Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/20305
DOI URL: http://dx.doi.org/10.21474/IJAR01/20305

RESEARCH ARTICLE

CUSTOMER CHURN PREDICTION AND CATEGORIZATION A MACHINE LEARNING APPROACH TO ANALYSE CUSTOMER BEHAVIOR AND DECISION MAKING IN THE TELECOMMUNICATIONS INDUSTRY

Ishaan Gangwani[1], Sumedh Jadhav[1] and Mustafa Saifee[2]

1. Indus International School Pune.
2. Carnegie Mellon University.

Manuscript Info

Manuscript History
Received: 25 November 2024
Final Accepted: 28 December 2024
Published: January 2025

Abstract

Customer churn remains a critical challenge for subscription- based businesses, particularly in the telecommunications indus- try, where retaining customers is significantly more cost-effective than acquiring new ones. This study leverages machine learn- ing to develop a robust churn prediction framework and identify key behavioral drivers of churn. Using the Telco customer churn dataset, we employ an ensemble Voting Classifier composed of Logistic Regression, Random Forest, XGBoost, and CatBoost models. The ensemble achieves a high predictive accuracy, with an AUC of 0.98, effectively distinguishing between churned and retained customers. Beyond prediction, the study introduces a structured catego- rization of churn reasons into four primary classes-Attitude and Expertise, Service and Product Issues, Competitor and Price, and Other Reasons. A multi-class classification model using XGBoost achieves an accuracy of 0.63, outperforming random guess baselines. The categorization reveals that factors such as short-term contracts, lack of technical support, and high monthly charges are significant contributors to churn. Conversely, long- term contracts and automated payment methods demonstrate strong retention effects. The findings provide actionable insights into customer behav- ior and decision-making, emphasizing the importance of improv- ing technical support, addressing cost concerns, and creating in long-term commitments. By combining predictive accuracy with interpretability, this study enables targeted retention strategies to minimize churn and enhance customer lifetime value.

Introduction:-

Customer churn, defined as the rate at which customers discontinue their subscriptions to a service, represents a significant challenge for subscription-based businesses, particu- larly in the telecommunications (Telco) industry. Mathematically, churn rate C can be expressed as:

Corresponding Author:- Ishaan Gangwani
Address:- Indus International School Pune.

$$C = \frac{L}{T} \quad (1)$$

where L denotes the number of lost customers during a specific time period, and T represents the total customer base at the start of that period. A high churn rate directly correlates with reduced revenue, increased customer acquisition costs, and diminished customer lifetime value (CLV).[3] Research indicates that the cost of acquiring a new customer is approximately five times greater than retaining an existing one.

The core problem addressed in this study is the identification and classification of churn reasons using a data-driven, machine-learning approach[4]. In essence, the task is to map each churn instance i to a specific reason R_i, such that:

$$R_i \in \{R_1, R_2, \ldots, R_n\} \quad (2)$$

where $\{R_1, R_2, \ldots, R_n\}$ represents a predefined set of churn categories (e.g., pricing dissatisfaction, service quality issues, or contract concerns). This classification facili- tates targeted retention strategies, thereby reducing C and improving overall business performance.

To perform this analysis, we sourced the Telco customer churn dataset from the IBM community platform, https://community.ibm.com/community/user/businessanalytics/ blogs/steven-macko/2019/07/11/telco-customer-churn-1113. The dataset provides comprehensive details about customer demographics, account information, and service us- age, which serve as essential features for understanding and predicting churn behavior.

By leveraging this dataset, we aim to build robust machine-learning models that uncover key insights and enable actionable interventions for reducing churn.

Research Objectives:-

1. Churn Prediction: Formulate a supervised learning problem where the target vari- able Y indicates whether a customer will churn ($Y = 1$) or remain ($Y = 0$). The predictor variables X include contract details, payment methods, and demographic factors.
2. Feature Engineering: Identify key explanatory variables $X_1, X_2, \ldots, X_k$ through techniques like correlation analysis, mutual information, and feature importance scores derived from ensemble models such as XGBoost.
3. Categorization of Churn Reasons: Develop a multi-class classification framework to assign each churn instance i to a churn category R_i. This process leverages clustering and decision-tree-based feature importance to improve interpretability.
4. Retention Strategy Design: Translate the categorized churn reasons R_i into action- able business strategies, aimed at minimizing the probability $P(Y = 1 \mid X)$ for future customers.

