An

INTERNSHIP REPORT ON

"Telecom Churn Prediction"

SUBMITTED TO THE INFOSYS LIMITED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE CERTIFICATE

OF

SPRINGBOARD INTERNSHIP

SUBMITTED BY

Mr.:- Atharav Uttam Jagtap **Email :- atharavjagtap11@gmail.com**

Area of Internship:- Pune (Online) **Class :- Third Year**

Infosys Internal Rd, Konappana Agrahara,
Electronic City, Bengaluru, Karnataka 560100

2024

ACKNOWLEDGEMENT

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

Working in the **Infosys Limited** was interesting. During these one months of internship, Ilearnt a lot in Data Visualization and Machine Learning, especially completing milestones are helpful for understanding all concepts in Machine Learning.

I have to thank **Infosys Limited** for giving opportunity and platform for internship.

Therefore, I am grateful to the people in the **Infosys Limited** for the chance to make this experiment. And Opportunity to build a project and that very helpful to my knowledge.

Further on, I want to thank the mentors and interns in the **Infosys Limited** who made this demanding time joyful but always efficient.

I am extremely great full to my Mentor **Mr. Bhaskar Sir** who Guided and helped me in successful completion of this internship.

- Atharav Uttam Jagtap

ABSTRACT

This internship report presents an overview of my experience working as Data Visualization and Machine Learning Intern at **Infosys Limited** . During my internship, I was able to plan and develop ML Model using modern technologies, including Python, Jupyter Notebook, Python Libraries.

The report highlights the various milestones I worked on during my internship, including building a real-time model. Additionally, it discusses the technical skills I gained, such as Python development and project management.

Throughout the internship, I was able to learn from and collaborate with experienced developers , who provided valuable feedback and guidance on my work. I also developed strong communication and team collaboration skills, working closely with my colleagues to ensure project success.

Overall, this internship has provided me with valuable real-world experience in Data Visualization and has equipped me with the skills and knowledge necessary to pursue a career in the field.

CONTENTS

ABOUT COMPANY

Infosys Limited is an Indian multinational information technology company that provides business consulting, information technology and outsourcing services. The company was founded in Pune and is headquartered in Bangalore. Infosys is the second-largest Indian IT company, after Tata Consultancy Services, by 2020 revenue figures.

Infosys was founded by seven engineers in Pune, Maharashtra, India, with an initial capital of $250. It was registered as Infosys Consultants Private Limited on 2 July 1981. In 1983, it relocated to Bangalore, Karnataka.The company changed its name to Infosys Technologies Private Limited in April 1992 and to Infosys Technologies Limited when it became a public limited company in June 1992. It was renamed Infosys Limited in June 2011.

Infosys shares were listed on the Nasdaq stock exchange in 1999 as American depositary receipts (ADR). It became the first Indian company to be listed on Nasdaq.[citation needed] The share price surged to ₹8,100 (equivalent to ₹35,000 or US$440 in 2023) by 1999, making it the costliest share on the market at the time. At that time, Infosys was among the 20 biggest companies by market capitalization on the Nasdaq. The ADR listing was shifted from Nasdaq to NYSE Euronext to give European investors better access to the company's shares.

In July 2010, then-British Prime Minister David Cameron visited Infosys HQ in Bangalore and addressed Infosys employees. Infosys, Bangalore In 2012, Infosys announced a new office in Milwaukee, Wisconsin, to serve Harley-Davidson. Infosys hired 1,200 United States employees in 2011 and expanded the workforce by 2,000 employees in 2012. In April 2018, Infosys announced expansion in Indianapolis, Indiana.

In July 2014, Infosys started a product subsidiary called EdgeVerve Systems, focusing on enterprise software products for business operations, customer service, procurement and commerce network domains. In August 2015, assets from Finacle Global Banking Solutions were transferred from Infosys, thus becoming part of the product company EdgeVerve Systems' product portfolio.

Its annual revenue surpassed US$10 million in FY 1995, US$100 million in FY 1999, US$1 billion in FY 2004, and US$10 billion in FY 2017. Its most up to date report, as of December 2023, shows US$18 bill

INTRODUCTION

I am pleased to present this report on my one-month internship in Data Visualization. During my internship, I worked on a project that involved creating a Machine Learning model for the Telecom Churn Prediction.

