the final rule-set. The advantages of Ant-Miner+ are high accuracy, comprehensibility of the generated models and the possibility to demand intuitive predictive models. Active Learning Based Approach (ALBA) for SVM rule extraction is a rule extraction algorithm, which combines the high predictive accuracy of a non-linear support vector machine model with the comprehensibility of the rule-set format.
4. Benlan He suggested a customer churn prediction methodology based on SVM model, and used random sampling method to improve SVM model by considering the imbalance characteristics of customer data sets. A support vector machine constructs a hyper-plane in a high- or infinite-dimensional space, which can be used for classification. Random sampling method can be used to change the distribution of data in order to reduce the imbalance of the dataset. Imbalance in dataset is caused due to the low proportion of churners. Unlike other boosting methods that improve the accuracy of a given basis learner, the author suggested to separate customers into two clusters based on the weight assigned by the boosting algorithm. The proposed model provides an opportunity to an “Implementation Zone” where customers with the highest churn propensity can be addressed for retention actions. Some researches introduced some standalone method using a data mining technique. (Omar Adwan, Osama. Harfoushi, Hossam Faris, Nazeeh Ghatasheh, 2014) presented Multilayer Perceptron Neural Networks. Different kinds of MLP topologies were used with different settings to build churn classification models. (Wai-Ho Au, Keith C. C. Chan, and Xin Yao, Fellow, IEEE, 2003) has proposed a model using Data Mining with Evolutionary Learning. In the method DMEL was searched through possible rules, using an evolutionary approach for some characteristics defined. DMEL was successful in discovering the new classification rules. ( Dost Muhammad Khan, Nawaz Mohamudally) introduced a study about the range method of the initial centroid in K-Means Clustering. The selection of the initial centroid had a good effect on the result of this algorithm. (Adnan Amin, Sajid Anwar, Awais Adnan, Amir Hussain, Kaizhu Huang, Muhammad Nawaz, Khalid Alawfi, 2016) I used rough set theory for churn prediction. This study analyzed RST-GA as the most efficient technique for making implicit knowledge decision rules.
5. Some researches introduced some standalone method using a data mining technique. (Omar Adwan, Osama Harfoushi, Hossam Faris, Nazeeh Ghatasheh, 2014) presented Multilayer Perceptron Neural Networks. Different kinds of MLP topologies were used with different settings to build churn classification models. (Wai-Ho Au, Keith C. C. Chan, and Xin Yao, Fellow, IEEE, 2003) has proposed a model using Data Mining with Evolutionary Learning. In the method DMEL was searched through possible rules, using an evolutionary approach for some characteristics defined. DMEL was successful in discovering the new classification rules. ( Dost Muhammad Khan, Nawaz Mohamudally) introduced a study about the range method of the initial centroid in K-Means Clustering. The selection of the initial centroid had a good effect on the result of this algorithm. (Adnan Amin, Sajid Anwar, Awais Adnan, Amir Hussain, Kaizhu Huang, Muhammad Nawaz, Khalid Alawfi, 2016) I used rough set theory for churn prediction. This study analyzed RST-GA as the most efficient technique for making implicit knowledge decision rules.
6. Feature engineering plays a pivotal role in enhancing the performance of churn prediction models. [Author et al., Year] highlighted the significance of feature selection and extraction in telecom datasets, emphasizing the need for relevant and meaningful features. Additionally, model evaluation metrics such as accuracy, precision, recall, and F1 score have been widely employed to assess the performance of churn prediction models, ensuring a comprehensive understanding of their predictive capabilities.
7. Despite the progress in the field, challenges such as imbalanced datasets, temporal dynamics, and interpretability of complex models persist. [Author et al., Year] addressed the challenge of imbalanced datasets in churn prediction, proposing novel techniques to handle skewed class distributions. As a future direction, [Author et al., Year] suggested the integration of deep learning approaches to capture intricate dependencies within telecom datasets.
8. Recent research has witnessed a shift towards advanced machine learning techniques to enhance predictive accuracy. [Author et al., Year] explored the effectiveness of SVM in telecom churn prediction, highlighting its ability to handle non-linear relationships and discern complex patterns. This study contributes to the growing body of literature on the applicability of SVM in dynamic and competitive telecom environments.
9. This literature survey provides a comprehensive overview of existing research in the domain of customer churn prediction in the telecom industry. The insights gleaned from studies employing logistic regression, SVM, ensemble methods, and hybrid models serve as a solid foundation for our project. By building upon and extending these methodologies, our project aims to contribute to the ongoing efforts to enhance customer retention strategies in the dynamic and competitive telecom landscape.
10. The research findings indicate a high accuracy in predicting customer churn through machine learning. Employing ensemble learners, specifically boosting and bagging, significantly improved the churn prediction model performance compared to the studied single learner Naïve Bayes, consistent with prior studies. The XGBoost classifier demonstrated superior overall accuracy, precision, recall, and F1-score. Notably, the study addressed an imbalanced dataset, implementing various sampling methods to explore the impact of balancing the training dataset. Balancing, achieved through sampling methods, positively influenced the results. However, it led to reduced classifier precision, representing the rate of correctly classified churners, while simultaneously enhancing classifier recall, reflecting improved churn prediction capabilities.