Methodology:-

Data Preprocessing

The Telco customer churn dataset consists of $m = 7043$ observations (customers) and $n = 21$ original features representing demographic, account-level, and service usage data. Let $X \in R^{m \times n}$ be the feature matrix and $y \in \{0, 1\}^m$ the binary target variable, where $y_i = 1$ indicates churn and $y_i = 0$ indicates no churn. Preprocessing addresses missing values, feature scaling, and categorical encoding to create a final input space $X_{final} \in R^{m \times n'}$, where $n' > n$.

[3]Kline, Rex B. Principles and Practice of Structural Equation Modeling. 5th ed., Guilford Press, 2011.

[4]dwards, A. L. An Introduction to Linear Regression and Correlation. W.H. Freeman, 1976.

Handling Missing Values. The TotalCharges feature contains missing values for customers with Tenure = 0, corresponding to new subscribers. Missing values are deter- ministic and imputed using[5]:

$$X_{\text{TotalCharges},i} = \begin{cases} X_{\text{MonthlyCharges},i} \cdot X_{\text{Tenure},i}, & \text{if } X_{\text{Tenure},i} = 0 \wedge X_{\text{TotalCharges},i} = \text{NaN}, \\ X_{\text{TotalCharges},i}, & \text{otherwise.} \end{cases} \tag{3}$$

Other features with missing entries are evaluated based on the missing rate[6]:

$$\text{MissingRate}_j = \frac{M_j}{m}, \tag{4}$$

where M_j is the count of missing values for feature X_j. Features are dropped if $\text{MissingRate}_j > \theta$, where $\theta = 5\%$. For continuous features below this threshold, missing values are replaced using mean imputation[7]:

$$X_{ij} = \frac{1}{m - M_j} \sum_{k \in O_j} X_{kj}, \quad \forall X_{ij} \in \text{NaN}, \tag{5}$$

where O_j is the set of observed (non-missing) entries.

Feature Scaling. Continuous features such as MonthlyCharges, TotalCharges, and Tenure are scaled using Min-Max Normalization to ensure uniform feature contribution[8]. The transformation for a feature X_j is defined as[9]:

$$X_{ij}^{\text{scaled}} = \frac{X_{ij} - \min(X_j)}{\max(X_j) - \min(X_j)}, \quad \forall i \in [1, m]. \tag{6}$$

Categorical Feature Encoding. Binary categorical features such as Partner and Dependents are encoded as[10]:

$$X_{ij} = \begin{cases} 1, & \text{if } X_j = \text{Yes}, \\ 0, & \text{if } X_j = \text{No}. \end{cases} \tag{7}$$

For multi-class features like Contract with k = 3 classes, One-Hot Encoding (OHE) generates k binary variables $X_{j1}, X_{j2}, \ldots, X_{jk}$:

$$X_{j\ell} = \begin{cases} 1, & \text{if } X_j = \ell, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

After encoding, the expanded feature space n′ captures class-specific relationships, ensuring compatibility with downstream models.

[5]Galli, Soledad. Python Feature Engineering Cookbook: Over 70 Recipes for Creating, Engineering, and Transforming Features. Packt Publishing, 2020.

[6]Ozdemir, Sinan, and Divya Susarla. Feature Engineering Made Easy. Packt Publishing, 2018.

[7]Kuhn, Max, and Kjell Johnson. Feature Engineering and Selection: A Practical Approach for Predictive Models. Chapman and Hall/CRC, 2020.

[8]Analytics India Magazine. "Why Data Scaling Is Important in Machine Learning." Analytics In- dia Magazine, 2021, analyticsindiamag.com/why-data-scaling-is-important-in-machine-learning-how-to-effectively-do-it/.

[9]Analytics Vidhya. "Feature Scaling in Machine Learning: Normalization and Standard- ization." Analytics Vidhya, 2020, www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine- learning-normalization-standardization/.

Exploratory Data Analysis Churn

Proportion Overview

The first plot provides an overview of the proportion of customers who churned (**Yes**) compared to those who did not churn (**No**). A significant class imbalance is observed, with the majority of customers (over 5000) not churning, while a smaller subset (around 1800) represents churned customers. This imbalance highlights that churn is a relatively rare event when compared to customer retention. Such imbalances can introduce bias in predictive models, as they may skew results toward the majority class.

