Regression Analysis of Customer Churn Using Random Forest,Decision Tree,Gradient Boost.

ABIJITH PANDATH
ID No: 13395186
pandatha@uni.coventry.ac.uk
7072CEM - Machine Learning

***Abstract*— This paper looks at a predictive model to calculate the number of customers who churn based on various customer characteristics and usage trends. To analyze the dataset and create an accurate churn prediction model, we use machine learning techniques such as Random Forest Regression, Decision Tree Regression and Gradient Boost Regression. We begin by undertaking exploratory data analysis to gather insights into the dataset's properties, then we apply various data preprocessing techniques such as handling duplicate entries. Handling missing values, renaming columns, outlier detection and treatment and dropping irrelevant columns. Prediction findings and algorithm performance measurements were gathered and visualized for comparison and debate using the Python programming language and machine learning resources**.

Keywords; Machine Learning, regression models, random forest, gradient boost, Decision Tree.

I. INTRODUCTION

Our goal is to use machine learning algorithms to create prediction models that can properly anticipate customer churn. We hope to use this dataset to uncover the important characteristics that drive churn behavior and to develop algorithms capable of reliably identifying consumers as churners or non-churners.

Customer churn, or the pattern of consumers abandoning their engagement with a company, has become a key concern for businesses in a variety of industries. As client acquisition expenses continue to climb, organizations are increasingly focusing on customer retention as a method of sustaining growth and profitability. Understanding the elements that lead to customer turnover and deploying efficient churn prediction models may have a substantial influence on business performance. Several studies on customer turnover have been conducted in a variety of industries, including telecommunications, banking, e-commerce, and subscription-based services. To analyze and forecast customer turnover, researchers have used statistical and machine learning approaches like as logistic regression, decision trees, random forests, and neural networks Because of the significant competition and the prevalence of subscription-based services, customer turnover has been widely examined in the telecommunications industry, for example. According to research, call duration, frequency of service complaints, contract term, and customer demographics are all important indicators of turnover. Random forest regression models have been shown to be successful in capturing the complicated correlations between these variables and customer attrition. This investigation' findings can help Iranian telecommunications firms develop successful client retention strategies, improve customer satisfaction, and optimize their operations. Furthermore, the project's results may have larger significance for the global telecoms business, as churn behavior reveals shared patterns and motivations across economies. Understanding and anticipating customer churn, or the phenomena of consumers cancelling their subscriptions, is critical. Service providers should take proactive actions to retain clients and enhance overall customer satisfaction by recognizing variables that lead to churn.

II. PROBLEM AND DATA SET

Dataset was collected from UCI Machine Learning Repository and this data set consist of randomly collected details of customers from an Iranian Telecom company over a period of 12 months. The dataset contains a variety of consumer information, such as call failure rates, customer complaints, subscription durations, charge amounts, use patterns (such as the number of seconds used, frequency of use, and frequency of SMS), and demographic parameters such as age and tariff plans. The information also includes a churn indicator, which indicates whether a client has churned. Each attribute in the data set is about the telecom service-related data describing the frequency, duration, count, and tariff related information. Each instance is individual customers information on the service they have with the service provider. The columns' data types are mostly integers (int64), except for the "Customer Value" column, which is a floating-point number (float64). There are no missing values in the dataset (the non-null count is the same as the total count for all columns).

TABLE 1: DATASET FEATURES

SL No	Description	Type	Description
1	Call failure	Numerical	No. of call failures experienced by customers
2	Complaints	Numerical	No. of complaints made by customers
3	Subscription Length	Numerical	Length of the customer's subscription in months
4	Charge Amount	Numerical	Amount charged to customers
5	Seconds of Use	Numerical	Total No. of seconds customer used the service
6	Frequency of Use	Numerical	Frequency of service usage by customers
7	Frequency of SMS	Numerical	Frequency of SMS usage by customers
8	Distinct Called Numbers	Numerical	No. of unique phone numbers used by customers
9	Age Group	Categorical	Categorization of customers into different age groups
10	Tariff Plan	Categorical	Pricing Package chosen by customers
11	Status	Categorical	Customer status
12	Age	Numerical	Age of the customers
13	Churn	Categorical	Churn Status of customers: 1 = Churned, 0 = Not Churned

III. METHODS

Regression attempts to establish a link between a dependent variable (also known as the target variable or outcome variable) and one or more independent variables (also known as features or predictors). The dependent variable is the quantity we wish to forecast, and the independent variables are the variables that can impact the dependent variable.