A Data Visualization internship is an exciting opportunity for individuals interested in gaining practical experience in the field of Machine Learning. It provides hands-on training and exposure to real-world projects, allowing interns to apply their knowledge and skills in a professional setting.

During my internship, I learn to create the model which can predict the telecom churn based on the data.

They may be involved in various stages of the development process, including data collection, data cleaning , data transformation , model designing, model building, model testing, and deployment.

During my internship I work with a range of technologies and programming languages commonly used in Machine Learning, such as Python along with its libraries like pandas, numpy , matplotlib , seaborn and sklearn for machine learning models.

In addition to technical skills, this internship can help me develop important professional skills, such as problem-solving, communication, and time management. They will have the opportunity to collaborate with other team members, receive feedback on their work, and contribute to the overall success of Machine Learning projects.

This internship provides a unique opportunity for me to gain practical experience, develop technical skills, and enhance my professional growth in the field of Data Visualization.

PROBLEM STATEMENT

The objective of this project is to build a machine learning model that can accurately predict whether a customer is likely to churn. By analyzing historical data on customer behavior, demographics, service usage, and other relevant factors, the model will identify patterns and signals that indicate an increased risk of churn. So develop the model that can predict customer churn in the telecom industry which will enable telcos to make data-driven decisions and take predictive measures to retain their valuable customers.

MILESTONE 1 : DATA GATHERING AND ANALYSIS

Data Gathering :

Data gathering is a crucial first step in customer churn prediction. It involves collecting relevant data about customers, such as demographics, usage history, and interactions with the company. The key elements of data gathering include:

[1] Feature Engineering :
To predict churn for a particular customer, they are compared to a similar group of customers based on specific pieces of information called "features". These features include data about the customer's demographics, interaction history with the service, and other relevant information. Examples of features are the customer's age, education level, number of logins, time since last login, device type, etc.

[2] Data Sources :
Customer data can come from various sources within the company, such as the customer relationship management (CRM) system, billing records, support tickets, and usage logs. External data sources like social media and third-party data providers can also be incorporated to enrich the dataset.

[3] Data Preparation :
Once the raw data is collected, it needs to be cleaned, transformed, and formatted for analysis. This may involve handling missing values, converting data types, and creating new derived features from the raw data. Feature engineering is an important part of data preparation, where new predictive features are engineered from the raw data.

[4] Defining Churn :
One of the most critical steps in building a churn prediction model is properly defining what churn actually means for the business. The definition of churn depends on the business model and can vary widely from one company to another. For example, churn could be defined as cancelling a subscription, letting a contract lapse, or simply not using the product as often as before.

By gathering comprehensive customer data and defining churn appropriately, businesses can build accurate predictive models to identify customers at risk of leaving and take proactive measures to retain them.

Installing Required Tools :

Based on the search results, here are the key tools and steps required for building machine learning models in Python:

[1] Python Installation :
- Install the latest version of Python on your system, whether it's Windows, macOS, or Linux.

- On Windows, you can get Python from the Windows Store. On macOS, you may need to install it or update to the latest version.
- Ensure you have the necessary Python packages installed, such as NumPy, SciPy, and Matplotlib.

[2] Jupyter Notebooks :
- Jupyter Notebook is a popular tool for data exploration, model experimentation, and reporting.
- Jupyter Notebook allows you to mix code, visualizations, and markdown documentation in a single document[5].

[3] Machine Learning Libraries :
- Scikit-learn is a widely-used library for traditional machine learning tasks like classification, regression, and clustering.
- Keras is a high-level deep learning library that runs on top of TensorFlow, making it easier to build and train neural networks.

[4] Version Control :
- Use Git for version control and collaboration on your machine learning projects.
- Install Git on your system, especially if you're on Windows, to work with Git repositories.

By setting up Python, Jupyter Notebooks, and key machine learning libraries, you'll have the necessary tools to start building and training models in Python.

MILESTONE 2 : DATA PREPROCESSING

[1] Converting data types of misclassified variables :

Determine which variables in your dataset have been recorded in the wrong category or have incorrect values. Decide on the appropriate data type for each misclassified variable based on the nature of the data and the analysis requirements. Common data types include numeric, categorical, ordinal, and binary.