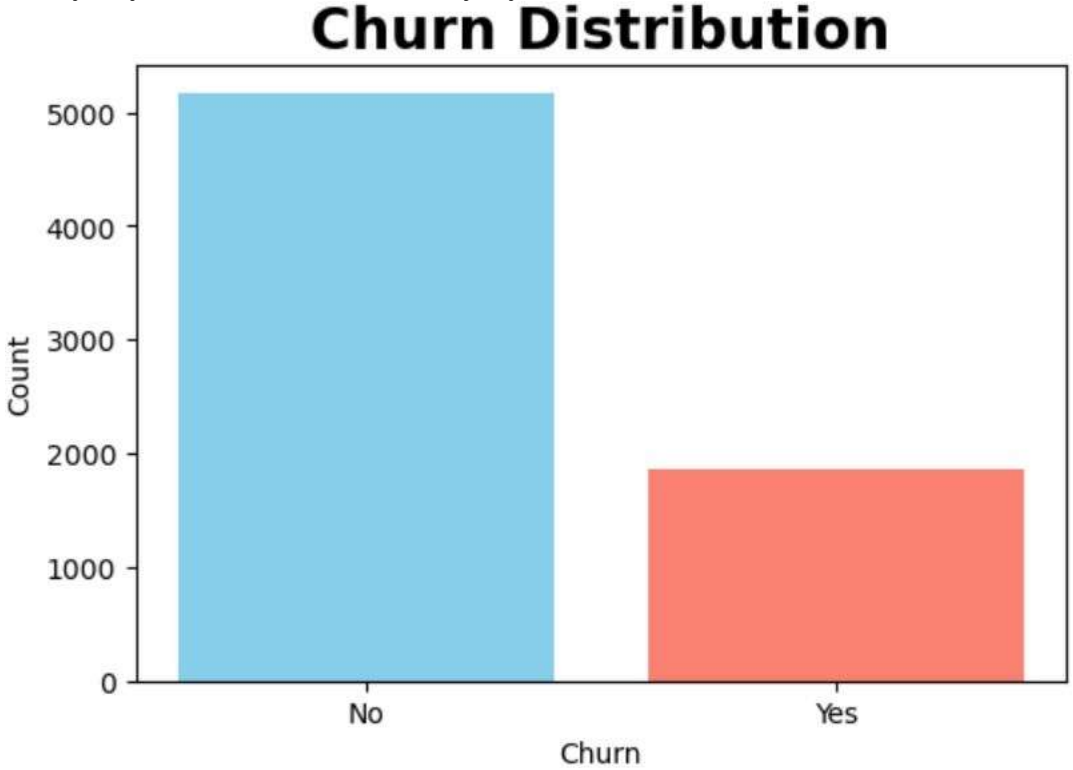

Figure 1:- Proportion of Churned vs Non-Churned Customers.

Gender-Wise Churn Distribution

The second plot analyzes gender-wise churn distribution. The data reveals that churn trends remain similar across male and female customers, where both genders show a significantly higher number of retained customers (**No**) compared to those who churned (**Yes**). This observation indicates that gender does not have a dominant influence on churn behavior. However, it is essential to validate this assumption during the model training phase using statistical significance tests, as some subtle gender-based differences might still exist.

[10]Géron, Aur´elien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed., O'Reilly Media, 2019.