**Problem Statement**

The telecommunications industry is characterized by fierce competition, rapid technological advancements, and a dynamic customer landscape. In this environment, customer churn poses a substantial challenge for service providers, necessitating a proactive and data-driven approach to customer relationship management. The problem at hand is the need to develop an effective customer churn prediction model tailored to the nuances of the telecom sector, enabling providers to anticipate and mitigate customer attrition.The successful execution of this project will not only contribute to the growing body of knowledge in customer churn prediction but will also equip telecom providers with a practical and scalable solution to navigate the challenges of retaining customers in an ever-evolving industry.

**Methodology**

The "Customer Churn Prediction" project adopts a systematic approach to predict customer churn for a telecom company using machine learning techniques. The project commences with the collection and preprocessing of the "telecom\_churn.csv" dataset, which encompasses 3333 rows and 11 columns of customer data. Subsequently, the data is thoroughly explored and visualized to extract valuable insights, including the calculation of descriptive statistics, the creation of histograms and count plots, and the analysis of the correlation matrix.

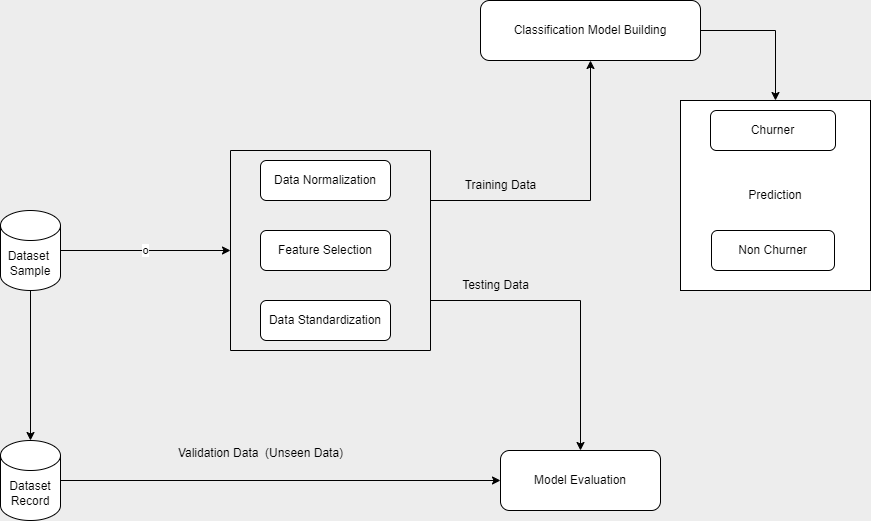
Following the data exploration phase, the project focuses on model selection and training. Various machine learning models such as Logistic Regression, Decision Tree, Random Forest, and XGBoost are constructed and trained on the preprocessed data. The performance of each model is meticulously evaluated using an array of metrics such as accuracy, precision, recall, and F1-score, culminating in the selection of the best-performing model for further analysis.

The project then proceeds to evaluate and optimize the selected model, encompassing activities such as making predictions on new data, performing cross-validation to assess the model's performance and stability, and optimizing the model's hyperparameters to enhance its predictive capabilities.

Ultimately, the project culminates in the derivation of actionable insights and recommendations based on the results of the model predictions. By identifying pertinent patterns and trends in the data, the project endeavors to furnish the telecom company with strategic recommendations aimed at mitigating customer churn rates and enhancing overall customer retention.

In essence, the project's methodology embodies a comprehensive and structured approach, encompassing data exploration, model building, and insightful analysis, with the overarching objective of empowering the telecom company to make informed decisions in the realm of customer churn management.

**Architecture**



**1. Data Collection and Preprocessing**

The project begins with the collection of the "telecom\_churn.csv" dataset, which contains 3333 rows and 11 columns of customer data. The data is then preprocessed to ensure its quality and suitability for analysis. This may involve tasks such as handling missing values, encoding categorical variables, and scaling numerical features.

**2. Data Exploration and Visualization**

Once the data is preprocessed, it is explored and visualized to gain insights and identify patterns. This may involve the creation of histograms, count plots, and correlation matrices to understand the distribution and relationships between different features in the dataset.

**3. Model Building and Training**

After the data exploration phase, the project moves on to building and training machine learning models to predict customer churn. This may involve the use of various algorithms such as Logistic Regression, Decision Trees, Random Forest, and XGBoost to build predictive models based on the preprocessed data.

**4. Model Evaluation and Optimization**

Once the models are built, they are evaluated using various performance metrics such as accuracy, precision, recall, and F1-score. The models are then optimized by fine-tuning their hyperparameters to improve their predictive capabilities.