Replace the incorrect values with the correct ones based on the appropriate data type. This may involve converting numeric values to categorical or ordinal, or vice versa. If there are missing values in the misclassified variables, decide how to handle them, such as imputing the missing values or excluding the observations with missing data.

It is important to note that misclassification can occur due to various reasons, such as errors in data entry, incorrect categorization, or systematic biases in data collection. Addressing misclassification is crucial for ensuring the accuracy and reliability of data analysis and modeling.

[2] Removing duplicate records :

Determine which combination of columns should be used to identify duplicate records. For example, in a customer dataset, a unique record might be defined by the combination of first name, last name, and email address.Sort the data by the columns that define a unique record. This will group duplicate records together and make them easier to identify.

In Python, you can use the drop_duplicates() method on a DataFrame to remove duplicate rows. By default, it keeps the first occurrence and drops all the rest. After removing duplicates, review the data to ensure that all duplicate records have been removed and that no valid records were accidentally deleted. The key is to define what constitutes a duplicate record, identify duplicates using sorting, filtering or conditional formatting, remove the duplicates, and validate the results. Using the built-in tools in Python can greatly simplify the process of removing duplicate records.

[3] Removing unique value variables :

Removing unique value variables involves identifying and eliminating columns with only one unique value from a dataset. These columns are typically uninformative for analysis as they do not contribute to the variability of the data and do not provide meaningful insights for prediction or statistical analysis. Columns with only one unique value have zero variance, meaning they do not change across observations. Such columns do not provide any discriminatory power for predictive modeling or statistical analysis.

First, identify columns with only one unique value in your dataset. This can be done by calculating the number of unique values in each column. Use data manipulation tools like pandas in Python to filter out columns with only one unique value. For example, in Python, you can use df.nunique() to identify columns with a single unique value and then drop these columns from the dataset.

Removing columns with one unique value is part of the data cleaning process, which aims to improve the quality and relevance of the data for analysis. By eliminating uninformative variables, you streamline the dataset and focus on features that contribute meaningfully to the analysis.

[4] Removing zero variance variables :

Calculate the variance of each column and identify the ones with a variance of exactly zero. The variance of a random variable X is defined as the expected squared deviation from the mean, i.e., Var(X) = E[(X - E[X])^2]. If a column in a dataset has only one unique value, then the variance of that column will be zero. This is because the deviation of each data point from the mean will be zero, since the mean is equal to the single unique value.

Removing columns with zero variance is justified for a few reasons:

1. These columns do not contain any useful information for predictive modeling, as they have no variability. Including them in the model would not improve its performance.

2. Removing zero variance columns can help reduce the dimensionality of the dataset, which can improve the efficiency and generalization performance of machine learning models, especially when the dataset has a large number of features.

In summary, removing zero variance columns is a recommended preprocessing step in data analysis and machine learning, as it helps improve the quality and efficiency of the modeling process by removing uninformative features. The theoretical justification is based on the definition of variance and the potential issues that zero variance columns can cause for certain algorithms.

[5] Outlier treatment :

The first step is to identify the outliers in your dataset. There are several methods to detect outliers:

- Univariate method : Look for data points with extreme values on one variable
- Multivariate method : Look for unusual combinations of all the variables
- Z-score : Calculate the standard deviation of the data points and identify outliers as those with Z-scores exceeding a certain threshold (typically 3 or -3)
- Interquartile Range (IQR) : Identify outliers as data points falling outside the range defined by Q1-1.5 IQR and Q3+1.5 IQR
- Distance from the mean : Calculate the Euclidean distance of the data points from their mean and convert the distances into absolute z-scores. Any z-score greater than a pre-specified cut-off is considered an outlier.

Once you have identified the outliers, you can treat them as following:

Remove the outliers from the dataset before training the model. Transform the outliers to reduce their influence, such as capping the values at a certain percentile. Use algorithms that are less sensitive to outliers, such as robust regression, M-estimators, or outlier-insensitive clustering algorithms. Explicitly model the outliers as a separate group by adding a separate feature or using a mixture model.

After treating the outliers, validate the results by checking if the outliers have been effectively handled and that the treatment did not introduce any new issues in the data.

Treating outliers is an important step in data preprocessing for machine learning. By identifying and handling outliers appropriately, you can improve the accuracy and robustness of your machine learning models.