In regression, the algorithm learns from a training dataset that includes input characteristics and target values. The method attempts to identify the best-fitting mathematical function that translates the input characteristics to the target variable during the learning process. The regression model is the name given to this function.

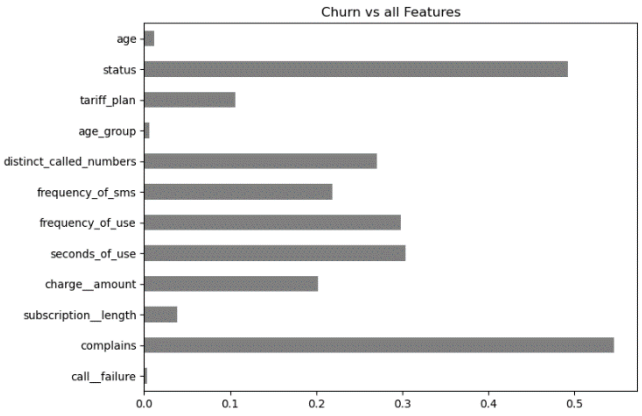
The prediction error is often used to assess the performance of a regression model. Mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and R-squared (coefficient of determination) are all common error measures. These measures measure how similar the expected and actual values are. In this paper we majorly focus 3 different Regression model from machine learning such as Random Forest, Decision Tree and Gradient Boost. Says, J. (2021, January 10).

Random Forest Regression : The algorithm in Random Forest Regression generates a forest, which is a collection of decision trees. Each decision tree is constructed from a randomly selected portion of the training data and a subset of the input characteristics. This unpredictability contributes to the diversity of the forest's individual trees. Random Forest Regression is commonly utilized in fields like as finance, healthcare, and environmental sciences, where accurate prediction of continuous variables is critical. It is well-known for its capacity to handle high-dimensional data, handle missing values, and give insights into the relevance of features. A Random Forest Regression model (rf_reg) is created, trained on training data (X_train and Y_train), and used to predict the target variable (Y_pred_rf) for testing data (X_test). R-squared value, mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) are all determined evaluation measures.

Decision Tree Regression: Decision tree regression is a machine learning approach that is used to predict continuous numerical values in regression applications. It makes predictions based on input features using a hierarchical system of decision rules. The approach constructs a tree-like model in decision tree regression, where each internal node represents a choice based on a characteristic and each leaf node represents a projected value. The tree is built by recursively splitting the input feature space depending on the features chosen and their threshold values. This stage is similar to Random Forest Regression in that it involves creating a Decision Tree Regression model (decision_tree), training it with training data, predicting the target variable (Y_pred_tree), and producing evaluation metrics.

Gradient Boost Regression: Gradient Boost Regression is a strong machine learning technique that performs regression tasks by combining the ideas of gradient descent optimization and ensemble learning. It is especially useful for developing reliable predictive models by iteratively minimizing prediction errors. Gradient Boost Regression is widely employed in many sectors and is well-known for producing cutting-edge results in numerous machine learning contests. It works especially well when paired with decision trees as basis models, resulting in algorithms like XGBoost, LightGBM, and CatBoost, which have become popular alternatives for regression problems. This stage follows the same pattern as the previous two. A Gradient Boosting Regression model (gradient boosting) is developed, trained using the training data, and used to predict the target variable (Y_pred_gradient) for the testing data, as well as assessment metrics.