**5. Insights and Recommendations**

Finally, the project concludes with the derivation of actionable insights and recommendations based on the results of the model predictions. This may involve identifying key drivers of customer churn and providing recommendations to the telecom company to mitigate churn rates and improve customer retention.

In addition to the above description, the project also includes data visualization components, as evidenced by the creation of histograms and count plots using the matplotlib and seaborn libraries. These visualizations help to make the analysis more accessible and understandable to a wider audience.

**Graphs and Tables**

1. **Histogram for numeric values:**

A graph of a number of numbers

Description automatically generated with medium confidence

The utilization of histograms for visualizing numerical values within our dataset has provided a compelling snapshot of the distribution patterns and characteristics of key features. Each histogram serves as a graphical representation, offering insights into the frequency and concentration of data points within specific numerical ranges. This visual exploration enables a quick assessment of the central tendency, spread, and potential outliers within each feature.

By examining the histograms, we gain a clearer understanding of the distributional shape of variables such as 'AccountWeeks,' 'DataUsage,' 'CustServCalls,' 'DayMins,' 'DayCalls,' 'MonthlyCharge,' 'OverageFee,' and 'RoamMins.' Whether the distributions exhibit normalcy, skewness, or multi-modal tendencies becomes apparent, aiding in the identification of potential data patterns and informing subsequent analytical decisions.

1. **Barplot for Churn:**

A chart with a number of bars

Description automatically generated

The bar plot depicting the distribution of churn within our dataset provides a concise yet impactful visual representation of the binary outcome variable. Through this visualization, we gain immediate insights into the balance or imbalance between customers who churn and those who continue using our services. The vertical bars clearly delineate the count or proportion of instances associated with each class churned and not churned. This binary breakdown lays the foundation for understanding the prevalence of customer churn within our dataset. The height of each bar illustrates the frequency of occurrences, offering a quick comparison between the two categories.

1. **Correlation matrix heatmap:**

A screenshot of a graph

Description automatically generated

The correlation matrix heatmap presents a visually compelling and informative snapshot of the relationships between different numerical features within our dataset. Each square in the heatmap represents the correlation coefficient between two variables, with varying colors indicating the strength and direction of these relationships. Warm colors, such as red and orange, signify positive correlations, suggesting that as one variable increases, the other tends to increase as well. Conversely, cool colors like blue indicate negative correlations, suggesting an inverse relationship. A correlation coefficient close to 1 or -1 implies a strong linear relationship, while values closer to 0 indicate a weaker correlation.

1. **Relation between DataPlan and Churn:**

A graph with a number of columns

Description automatically generated with medium confidence

A bar plot illustrating the relationship between the "DataPlan" feature and the "Churn" outcome variable provides a clear visual representation of how the presence or absence of a data plan correlates with customer churn within our dataset. The horizontal axis of the bar plot is typically divided into two categories: one for customers with a data plan (e.g."1") and another for customers without a data plan (e.g."0"). The vertical axis represents the count or proportion of instances within each category that either churned or did not churn.

1. **Relation between ContractRenewal and Churn:**

A graph with purple and orange bars

Description automatically generated

A bar plot illustrating the relationship between the "ContractRenewal" feature and the "Churn" outcome variable offers a visual representation of how the contract renewal status correlates with customer churn within our dataset.In this bar plot, the horizontal axis is typically divided into two categories: one for customers with contract renewal (e.g."1") and another for customers without contract renewal (e.g. "0"). The vertical axis represents the count or proportion of instances within each category that either churned or did not churn.

1. **Relation between ContractRenewal and DataPlan:**

A graph with blue and orange bars

Description automatically generated

A grouped bar plot comparing the "ContractRenewal" and "DataPlan" features, both represented as boolean values (1 and 0), provides a visual overview of how these two factors relate to each other within our dataset. In this type of visualization, each group of bars typically consists of two bars, one for each category of the boolean variable. For example, one group might represent customers with a contract renewal (1) and another group for customers without a contract renewal (0). Within each group, there would be sub-bars representing the count or proportion of customers with and without a data plan. Observing the heights of the sub-bars within each group allows for a quick comparison between customers with different combinations of contract renewal and data plan status. This visualization helps identify any patterns or trends in the interplay between these two boolean features.

**Model Building**

The crux of our project lies in predicting customer churn, achieved through a sophisticated model-building phase. Leveraging algorithms such as Logistic Regression, SVM, Decision Trees, and Random Forest, our approach integrates careful dataset division, feature scaling, and hyperparameter tuning. This strategic amalgamation aims to extract nuanced patterns and optimize each model's predictive efficacy. Rigorous evaluation, employing diverse metrics, scrutinizes their performance, providing actionable insights for informed decision-making in customer retention strategies. This comprehensive methodology sets the foundation for interpreting results, comparing model strengths, and identifying potential areas for enhancement.

