WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Customer Churn Analysis and Prediction

Priya Rani Mourya*, Shristi Kumari* ,Fauqia Taufique*, Divyansha Jhadi*

* Department of Computer Science, Shri Shankaracharya Group of Institutions, Durg, India.

ABSTRACT :

Customer churn, or customers who stop doing business with a company, is a significant concern for businesses. Predicting customer churn can help companies take proactive measures to retain customers. This paper provides an overview of customer churn prediction with a focus on machine learning algorithms such as logistic regression, decision trees, random forests, and artificial neural networks. The paper highlights the challenges of predicting customer churn and reviews the current state-of-the-art in customer churn prediction, including recent advances in deep learning and natural language processing. Finally, the paper discusses the practical implications of customer churn prediction, including its potential to improve customer retention and increase profits. The paper emphasizes that customer churn prediction is a complex and challenging problem that can benefit from machine learning algorithms, making it an increasingly important area of research for businesses looking to improve customer retention and maximize profits, health monitoring to help users reach their fitness goals safely and efficiently.

Keywords: Customer Churn, Random Forest, Machine Learning, AdaBoost, Random Forest.

I. INTRODUCTION

In today's fast-paced and competitive market, keeping your current customers happy is just as crucial—if not more—than bringing in new ones. When customers leave, known as *customer churn*, it can seriously affect a company's revenue and long-term success. That's why it's so important to understand *why* customers leave and, even better, to be able to predict *which* ones are most likely to go.

This project dives into Customer Churn Analysis and Prediction using Python. The goal is to dig into customer data, spot trends in their behavior, and build models that can predict whether a customer is likely to churn. With the help of data analysis and machine learning, we can uncover insights that help businesses take action before it's too late—like improving service, offering targeted promotions, or changing how they engage with customers.

We're using Python for this project because it's loaded with powerful tools for data science. Libraries like Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, and AdaBoost make it easy to clean data, visualize trends, and build predictive models.

By accurately identifying who's at risk of leaving and why, businesses can create smarter strategies to keep customers around longer—and that can make a real difference to the bottom line.

II. LITERATURE REVIEW

Over the years, many researchers and data scientists have explored how businesses can predict customer churn using data-driven methods. With the rise of big data and machine learning, this field has seen a lot of innovation—especially in how we gather insights from customer behavior and use that information to make better decisions.

A number of studies have shown that machine learning algorithms like **Logistic Regression**, **Decision Trees**, **Random Forests**, and **Gradient Boosting** are highly effective at identifying patterns that suggest a customer might leave. For example, Telco companies often rely on features such as monthly charges, contract type, and customer service interactions to predict churn. Researchers have found that models trained on these types of data can help companies intervene before a customer decides to leave.

One study by Idris et al. used ensemble methods, which combine multiple models to boost performance, and found that these approaches significantly improve accuracy when predicting churn. Another popular technique is feature selection, where only the most important customer attributes are used to train the model. This not only reduces noise in the data but also makes the model more interpretable and easier to act on.

Python, with its rich ecosystem of data science libraries like Pandas, NumPy, Scikit-learn, and AdaBoost, has become the go-to language for implementing these solutions. These tools make it easier to handle large datasets, build machine learning models, and visualize insights clearly.

III. METHODOLOGY

1. Data Collection:

The dataset used in this project was obtained from Kaggle and includes detailed information on 7,043 customers. It contains 21 different attributes, covering everything from customer identification and registration details to demographic data and account-related information. Before performing any analysis or building predictive models, it was important to thoroughly clean and prepare the data.

After this, the data preparation process by cleaning the dataset—removing missing values, duplicate entries, and any significant outliers that could affect the accuracy of the results. Once the data was clean and consistent, we proceeded with exploratory data analysis and started building models to predict customer churn, then also explored the creation of new features based on existing data. By identifying patterns in customer behavior, such as how frequently services were used, we were able to generate additional features that helped the model gain a deeper understanding of user habits. These insights contributed significantly to improving the model's prediction accuracy.