Fig I: Churn Vs All Features



IV. EXPERIMENTAL SETUP

Data preparation is critical for guaranteeing the quality, dependability, and applicability of data for machine learning tasks. It enhances model performance, robustness, and generalization capabilities, resulting in more accurate and relevant insights. As a result, investing time and effort in data preparation is critical for the success of any machine learning project. This part covers data analysis on dataset such as handling duplicates, handling missing values, renaming columns, Outlier detection and treatment, dropping irrelevant columns and scaling the data set.

Handling Duplicate Entries: Applying drop_duplicates() removes any rows in the Data Frame that have identical values in all columns, ensuring that each row is unique. This can be beneficial when dealing with data that has duplicate items that must be removed to avoid misleading or duplicated information. Duplicate items in a dataset might skew analysis and produce false results. By removing duplicate entries, each data point is guaranteed to be unique, offering a more accurate depiction of the underlying population. This phase contributes to the dataset's integrity and dependability.

Handling Missing Values: You get the count of missing values for each column in the Data Frame by using isnull().sum() function. Missing values in datasets can arise for a variety of reasons, including poor data input, data corruption, or data transformation procedures. To avoid biased analysis, inaccurate calculations, or mistakes in downstream data processing, it is critical to handle missing numbers effectively. Missing values in a dataset can occur for a variety of reasons, including data collecting mistakes or incomplete data. Handling missing values is critical to ensuring the data analysis's quality and dependability.

Renaming Columns: In data preparation, renaming columns refers to the process of altering the names or labels of columns in a dataset. It is a standard stage in data preparation and data cleaning operations that tries to improve the data's clarity, uniformity, and interpretability. Column renaming entails standardizing column names to adhere to a uniform naming convention. This phase enhances the dataset's readability, maintainability, and compatibility. Consistent column names make data management, merging, and feature selection easier. The dataset becomes more accessible and favorable to efficient analysis and modelling by eliminating spaces with underscores and changing column names to lowercase. Ensuring the data set column names and the variables are same will be a best practice for better results.

Outlier Detection and Treatment: Outlier detection and treatment are critical processes in data preparation because outliers have a large influence on the quality and dependability of a machine learning model. Outliers are data points that depart dramatically from most the data's usual patterns or expected behavior. These anomalies might occur because of data collecting problems, measurement errors, or uncommon events. Data points that differ greatly from the rest of the dataset are referred to be outliers. Detecting and handling outliers is critical to ensuring that extreme results do not have an undue impact on the analysis or model performance. Outliers can corrupt statistical metrics, impair machine learning model performance, and lead to biased predictions. We limit the possible influence of extreme values on subsequent analysis by detecting outliers and restricting their values to a suitable upper bound (in this example). This allows for a more accurate depiction of the data.

Dropping Irrelevant Columns: Dropping unnecessary columns, also known as feature selection or feature removal, is an important step in machine learning data preparation. It entails eliminating characteristics from the dataset that are not relevant or significant to the predictive modelling objective. Irrelevant columns are eliminated from the dataset if they do not add significantly to the analysis or modelling job. This aids in data streamlining and increases computing efficiency. Removing unneeded columns minimizes data noise, simplifies future analysis, and avoids excessive processing expense. It narrows the emphasis of the analysis to the most important and relevant features, resulting in more interpretable and accurate results.