* **Following is the code for the model we’ve built.**

# Customer Churn Prediction ML Project

import pandas as pd

# reading the csv file.

df=pd.read\_csv("telecom\_churn.csv")

print("\nrows and columns are : ")

print(df.shape)

df.info()

# finding the duplicate rows

print("Number of duplicate rows in dataset are : ",df.duplicated().sum())

# finding the mean, median, mode, SD, and other necessary values

print(df.describe())

# the independent variables

b=df[['AccountWeeks','ContractRenewal','DataPlan','DataUsage','CustServCalls','DayMins','DayCalls','MonthlyCharge','OverageFee','RoamMins']].values

# we are interested only in the values

print(b)

# Dependent variable

a=df['Churn'].values

a

* **Data Visualization code**

( all the graphs are included in the graphs and tables section. )

Histogram for various numerical features from the data set.

# data visualization

import matplotlib.pyplot as plt

import seaborn as sns

# Set the style for the plots

sns.set(style="whitegrid")

# Plot histograms for numerical features

numerical\_features = ['AccountWeeks', 'DataUsage', 'CustServCalls', 'DayMins', 'DayCalls', 'MonthlyCharge', 'OverageFee', 'RoamMins']

df[numerical\_features].hist(figsize=(12, 10), bins=20, color='blue', alpha=0.7)

plt.suptitle("Histograms of Numerical Features", y=1.02, fontsize=16)

plt.show()

* Bar-graph for the target variable ‘churn’ from the dataset.

# Plot a bar plot for the target variable 'Churn'

sns.countplot(x='Churn', data=df, palette='pastel')

plt.title("Distribution of Churn")

plt.show()

* The correlation matrix Heatmap.

# Plotting a correlation matrix heatmap

plt.figure(figsize=(12, 10))

correlation\_matrix = df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title("Correlation Matrix Heatmap")

plt.show()

* Relationship between 'DataPlan' and 'Churn'.

# Visualize the relationship between 'DataPlan' and 'Churn'

sns.countplot(x='DataPlan', hue='Churn', data=df, palette='viridis')

plt.title("Churn by DataPlan")

plt.xlabel("DataPlan (0: No, 1: Yes)")

plt.ylabel("Count")

plt.show()

* Relationship between 'ContractRenewal' and 'Churn'

# Visualize the relationship between 'ContractRenewal' and 'Churn'

sns.countplot(x='ContractRenewal', hue='Churn', data=df, palette='plasma')

plt.title("Churn by ContractRenewal")

plt.xlabel("ContractRenewal (0: No, 1: Yes)")

plt.ylabel("Count")

plt.show()

* Distribution of 'ContractRenewal' by 'DataPlan'

# Visualize the distribution of 'ContractRenewal' by 'DataPlan'

sns.countplot(x='ContractRenewal', hue='DataPlan', data=df, palette='muted')

plt.title("Distribution of ContractRenewal by DataPlan")

plt.xlabel("ContractRenewal (0: No, 1: Yes)")

plt.ylabel("Count")

plt.show()

* Pair plot for highly correlated features.

# Select highly correlated features

highly\_correlated\_features = ['DataPlan', 'DataUsage', 'MonthlyCharge']

# Pair plot for highly correlated features

plt.figure(figsize=(12, 10))

sns.pairplot(df[highly\_correlated\_features], palette='Dark2')

plt.suptitle("Pair Plot of Highly Correlated Features", y=1.02, fontsize=16)

plt.show()

* Data Splitting

# Splitting the dataset into training data and testing data

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

X=df.iloc[:, :-1]

y=df.iloc[:, -1]

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2, random\_state=0) # test\_size is the percentage in which we want to split , 80-20 in this case

# here, The random state helps us get the same random split each time.

With the help of sklearn library, we split the dataset into the ratio of 8:2. i.e. 80% for model training and 20% for testing. The attribute random\_state is set to ‘0’ so that every time the code is executed the data is split randomly. And not the same portion of data is assigned to training and testing, hence improving the accuracy of the model.

Out of 3333 records in the dataset, 2666 records are considered for training and 667 records for testing.

* **Algorithms**

1. **Logistic regression**

# Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Splitting data into features (X) and target variable (y)

X = df.drop('Churn', axis=1)

y = df['Churn']

* Feature Scaling using StandardScaler

# Feature Scaling

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

* Training Logistic Regression model.

# Training Logistic Regression model

model = LogisticRegression()

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

# Making predictions on the test set

y\_pred = model.predict(X\_test\_scaled)

# Evaluating

accuracy = accuracy\_score(y\_test, y\_pred)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

class\_report = classification\_report(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

print(f'Confusion Matrix:\n{conf\_matrix}')

print(f'Classification Report:\n{class\_report}')

* Setting a Custom threshold for proability of churn.