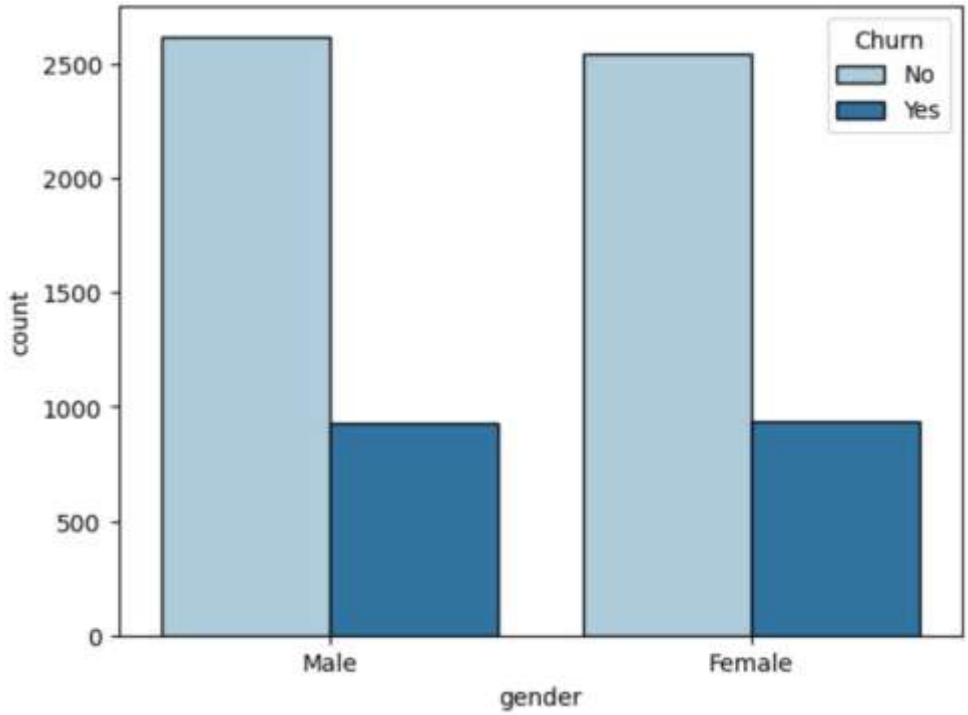

Figure 2:- Churn Distribution by Gender.

Churn by Senior Citizen Status

In the third plot, churn by senior citizen status is examined. Non-senior citizens (0) constitute the majority of the dataset and exhibit a higher retention rate, while senior citizens (1), although smaller in number, display a disproportionately higher churn rate relative to their population size. This suggests that senior citizen status is a meaningful predictor of churn. Customers who are senior citizens may require targeted retention strategies, such as personalized offers, improved customer support, or loyalty programs tailored to address their specific needs and concerns.

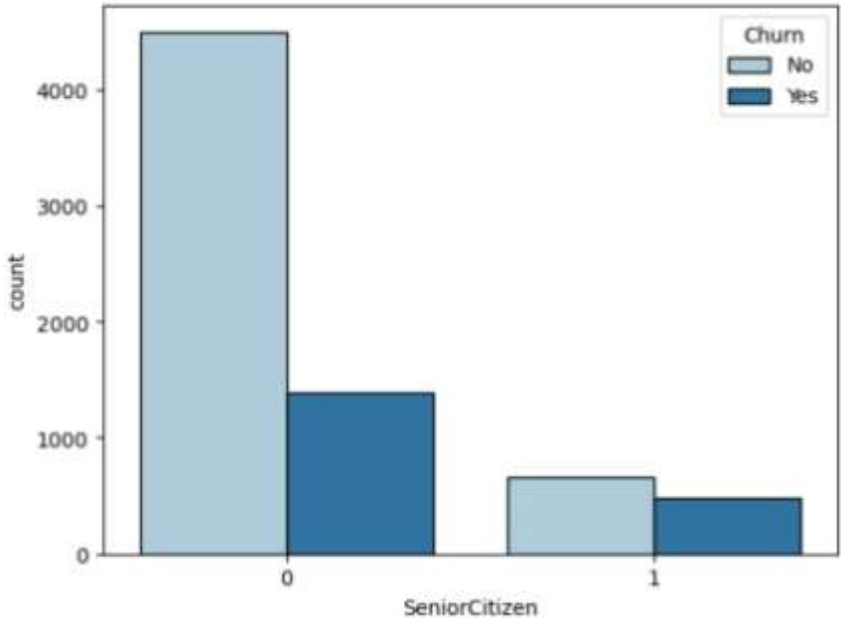

Figure 3:- Churn Distribution by Senior Citizen Status.

Churn by Partner Status

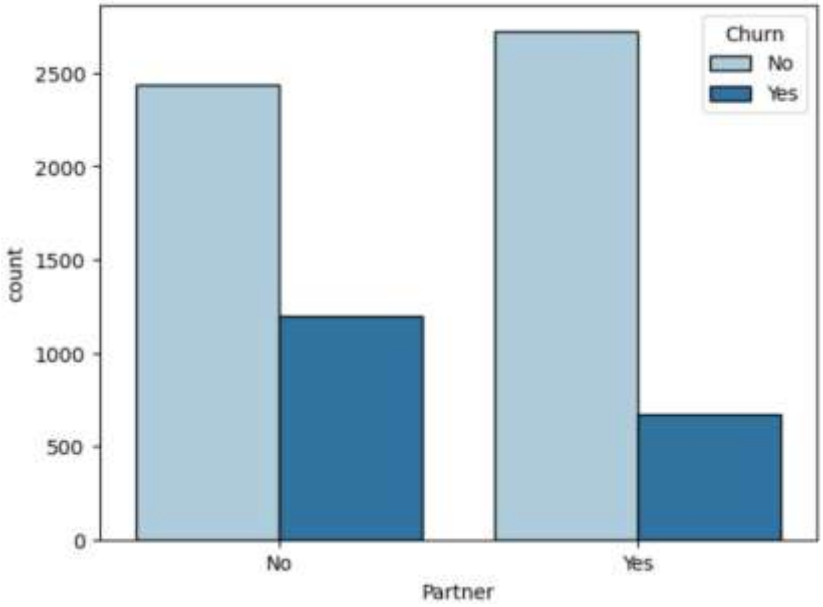

Figure 4:- Churn Distribution by Partner Status.