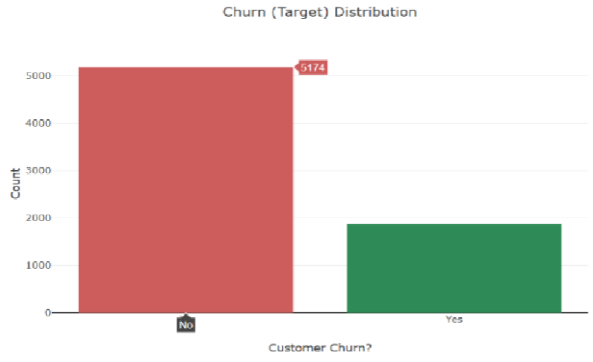
	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetServi
0	Female	0	Yes	No	1	No	No phone service	D
1	Male	0	No	No	34	Yes	No	D
2	Male	0	No	No	2	Yes	No	D
3	Male	0	No	No	45	No	No phone service	D
4	Female	0	No	No	2	Yes	No	Fiber op
...	...	...	...	...	...	...	...	
7038	Male	0	Yes	Yes	24	Yes	Yes	D
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber op
7040	Female	0	Yes	Yes	11	No	No phone service	D
7041	Male	1	Yes	No	4	Yes	Yes	Fiber op
7042	Male	0	No	No	66	Yes	No	Fiber op

2. Exploratory Data Analysis (EDA):

To better understand the data, the first step involved exploring it through both statistics and visualizations. This included looking at how the values of different features were spread out, checking if any information was missing, and spotting any unusual or extreme values that didn't fit the general pattern. Some new features were also created by transforming the existing ones to better capture customer behavior, while any data that didn't add value or seemed unnecessary was removed. It was also important to check how the features related to each other, especially to avoid situations where two variables told the same story—something that can confuse machine learning models. Visual tools like box plots were helpful in showing how different features might influence whether a customer stays or leaves. By looking at features one by one and also in combination, the most important factors related to customer churn started to become clear, setting the stage for building a strong prediction model.

The data was then organized based on the type of information each feature held. Some features were numerical—like charges or number of services used—and could go directly into the model, though they sometimes needed to be scaled so no single value had too much influence. Others were categorical, like gender or contract type, and needed to be converted into a numerical format first. This step was key because models work best when data is structured in a way they can understand.

Finally, while examining the target variable—whether customers churned or not—it became clear that the data was imbalanced. There were almost three times as many people who didn't leave compared to those who did. This kind of imbalance can lead a model to lean heavily toward predicting that customers will stay, which means it might miss the ones who are actually at risk of leaving. Knowing this ahead of time helps in planning how to address it, so the model can give fair and accurate predictions.



3. Feature Engineering

Once the data was cleaned and prepared, the next step was to focus on the most useful information. Important features like call duration, how often customers made calls, and how long they had been with the company were extracted from the dataset. Categorical details—such as contract type or payment method—were converted into a format that the machine learning models could understand using one-hot encoding. To make sure all the features were treated fairly by the models, their values were scaled so they were on a similar range. This step made a big difference in helping the models perform better by giving them clean, consistent, and meaningful data to work with.

4. Model Selection:

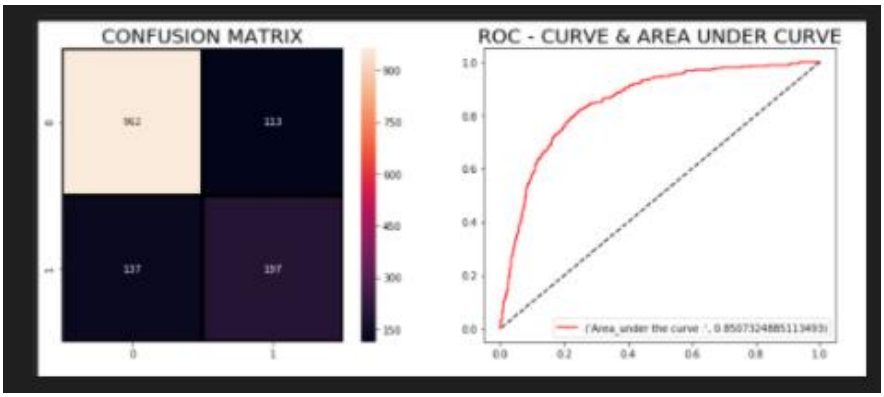
Several machine learning models were tested to see which ones could best predict customer churn. These included AdaBoost, Random Forest, and Logistic Regression classifiers. To get a fair and reliable assessment of each model's performance, 5-fold cross-validation was used, allowing each model to be tested across different subsets of the data. The models were evaluated using key metrics like accuracy, precision, recall, and F1 score. After comparing the results, the best-performing models were chosen for deeper analysis and refinement.

5. Model Evaluation:

In the evaluation phase, the performance of each trained model was carefully analyzed using a variety of metrics. While accuracy gave a quick snapshot of how many predictions were correct overall, it wasn't the only measure considered—especially since the dataset was imbalanced, with far more customers not churning than those who did. To get a more balanced view of performance, the F1-score was also calculated, as it takes both precision and recall into account, providing a better sense of how well the model handled both correct positive predictions and missed or incorrect ones.