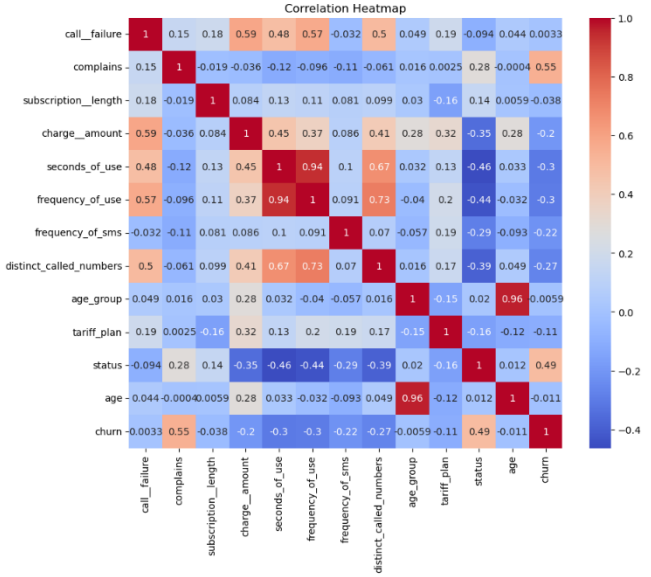
Splitting into Features and Target Variables: A crucial stage in supervised machine learning is splitting the dataset into features and target variables. It enables us to decouple the input attributes from the variable we wish to forecast, in this instance the churn state. We can train machine learning models to understand patterns and correlations between features and the target variable using this separation. The preprocessed dataset is divided into input features (X) and target variables (Y) in this stage. To produce the feature matrix (X), the 'churn' column is removed from the dataset, while the 'churn' column itself is allocated as the target variable (Y).

Scaling the Dataset: The StandardScaler is a preprocessing tool that standardizes numerical characteristics by scaling them to have a mean of zero and a variance of one. An instance of the StandardScaler is created in this phase, which will be used to conduct feature scaling on the feature matrix. The importance of nature scaling is that it puts all numerical characteristics to a same scale. It guarantees that no single trait, due to its greater size, dominates the learning process. Standardizing features also aids in improving model performance and stability, especially for models that rely on distance-based computations or optimization methods.

Fitting the Scalar and transforming the Data: The fit_transform() function is used to fit the scaler to the feature matrix (X). During the fitting phase, the scaler learns the mean and standard deviation of each feature and applies the scaling transformation to the data. The feature matrix (X) is now scaled and standardized after fitting the scaler to the features and processing the data. The modified data, which is saved in X_scaled, is now available for analysis and model training. Scaling the features guarantees that they are all on the same size, which improves the performance and convergence of many machine learning algorithms.

Splitting the Dataset into Training and Testing Sets: To divide the dataset into training and testing sets, use the train_test_split function. The function accepts as parameters the input characteristics (X) and the target variable (Y), as well as extra optional arguments. It is critical to divide the dataset into training and testing sets when evaluating the performance and generalization capabilities of machine learning models. The training set is used to train the model, whereas the testing set is used to evaluate how well the model works on unobserved data. The test_size option sets the percentage of the dataset that will be used for testing (in this example, 20%). The stratify option guarantees that the distribution of the target variable is preserved in both the training and testing sets. This is especially crucial when dealing with unbalanced datasets since it ensures that the class distributions in the training and testing sets are similar.

Fig II: Correlation Heat Map



A correlation heatmap is a graphical depiction of the correlation matrix, which displays the pairwise correlations between variables in a dataset. The correlation coefficient between two variables is represented by each cell in the heatmap, and the color scale reflects the strength and direction of the link. We may learn about the links between variables in the dataset by examining the correlation heatmap. This data is useful for feature selection, finding redundant variables, comprehending relationships, and directing further analysis or modelling decisions. Correlation heatmaps may be used to detect multicollinearity, which happens when two or more variables are significantly associated with one another.

V. RESULTS

TABLE II: MODEL RESULTS

Models	R Squared	Mean Squared	Root Mean Squared	Mean Absolute error
Random Forest	0.767	0.0305	0.174	0.0659
Decision Tree	0.570	0.0565	0.237	0.0570
Gradient Boost	0.668	0.0437	0.209	0.1086

In terms of several assessment measures, the Random Forest Regression model beat the other two models. It had the greatest R-squared score of 0.767, suggesting that the model can explain about 76.8% of the variation in the target variable. This shows a significant relationship between the characteristics and the target variable. The Random Forest model also produced the lowest Mean Squared Error (0.0305), Root Mean Squared Error (0.174), and Mean Absolute Error (0.0659). These metrics suggest that the model's predictions were more accurate than the other models. The Decision Tree Regression model, on the other hand, performed worse than the Random Forest model. It had a lower R-squared value (0.570) as well as greater error measures, such as a higher Mean Squared Error (0.0565), Root Mean Squared Error (0.237), and Mean Absolute Error (0.0570). These higher error values imply that the Decision Tree model overfitted the training data, resulting in worse generalization to the test data.