The fourth plot investigates churn by partner status. The findings highlight that cus- tomers without a partner (**No**) are more likely to churn compared to those with a partner (**Yes**). The churn proportion is relatively high for individuals without partners, indicat- ing that perceived stability or support associated with having a partner may influence customer retention. To address this, targeted campaigns, such as bundled offers, rela- tionship incentives, or discounts for customers without partners, could help reduce churn rates.

Churn by Dependents

The fifth plot explores churn by dependents. Customers without dependents exhibit a significantly higher churn rate than those with dependents. Customers with dependents appear to be more loyal, as evidenced by their lower churn counts. This finding suggests that dependent status may reflect a customer's long-term attachment or commitment to the service provider. Retention strategies, such as family-oriented or dependent-specific plans, could be introduced to incentivize customers without dependents to remain loyal.

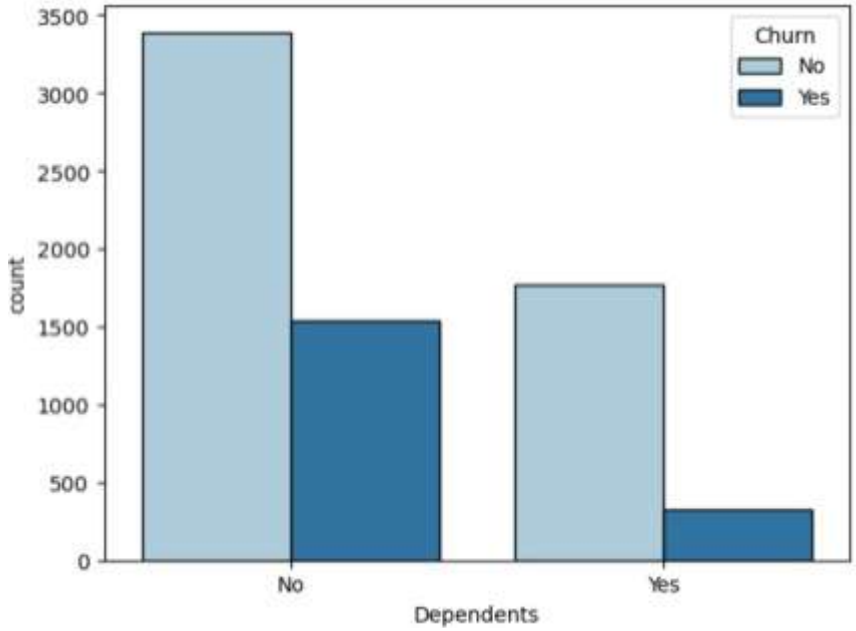

Figure 5:- Churn Distribution by Dependents.

Monthly Charges and Churn

The sixth plot illustrates the distribution of **Monthly Charges** by churn status. Cus- tomers with lower monthly charges, typically ranging between $20 and $30, are predomi- nantly retained, while churned customers tend to cluster around higher monthly charges, particularly above $70. This trend suggests that cost-sensitive customers may perceive higher monthly fees as unaffordable or insufficiently valuable. Service providers can im- plement tiered pricing models, loyalty discounts, or enhanced value-added services for high-paying customers to mitigate churn risks.