[6] Missing value treatment :

Missing data is a common issue when working with real-world datasets. Data can be missing for various reasons, such as sensor failure, improper data management, or

human error. Missing data can occur as single values, multiple values within one feature, or entire features may be missing.

It is important to identify and handle missing data appropriately before performing further data analysis or machine learning. Many algorithms cannot handle missing data and require entire rows with missing values to be deleted or replaced (imputed) with new values.

There are several methods to handle missing data in Python:

1. Dropping rows or columns with missing values :
   - Dropping rows with at least one null value is known as listwise deletion.
   - Dropping columns with missing values can be useful if a column is badly affected by missing data.

2. Filling missing values with a fixed value :
   - This method replaces missing values with a specified constant value.
   - However, using a fixed value may not be appropriate if the value is a legitimate data point.

3. Forward and backward filling :
   - Forward filling (ffill) propagates the last valid value forward until the next valid value is encountered.
   - Backward filling (bfill) propagates the next valid value backward until the previous valid value is encountered.

4. Filling with mean, median, or mode :
   - For small amounts of missing data, the mean or median of the existing observations can be used to replace missing values.
   - However, when there are many missing values, using the mean or median can result in a loss of variation in the data.

5. Interpolation :
   - Linear interpolation approximates missing values using two known values and their distance from the unknown point.
   - Seasonal adjustment with linear interpolation can be used for data with both trend and seasonality characteristics.

The choice of method depends on the nature of the missing data and the specific requirements of the analysis or model. It is important to understand why the data is missing, as it can help determine the most appropriate handling technique.

[7] Removing highly correlated variables :

Removing highly correlated variables is a common practice in machine learning to avoid multicollinearity, which can lead to unstable models and poor predictions. Here are the key steps to remove highly correlated variables:

1. Calculate the correlation matrix : Calculate the correlation matrix of your dataset using a function like corr() in Python.

2. Set a correlation threshold : Decide on a correlation threshold above which you consider two variables to be highly correlated. A common threshold is 0.99.

3. Identify highly correlated variables : Identify pairs of variables with a correlation above the threshold.

4. Remove one of the correlated variables : Remove one of the correlated variables from the dataset. The choice of which variable to remove can be based on domain knowledge or by selecting the variable with the lower variance.

Remember to evaluate the impact of removing correlated variables on your model's performance and adjust your approach as needed.

[8] Multicollinearity :

Based on the search results provided, here is a concise answer on handling multicollinearity using the variance inflation factor (VIF):

Multicollinearity occurs when there is a high degree of linear correlation between the predictor variables in a regression model. This can lead to unstable and unreliable regression coefficients, as the variance of the coefficients becomes inflated.

The variance inflation factor (VIF) is a widely used metric to detect and quantify the severity of multicollinearity. The VIF measures how much the variance of a regression coefficient is inflated due to multicollinearity.

A VIF value greater than 5 or 10 is generally considered to indicate problematic multicollinearity. The square root of the VIF indicates how much larger the standard error of the coefficient is compared to if that predictor had no correlation with the other predictors.

To handle multicollinearity using VIF, the recommended approach is to:

1. Identify the predictor variables with high VIF values (>5 or 10).

2. Remove or drop the predictor variables that are causing the multicollinearity, as indicated by their high VIF values.
3. Refit the regression model with the remaining predictors that have acceptable VIF values (<5 or 10).

The decision on which predictors to remove should be based on both statistical and practical considerations, such as the strength of the correlation between predictors and the ease of measuring the predictor variables.

By addressing multicollinearity using the VIF, the reliability and interpretability of the regression model can be improved.

MILESTONE 3 : DECISION TREE MODEL BUILDING

A Decision Tree is a popular supervised machine learning algorithm used for both classification and regression tasks. In Python, building a Decision Tree model involves several key steps:

1. Data Preparation :

- Import the necessary libraries like pandas for data manipulation and scikit-learn for machine learning algorithms.
- Load the dataset using pandas' read_csv() function.
- Display the top rows of the dataset to understand its structure.

2. Data Preprocessing :

- Separate the independent variables (features) from the dependent variable (target) using slicing methods.
- Convert categorical variables into numerical values if needed using techniques like mapping.

3. Splitting Data :

- Divide the data into training and testing sets using train_test_split() from scikit-learn.