The Gradient Boosting Regression model outperformed the Decision Tree model but not the Random Forest model. It outperformed the Decision Tree model in terms of R-squared (0.668), showing a superior fit to the data. Its error metrics, however, were greater than those of the Random Forest model. When compared to the Random Forest model, the Gradient Boosting model exhibited a greater Mean Squared Error (0.0437), Root Mean Squared Error (0.209), and Mean Absolute Error (0.1086). Overall, the Random Forest Regression model looks to be the better choice for this dataset based on the assessment criteria and performance comparison. It had the best R-squared and lowest error metrics, suggesting superior prediction accuracy and generalization. However, additional validation and comparison with different regression models might be performed to assure the chosen model's robustness and dependability.

Fig III: Actual Vs Predicted Plots for Random Forest

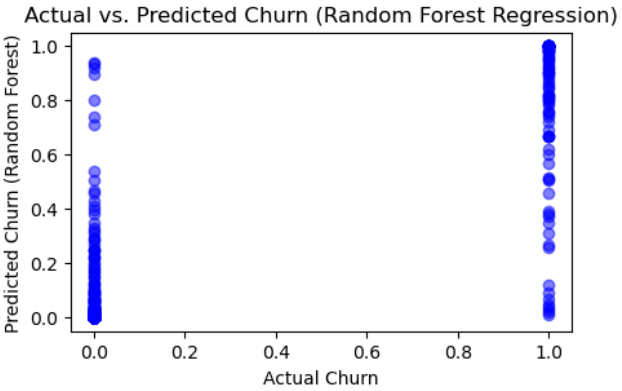


Fig IV: Actual Vs Predicted Plots for Decision Tree

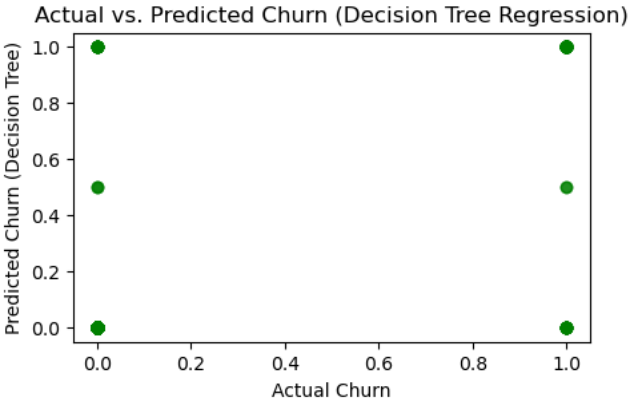
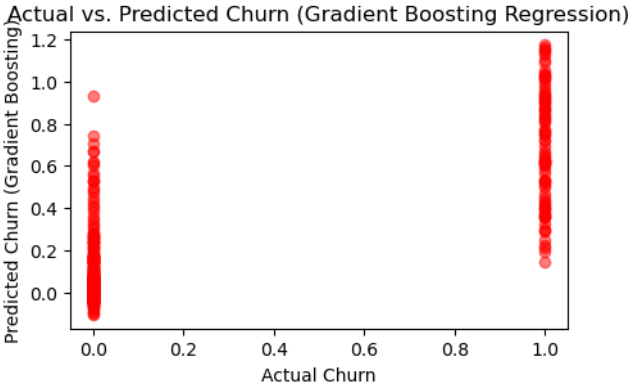
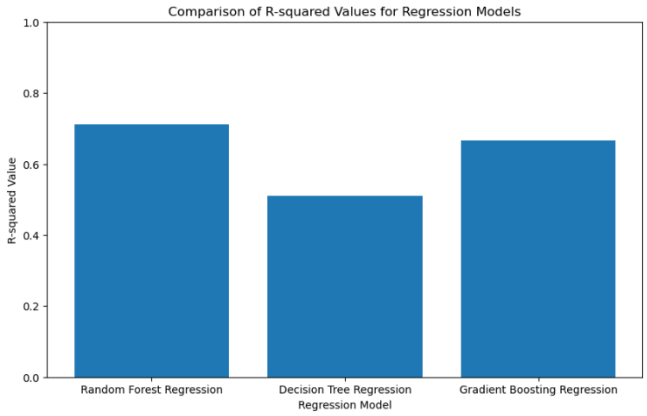