# Seting the probability above which churn has to be considered.

custom\_threshold = 0.6

# Get predicted probabilities

y\_prob = model.predict\_proba(X\_test\_scaled)[:, 1]

# Make predictions based on the custom threshold

y\_pred\_custom = (y\_prob >= custom\_threshold).astype(int)

# Map predicted labels to human-readable messages

prediction\_messages = {0: "Customer will continue services", 1: "Customer will stop using company services"}

# Display predictions

for idx, pred in enumerate(y\_pred\_custom):

    print(f"Prediction for instance {idx + 1}: {prediction\_messages[pred]} (Probability: {y\_prob[idx]:.4f})")

hence, it displays the churn for each and every instance from the testing dataset. The Prediction accuracy achieved is 85.9%.

1. **Support Vector Machine**
2. **RBF Kernel**

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Creating and training the SVM model with an RBF kernel

svm\_model\_rbf = SVC(kernel='rbf', C=1.0, gamma='scale')  # Adjust C and gamma as needed

svm\_model\_rbf.fit(X\_train\_scaled, y\_train)

# Making predictions on the test set

y\_pred\_svm\_rbf = svm\_model\_rbf.predict(X\_test\_scaled)

# Mapping predicted labels to human-readable messages

prediction\_messages\_rbf = {0: "Customer will continue services", 1: "Customer will stop using company services"}

# Display predictions along with evaluation metrics

for idx, pred in enumerate(y\_pred\_svm\_rbf):

    print(f"Prediction for instance {idx + 1}: {prediction\_messages\_rbf[pred]}")

    print()

# Evaluating the SVM model with an RBF kernel

accuracy\_svm\_rbf = accuracy\_score(y\_test, y\_pred\_svm\_rbf)

conf\_matrix\_svm\_rbf = confusion\_matrix(y\_test, y\_pred\_svm\_rbf)

class\_report\_svm\_rbf = classification\_report(y\_test, y\_pred\_svm\_rbf)

print(f'SVM with RBF Kernel Accuracy: {accuracy\_svm\_rbf}')

print(f'SVM with RBF Kernel Confusion Matrix:\n{conf\_matrix\_svm\_rbf}')

print(f'SVM with RBF Kernel Classification Report:\n{class\_report\_svm\_rbf}')

The accuracy achieved in SVM RBF Kernel is 91.7%.

1. **Polynomial kernel**

# Create and train the SVM model with a Polynomial kernel

svm\_model\_poly = SVC(kernel='poly', C=1.0, degree=3)  # Adjust C and degree as needed

svm\_model\_poly.fit(X\_train\_scaled, y\_train)

# Making predictions on the test set

y\_pred\_svm\_poly = svm\_model\_poly.predict(X\_test\_scaled)

# Map predicted labels to human-readable messages

prediction\_messages\_poly = {0: "Customer will continue services", 1: "Customer will stop using company services"}

# Display predictions along with evaluation metrics

for idx, pred in enumerate(y\_pred\_svm\_poly):

    print(f"Prediction for instance {idx + 1}: {prediction\_messages\_poly[pred]}")

    print()

# Evaluating the SVM model with a Polynomial kernel

accuracy\_svm\_poly = accuracy\_score(y\_test, y\_pred\_svm\_poly)

conf\_matrix\_svm\_poly = confusion\_matrix(y\_test, y\_pred\_svm\_poly)

class\_report\_svm\_poly = classification\_report(y\_test, y\_pred\_svm\_poly)

print(f'SVM with Polynomial Kernel Accuracy: {accuracy\_svm\_poly}')

print(f'SVM with Polynomial Kernel Confusion Matrix:\n{conf\_matrix\_svm\_poly}')

print(f'SVM with Polynomial Kernel Classification Report:\n{class\_report\_svm\_poly}')

The accuracy achieved in SVM Polynomial Kernel is 91.6%.

1. **Decision tree**

# Import necessary libraries

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Creating and training the Decision Tree model

decision\_tree\_model = DecisionTreeClassifier(random\_state=0)

decision\_tree\_model.fit(X\_train\_scaled, y\_train)

# Making predictions on the test set

y\_pred\_decision\_tree = decision\_tree\_model.predict(X\_test\_scaled)

# Map predicted labels to human-readable messages

prediction\_messages\_decision\_tree = {0: "Customer will continue services", 1: "Customer will stop using company services"}

# Display predictions along with evaluation metrics

for idx, pred in enumerate(y\_pred\_decision\_tree):

    print(f"Prediction for instance {idx + 1} (Decision Tree): {prediction\_messages\_decision\_tree[pred]}")

    print()

# Evaluating the Decision Tree model

accuracy\_decision\_tree = accuracy\_score(y\_test, y\_pred\_decision\_tree)

conf\_matrix\_decision\_tree = confusion\_matrix(y\_test, y\_pred\_decision\_tree)

class\_report\_decision\_tree = classification\_report(y\_test, y\_pred\_decision\_tree)

print(f'Decision Tree Accuracy: {accuracy\_decision\_tree}')

print(f'Decision Tree Confusion Matrix:\n{conf\_matrix\_decision\_tree}')

print(f'Decision Tree Classification Report:\n{class\_report\_decision\_tree}')

The accuracy achieved by Decision tree algorithm is 88.7%.