To dive deeper into the results, a confusion matrix was used. This simple yet powerful tool broke down predictions into true positives, true negatives, false positives, and false negatives, making it easier to understand exactly where the model was getting things right—or going wrong. From these values, precision, recall, and F1-score were further refined and interpreted.

Finally, to evaluate how well each model could distinguish between customers likely to churn and those likely to stay, the ROC curve was plotted, and the area under the curve (AUC) was calculated. A higher AUC value indicated a stronger ability to separate the two classes, giving an additional layer of confidence in the model's predictive power.

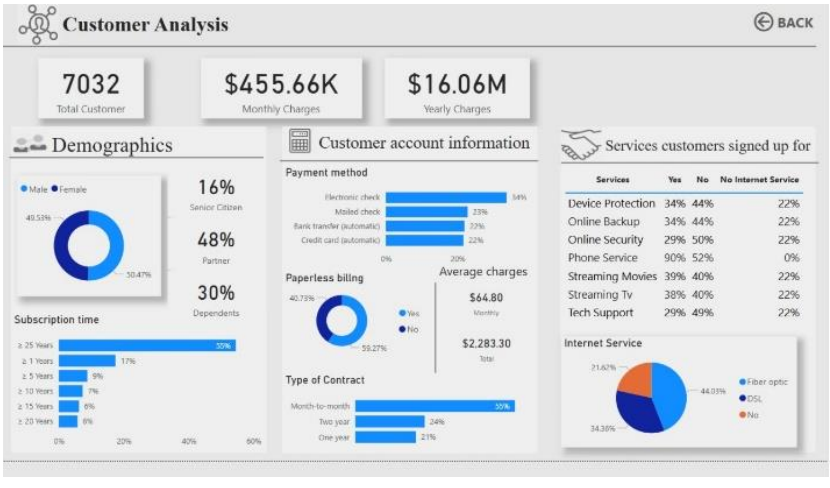


6. Predicting Customer Churn

Now the model is ready to predict customer churn. By entering a customer's details into the Streamlit web app—such as their contract type, monthly charges, tenure, and other relevant features—the system can instantly assess whether the customer is likely to stay or leave. Based on the logistic regression model's analysis, it provides a clear prediction, helping businesses take timely actions like offering discounts, personalized support, or loyalty rewards to reduce the chances of churn and improve customer retention.

IV. RESULTS

The result of the customer churn analysis part of the project led to the creation of an interactive Power BI dashboard that visually presents key insights from the data. This dashboard displays important trends related to customer demographics, service usage, billing patterns, and contract details. By analyzing these factors, the dashboard helps in identifying the common traits among customers who are likely to churn. It enables business teams to explore the data intuitively, filter views, and gain a deeper understanding of what drives customer attrition. This visual representation makes it easier to communicate findings and supports data-driven decision-making.



In the prediction phase, a logistic regression model was developed using Python to estimate the likelihood of customer churn based on historical data. The model was deployed through a Streamlit web application, making it easy to use and accessible without technical expertise. Users can enter customer details into the app, and the model instantly provides a prediction on whether that customer is likely to churn. This real-time prediction capability allows businesses to proactively identify at-risk customers and implement targeted strategies to retain them, enhancing customer satisfaction and reducing churn rates.

V.CONCLUSION

A telecom company can use insights from churn prediction models to better understand customer behavior and offer special deals or personalized incentives to keep customers from leaving. The results from the current model show that using machine learning techniques has led to more accurate and reliable predictions of which customers are likely to churn.

Going forward, removing unnecessary features could help make the model even more accurate, and testing additional machine learning methods might further improve its performance.

This approach also includes a comparison of different machine learning algorithms along with sampling strategies to better handle imbalanced data. The main goal is to identify customers at high risk of leaving and engage them proactively with targeted offers or communication, making them more likely to stay with the service.

VI.REFERENCES

[1] Idris, A., Khan, A., & Lee, Y. S. (2012). Intelligent churn prediction in telecom: employing mRMR feature selection and rotBoost based ensemble classification. *Applied Intelligence*, 39(3), 659–672. https://doi.org/10.1007/s10489-012-0380-9

[2] Verbeke, W., Martens, D., Mues, C., & Baesens, B. (2012). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 38(3), 2354–2364. https://doi.org/10.1016/j.eswa.2010.08.090

[3] Kaggle. (n.d.). *Telco Customer Churn Dataset*. Retrieved from https://www.kaggle.com/blastchar/telco-customer-churn

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[5] McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51–56.

[6] Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

[7] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.