4. Model Training :

- Create an instance of the Decision Tree classifier or regressor using DecisionTreeClassifier() or DecisionTreeRegressor() from scikit-learn.
- Fit the model on the training data using the fit() method.
- Training data : 80 % of dataset
- Testing data : 20 % of dataset

5. Visualization :

- Plot the decision tree model to visualize the decision-making process using plot_tree() from scikit-learn.
- Interpret the decision tree structure to understand how the model makes predictions based on the features.

6. Model Evaluation :

- Assess the model's performance using metrics like accuracy score, precision, recall, and F1-score.

- Use classification report to evaluate the model's performance on the test data.

- Performance of the model on Training data : **85.63** %

- Performance of the model on Testing Data : **77.24** %

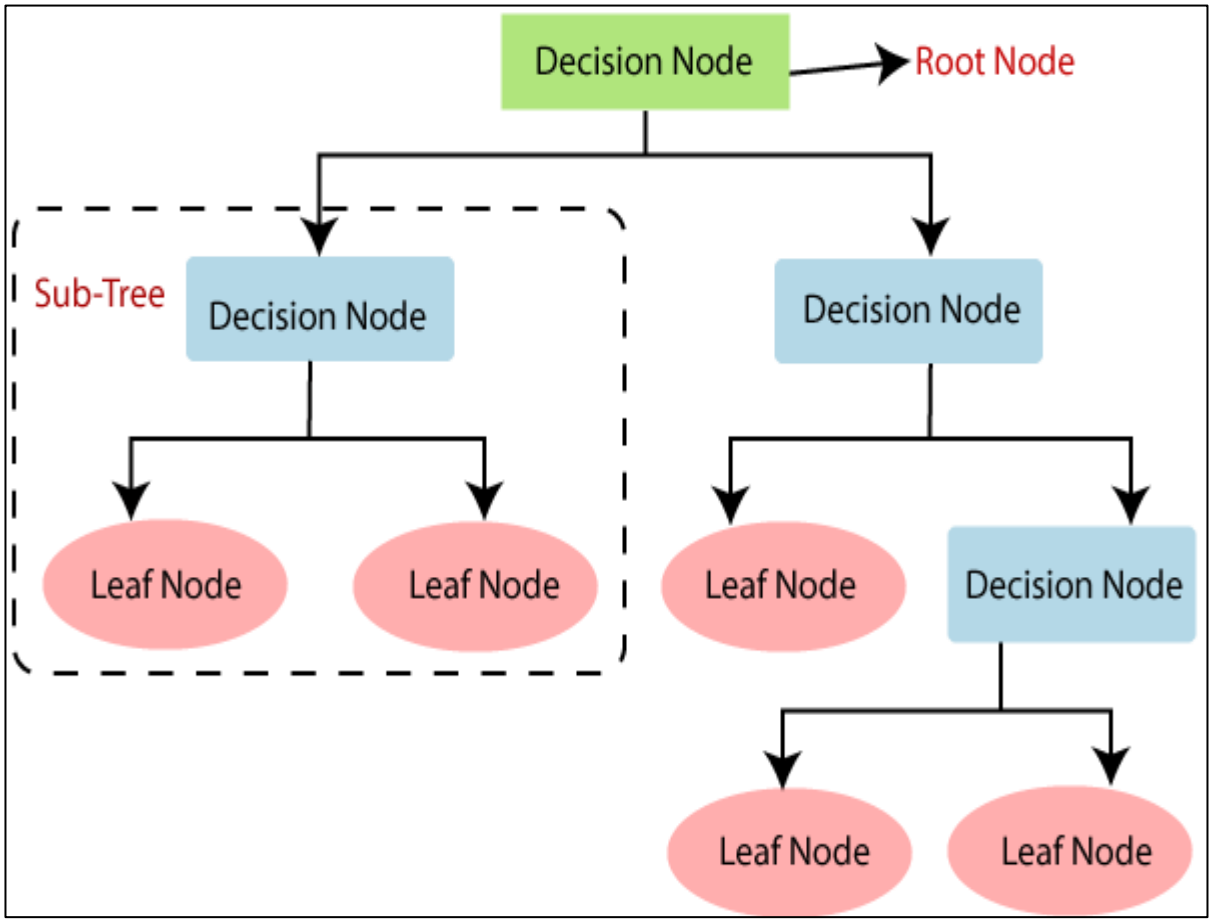
7. Hyperparameter Tuning (Optional):

- Fine-tune the model by adjusting hyperparameters like maximum depth, criterion, and minimum samples per leaf using techniques like GridSearchCV.