Fig V: Actual Vs Predicted for Decision Tree



Overall, the Random Forest Regression model looks to be the better choice for this dataset based on the assessment criteria and performance comparison. It had the best R-squared and lowest error metrics, suggesting superior prediction accuracy and generalization. However, additional validation and comparison with different regression models might be performed to assure the chosen model's robustness and dependability.

Fig VI: Comparison of Implemented Models



VI. DISCUSSION AND CONCLUSION

Three different regression models were employed to the Iranian Churn dataset: Random Forest Regression, Decision Tree Regression, and Gradient Boosting Regression. The R-squared statistic was used to assess the performance of these models. The Random Forest Regression model has the greatest R-squared value of among the models. This implies that the Random Forest Regression model suited the data the best. The Random Forest method mixes numerous decision trees, lowering overfitting and increasing generalization. As a result, it can capture complicated correlations in data, resulting in superior prediction performance than the Decision Tree and Gradient Boosting models. According to the findings of our investigation, all three regression models were able to capture the churn patterns in the Iranian churn dataset to some extent. However, they performed differently in terms of prediction. When compared to the other models, the Random Forest Regression model had the greatest R-squared value and the lowest MSE and MAE, indicating a better overall fit to the data and more accurate predictions. The Decision Tree Regression and Gradient Boosting Regression models performed similarly to Random Forest Regression, although with significantly greater errors.

To establish the overall performance of a regression model, it is critical to include several evaluation criteria and conduct detailed analysis. The most relevant metrics are determined. by the unique context and aims of the investigation. We have considered Mean Squared Error, Root Mean Squared Error and Mean Absolute Error as the other parameters for evaluating the model.

The findings can help telecoms firms understand the elements that influence customer turnover and build focused retention tactics. Businesses that can properly estimate customer turnover may take proactive steps to reduce churn rates, boost client retention, and eventually increase overall profitability. However, it is suggested that you explore the characteristics more, experiment with other techniques, and even include extra data to improve the model's performance.

REFERENCES

[1] *UCI Machine Learning Repository*. (n.d.). Archive.ics.uci.edu. Retrieved June 17, 2023, from https://archive.ics.uci.edu/dataset/563/iranian+churn+dataset

[2] Says, J. (2021, January 10). *A Quick Overview of Regression Algorithms in Machine Learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/01/a-quick-overview-of-regression-algorithms-in-machine-learning/

[3] x Training, P. (2023, January 13). *7 Regression Algorithms Used in Python for Machine Learning*. Pierian Training. https://pieriantraining.com/7-machine-learning-regression-algorithms-python/

[4] Maneesha Rajaratne. (2018, December 2). *Data Pre Processing Techniques You Should Know - Towards Data Science*. Medium; Towards Data Science. https://medium.com/towards-data-science/data-pre-processing-techniques-you-should-know-8954662716d6

[5] *6.3. Preprocessing data — scikit-learn 0.22.2 documentation*. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing

VII APPENDIX

Original Code and Data set can be accessed using the following link from GitHub.

https://github.com/pandatha/7072CEM---Machine-Learning.git