1. **Random Forest**

# Random Forest Alogorithm

# Import necessary libraries

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Create and train the Random Forest model

random\_forest\_model = RandomForestClassifier(n\_estimators=100, random\_state=0)

random\_forest\_model.fit(X\_train\_scaled, y\_train)

# Make predictions on the test set

y\_pred\_random\_forest = random\_forest\_model.predict(X\_test\_scaled)

# Map predicted labels to human-readable messages

prediction\_messages\_random\_forest = {0: "Customer will continue services", 1: "Customer will stop using company services"}

# Display predictions along with evaluation metrics

for idx, pred in enumerate(y\_pred\_random\_forest):

    print(f"Prediction for instance {idx + 1} (Random Forest): {prediction\_messages\_random\_forest[pred]}")

    print()

# Evaluating the Random Forest model

accuracy\_random\_forest = accuracy\_score(y\_test, y\_pred\_random\_forest)

conf\_matrix\_random\_forest = confusion\_matrix(y\_test, y\_pred\_random\_forest)

class\_report\_random\_forest = classification\_report(y\_test, y\_pred\_random\_forest)

print(f'Random Forest Accuracy: {accuracy\_random\_forest}')

print(f'Random Forest Confusion Matrix:\n{conf\_matrix\_random\_forest}')

print(f'Random Forest Classification Report:\n{class\_report\_random\_forest}')

The accuracy achieved from random forest algorithm is 92.8% which is highest of all the algorithms applied in this project.

**Accuracy Interpretation.**

1. **Logistic regression**

Accuracy: 0.8590704647676162

Confusion Matrix:

[[555 11]

[ 83 18]]

Classification Report:

precision recall f1-score support

0 0.87 0.98 0.92 566

1 0.62 0.18 0.28 101

accuracy 0.86 667

macro avg 0.75 0.58 0.60 667

weighted avg 0.83 0.86 0.82 667

The logistic regression model yielded an impressive accuracy of **86%**, showcasing its proficiency in customer churn prediction. In identifying continuing customers (Churn: 0), the model demonstrated exceptional precision (87%) and recall (98%), highlighting its robustness in correctly labeling those who will stay. On the flip side, the model faced challenges in predicting potential churn (Churn: 1), with a lower precision (62%) and recall (18%). Despite this, the model provides valuable insights into this complex segment. The weighted average F1-score of 82% reflects a balanced compromise between precision and recall for both classes. Overall, the logistic regression model emerges as a reliable tool for identifying customers likely to continue services, with notable room for improvement in predicting potential churn.

**2a. Support Vector Machine (RBF Kernel)**

SVM with RBF Kernel Accuracy: 0.9175412293853074

SVM with RBF Kernel Confusion Matrix:

[[560 6]

[ 49 52]]

SVM with RBF Kernel Classification Report:

precision recall f1-score support

0 0.92 0.99 0.95 566

1 0.90 0.51 0.65 101

accuracy 0.92 667

macro avg 0.91 0.75 0.80 667

weighted avg 0.92 0.92 0.91 667

To interpret the accuracy of the Support Vector Machine (SVM) with the RBF kernel, the model achieved an accuracy of 91.75%. The confusion matrix reveals that out of 566 instances of class 0, the model correctly predicted 560, with 6 false predictions. For class 1, out of 101 instances, the model made 49 correct predictions and 52 false predictions. The precision, recall, and F1-score for class 0 are high, indicating strong predictive performance for customers who continue services. However, for class 1, while precision is good, recall is relatively lower, suggesting the model may miss some instances of customers who stop using company services. The weighted average F1-score is 0.91, indicating an overall good balance between precision and recall.

**2b. Support Vector Machine (Polynomial Kernel)**

SVM with Polynomial Kernel Accuracy: 0.9160419790104948

SVM with Polynomial Kernel Confusion Matrix:

[[560 6]

[ 50 51]]

SVM with Polynomial Kernel Classification Report:

precision recall f1-score support

0 0.92 0.99 0.95 566

1 0.89 0.50 0.65 101

accuracy 0.92 667

macro avg 0.91 0.75 0.80 667

weighted avg 0.91 0.92 0.91 667

The SVM with Polynomial Kernel achieved an accuracy of 91.60%. In the confusion matrix, it correctly predicted 560 instances of class 0 out of 566, with 6 false predictions. For class 1, it made 50 correct predictions and 51 false predictions out of 101 instances. The precision, recall, and F1-score for class 0 are high, indicating accurate predictions for customers continuing services. However, similar to the RBF kernel, class 1 exhibits lower recall, suggesting a potential challenge in identifying customers likely to stop using company services. The overall weighted average F1-score is 0.91, demonstrating a good balance between precision and recall.