8. Interpretation :

- Understand the decision tree's nodes, branches, and leaf nodes to interpret how the model makes decisions based on the input features.

- Analyze the feature importance to identify which features have the most significant impact on the predictions.



In summary, building a Decision Tree model in Python involves data preparation, model training, visualization, evaluation, and interpretation. Decision Trees are known for their simplicity, interpretability, and ability to handle both numerical and categorical data effectively, making them a valuable tool in machine learning for various applications.

MILESTONE 4 : WORKING ON OTHER MODELS AND COMPARING ACCURACY

[A] Logistics Regression :

Logistic regression is a popular machine learning algorithm used for binary classification tasks. It aims to predict the probability of a binary outcome based on one or more predictor variables.

Key concepts of logistic regression include:

1. Sigmoid Function : The main function that ensures outputs are between 0 and 1 by converting a linear combination of input data into probabilities.

2. Hypothesis Function : Uses the sigmoid function and weights (coefficients) to combine input features to estimate the likelihood of falling into a particular class.

3. Log Loss : The optimization cost function is a measure of the discrepancy between actual class labels and projected probability.

4. Decision Boundary : The surface or line used to divide instances into several classes according to the determined probability.

5. Probability Threshold : A number (usually 0.5) that is used to calculate the class assignment using the probabilities that are anticipated.

To implement logistic regression in Python, you can use libraries like scikit-learn. The general steps are:

1. Import necessary libraries and load the dataset.
2. Preprocess the data, including handling missing values and encoding categorical variables.
3. Split the data into training and testing sets.
4. Create an instance of the LogisticRegression class and fit the model on the training data.
5. Evaluate the model's performance using metrics like accuracy, confusion matrix, and classification report.

Logistic regression has several assumptions:

- The dependent variable must be binary.
- The independent variables should be linearly related to the log odds.
- There should be little or no multicollinearity among the independent variables.
- The dataset should have a large sample size.

Logistic regression is widely used in various applications, such as predicting customer churn, credit card fraud detection, and medical diagnosis.

[B] Random Forest :

A Random Forest Classifier is a supervised machine learning algorithm that combines multiple decision trees to improve the accuracy and robustness of classification tasks. Here is a detailed overview of the concept:

A Random Forest Classifier is an ensemble learning method that creates a set of decision trees from a randomly selected subset of the training set. Each decision tree is constructed using a random subset of features and then the outputs from all trees are combined to make the final prediction. This approach helps to reduce overfitting and improve the overall performance of the model.

Working :

1. Data Preparation : The dataset is split into training and testing sets.

2. Tree Construction : Each decision tree is built using a random subset of features and a random subset of the training data. The trees are constructed recursively by splitting the data based on the best feature and the best threshold for that feature.

3. Combining Trees : The outputs from all trees are combined to make the final prediction. The class with the most votes from all trees is chosen as the predicted class.

Advantages :

1. Improved Accuracy : Random Forests can handle complex datasets and provide high accuracy even when individual trees are not accurate.

2. Robustness : The ensemble nature of the model makes it less prone to overfitting and more robust to noise and outliers.

3. Feature Importance : Random Forests can provide feature importance scores, which help in understanding the relative importance of each feature in the model.

In Python, you can use libraries like scikit-learn to implement a Random Forest Classifier. The general steps include data preprocessing, model training, and model evaluation.

HARDWARE & SOFTWARE USED

- Operating System – Windows11
- Anaconda Jupyter

CONCLUSION

The evaluation results indicate that decision tree models have been widely used and have shown strong performance in predicting customer churn in the telecommunications industry. Decision trees have consistently outperformed other techniques like logistic regression, neural networks, and SVM in terms of predictive accuracy for churn. Studies have reported decision tree models achieving very high predictive accuracy, up to 98.88% in one case.

In summary, the search results demonstrate that decision tree models are a highly effective and preferred technique for predicting customer churn in the telecommunications industry. Their interpretability, flexibility, and strong predictive performance make them a valuable tool for telecom companies looking to proactively manage customer attrition.