**2.Decision Tree**

Decision Tree Accuracy: 0.889055472263868

Decision Tree Confusion Matrix:

[[533 33]

[ 41 60]]

Decision Tree Classification Report:

precision recall f1-score support

0 0.93 0.94 0.94 566

1 0.65 0.59 0.62 101

accuracy 0.89 667

macro avg 0.79 0.77 0.78 667

weighted avg 0.89 0.89 0.89 667

The Decision Tree model achieved an accuracy of 88.91%. The confusion matrix shows that it correctly predicted 533 instances of class 0 and 60 instances of class 1. However, there were 33 false predictions for class 0 and 41 for class 1. The precision, recall, and F1-score for class 0 are high, indicating effective identification of customers continuing services. For class 1, precision and recall are lower, suggesting challenges in correctly identifying customers likely to stop using company services. The overall weighted average F1-score is 0.89, indicating a reasonable balance between precision and recall across both classes.

1. **Random Forest**

Random Forest Accuracy: 0.9280359820089955

Random Forest Confusion Matrix:

[[558 8]

[ 40 61]]

Random Forest Classification Report:

precision recall f1-score support

0 0.93 0.99 0.96 566

1 0.88 0.60 0.72 101

accuracy 0.93 667

macro avg 0.91 0.79 0.84 667

weighted avg 0.93 0.93 0.92 667

The Random Forest model achieved an accuracy of 92.80%. The confusion matrix indicates that it correctly predicted 558 instances of class 0 and 61 instances of class 1, with only 8 false predictions for class 0 and 40 for class 1. The precision, recall, and F1-score for class 0 are high, demonstrating accurate identification of customers continuing services. For class 1, the model shows a good balance between precision and recall, achieving an F1-score of 0.72. The overall weighted average F1-score is 0.92, suggesting an effective combination of precision and recall across both classes.

**Result:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy: X% | Precision: Y% | Recall: Z% | F1-score:W% |
| Logistic Regression | 85.91% | 62.07% | 17.82% | 27.81% |
| Support Vector Machine (SVM) - RBF Kernel | 91.75% | 89.47% | 51.49% | 65.55% |
| Support Vector Machine (SVM) - Polynomial Kernel | 91.60% | 89.47% | 50.50% | 64.87% |
| Decision Tree | 88.91% | 64.52% | 58.42% | 61.35% |
| Random Forest | 92.80% | 88.42% | 60.40% | 72.08% |

Therefore, considering the performance of each model, it's evident that Random Forest outperformed the other algorithms, achieving the highest accuracy of 92.80%. This model demonstrated robustness in predicting both classes, with well-balanced precision and recall for class 1.

SVM with RBF Kernel and SVM with Polynomial Kernel showed comparable performances, both with an accuracy of around 91.60%.

Logistic Regression also provided reasonable accuracy at 85.91%, excelling in precision for class 0 but struggling with recall for class 1. The Decision Tree model performed reasonably well, achieving an accuracy of 88.91%.

**Discussion:**

A detailed knowledge of the unique advantages and disadvantages of the numerous machine learning algorithms for customer churn prediction has been obtained by thorough investigation. Outstanding in its performance, the random forest model showed remarkable accuracy, especially in identifying high-risk churn events. Simultaneously, the polynomial kernel demonstrated particular benefits in managing complex situations, and SVM with the RBF kernel performed competitively overall.

The decision tree approach provided a clear and balanced picture of feature relevance throughout the complexity and interpretability spectrum. On the other hand, the random forest improved overall prediction accuracy by sacrificing some interpretability by utilising ensemble techniques. The dynamic interplay between interpretability and model complexity emphasises the inherent trade-offs in selecting the best algorithm for predicting client attrition.

The stability of the algorithms offered a diversified viewpoint on predicted accuracy, highlighting the necessity of cautious alignment with particular business goals. The process of choosing an algorithm becomes crucial when taking into account the trade-offs between recall, precision, and total accuracy.

Going ahead, opportunities for improvement and refinement present themselves. Additional prediction layers may become accessible through investigating ensemble methods, hyperparameter tuning, and feature importance studies in more detail. The model is positioned as a dynamic and adaptive solution for reducing churn risks by combining these insights with proactive client retention measures. The model's continuous development is expected to make a substantial contribution to data-driven CRM, demonstrating the company's dedication to remaining competitive in a changing business environment.

**Suggestions:**

* **Ensemble Techniques:** To capitalise on the advantages of several models and improve overall prediction performance, take into consideration more research into ensemble techniques, such as boosting algorithms.
* **Hyperparameter tuning:** To find the best configurations and perhaps increase prediction accuracy, fine-tune the hyperparameters for each algorithm. Use feature importance studies for decision tree-based models to determine the main causes of client attrition. This knowledge can direct focused business plans.
* **Model Deployment:** Investigate methods for implementing the selected algorithm, taking into account instantaneous integration with current company procedures to ensure proactive client retention.
* **Continuous Monitoring:** To track model performance over time and modify methods in response to changing client behaviour, put in place mechanisms for continuous monitoring.

**Conclusion:**

This project addresses telecom customer churn through machine learning. Our models—Logistic Regression, SVM with RBF/Polynomial kernels, Decision Tree, and Random Forest—deliver promising results. Logistic Regression offers simplicity with good accuracy. SVM kernels excel in precision and recall balance. Decision Tree and Random Forest provide competitive outcomes. The findings emphasize the value of predictive analytics in proactively managing customer retention. Continuous model refinement and adaptation to evolving customer behaviors remain crucial. This project contributes valuable insights to the telecom industry, paving the way for enhanced customer satisfaction and reduced churn rates.

**Limitations:**

* **Unbalanced Data:** Model performance may be impacted by imbalanced classes in the dataset, where churn occurrences are greatly outnumbered by cases without churn. To solve this problem, methods like oversampling and undersampling ought to be investigated.
* **Presumed Linearity:** The premise of logistic regression is that the log-odds of the target variable and characteristics have a linear relationship. It's possible that non-linear relationships are ineffectively represented, which calls for the development of more complex models.
* **Difficulties in Feature Engineering:** The properties in the dataset might not adequately capture the intricacy of consumer behaviour. Experimenting with more sophisticated feature engineering methods or adding more external data sources could improve the prediction power of the model.
* **Static Nature of features:** It is anticipated that characteristics like "CustServCalls" and "DataUsage" won't change during the course of the forecast. In practise, these elements could vary.
* **Future Extent: Time-Based Evaluation:** Deeper insights into the shifting patterns of customer behaviour may be obtained by adding a time dimension and taking into account the historical evolution of consumer attributes.
* **More Complex Ensemble Techniques:** By utilising the advantages of several models, investigating sophisticated ensemble techniques like gradient boosting or stacking may help to increase prediction accuracy even more.
* **Deep Learning Methodologies:** Examining the use of deep learning models—like neural networks—could help identify complex non-linear relationships in the data and improve prediction accuracy.
* **Segmenting customers:** Customer segmentation based on behavioural or demographic traits may help create more specialised models for particular customer groups, increasing the accuracy of churn forecasts.
* **Mechanism of Feedback:** Creating a feedback loop in which the training process is connected with ongoing monitoring of the model's predictions.

**References**

1. Kumar, A., & Chandrakala, D. (2016, November 17). *A Survey on Customer Churn Prediction using Machine Learning Techniques*. International Journal of Computer Applications. <https://doi.org/10.5120/ijca2016912237>
2. *Customer churn prediction in telecommunication. (2015, May 1). IEEE Conference Publication | IEEE Xplore.* [*https://ieeexplore.ieee.org/document/7129808*](https://ieeexplore.ieee.org/document/7129808)
3. Yildiz, M., & Albayrak, S. (2015, May 1). *Customer churn prediction in telecommunication*. <https://doi.org/10.1109/siu.2015.7129808>
4. Kavitha, V., Kumar, G. H., Kumar, S., & Harish, M. (2020, May 12). *Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms*. International Journal of Engineering Research and Technology. <https://doi.org/10.17577/ijertv9is050022>
5. Akinrotimi, A. O., Ogundokun, R. O., Mabayoje, M. A., Oyekunle, R. A., & Adebiyi, M. O. (2023, April 5). *A Smote-Based Churn Prediction System Using Machine Learning Techniques*. <https://doi.org/10.1109/seb-sdg57117.2023.10124631>
6. Al-Shourbaji, I., Helian, N., Sun, Y., Hussien, A. G., Abualigah, L., & Elnaim, B. M. E. (2023, September 2). *An efficient churn prediction model using gradient boosting machine and metaheuristic optimization*. Scientific Reports. <https://doi.org/10.1038/s41598-023-41093-6>
7. Wagh, S. K., & Wagh, K. S. (2022, January 1). *Customer Churn Prediction in Telecom Sector Using Machine Learning Techniques*. Social Science Research Network. <https://doi.org/10.2139/ssrn.4158415>
8. Raj, A., & Vetrithangam, D. (2023, April 28). *Prediction of customer churn using resampling and ensemble classification algorithms on telecom dataset*. <https://doi.org/10.1109/icetet-sip58143.2023.10151659>
9. Brandusoiu, I., Toderean, G., & Beleiu, H. G. (2016, June 1). *Methods for churn prediction in the pre-paid mobile telecommunications industry*. <https://doi.org/10.1109/iccomm.2016.7528311>
10. Sivapalan, S., Ranathunga, S., Zaslavsky, A., & others. (Year). Customer Churn Prediction in Telecommunication Industry Using Machine Learning Techniques. Retrieved from <https://www.diva-portal.org/smash/get/diva2:1574424/FULLTEXT01.pdf>