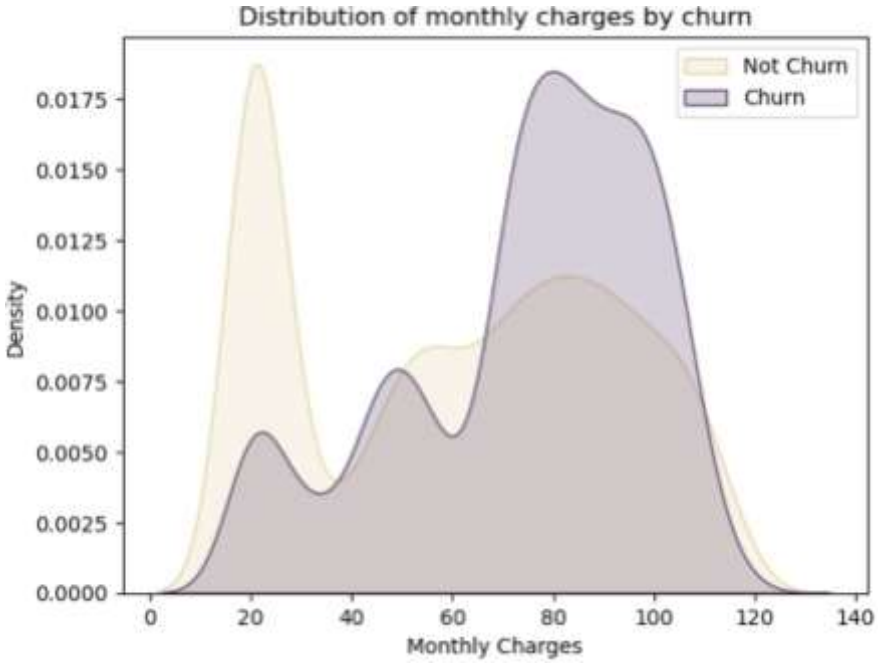

Figure 6:- Distribution of Monthly Charges by Churn Status.

K-Means Clustering Analysis

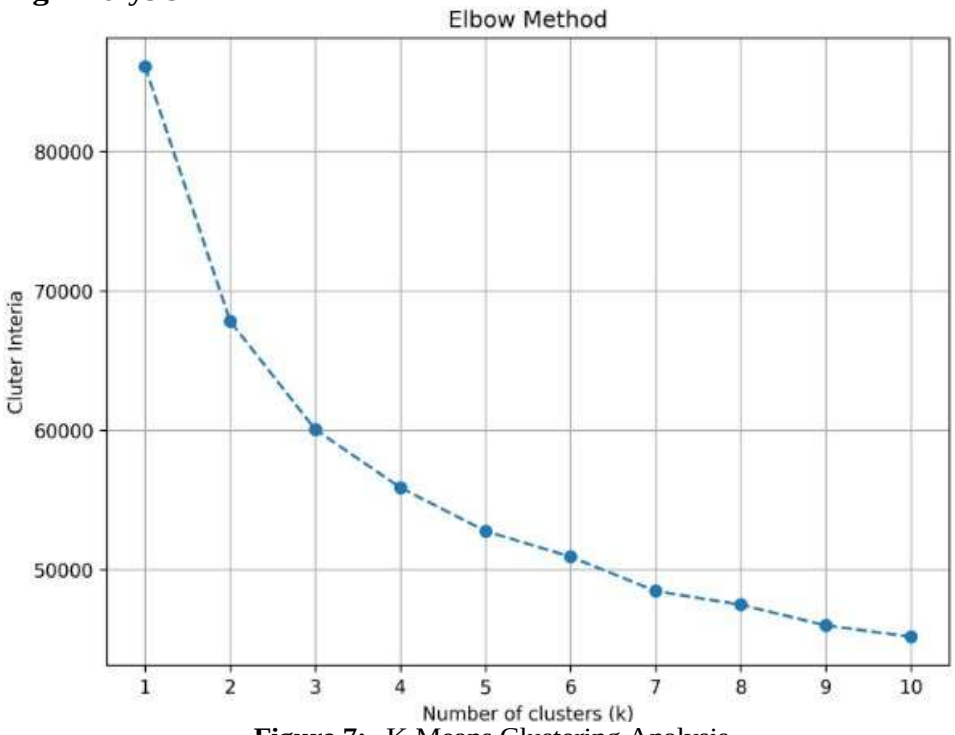

Figure 7:- K Means Clustering Analysis.

The K-Means clustering technique[11] was applied to segment the customer dataset into distinct groups. The **elbow method** was used to determine the optimal number of clus- ters by plotting the relationship between the **number of clusters (k)** and the **within- cluster sum of squares (WCSS)**. The resulting graph shows a sharp decrease in WCSS from k=1 to k=3k=3, followed by a slower decrease rate beyond this point. The **'elbow point'** occurs at k=4, where the improvement in WCSS becomes marginal with the ad- dition of more clusters. Based on this observation, **4 groups** were chosen as the optimal number, balancing the granularity of the segmentation and the interpretability.

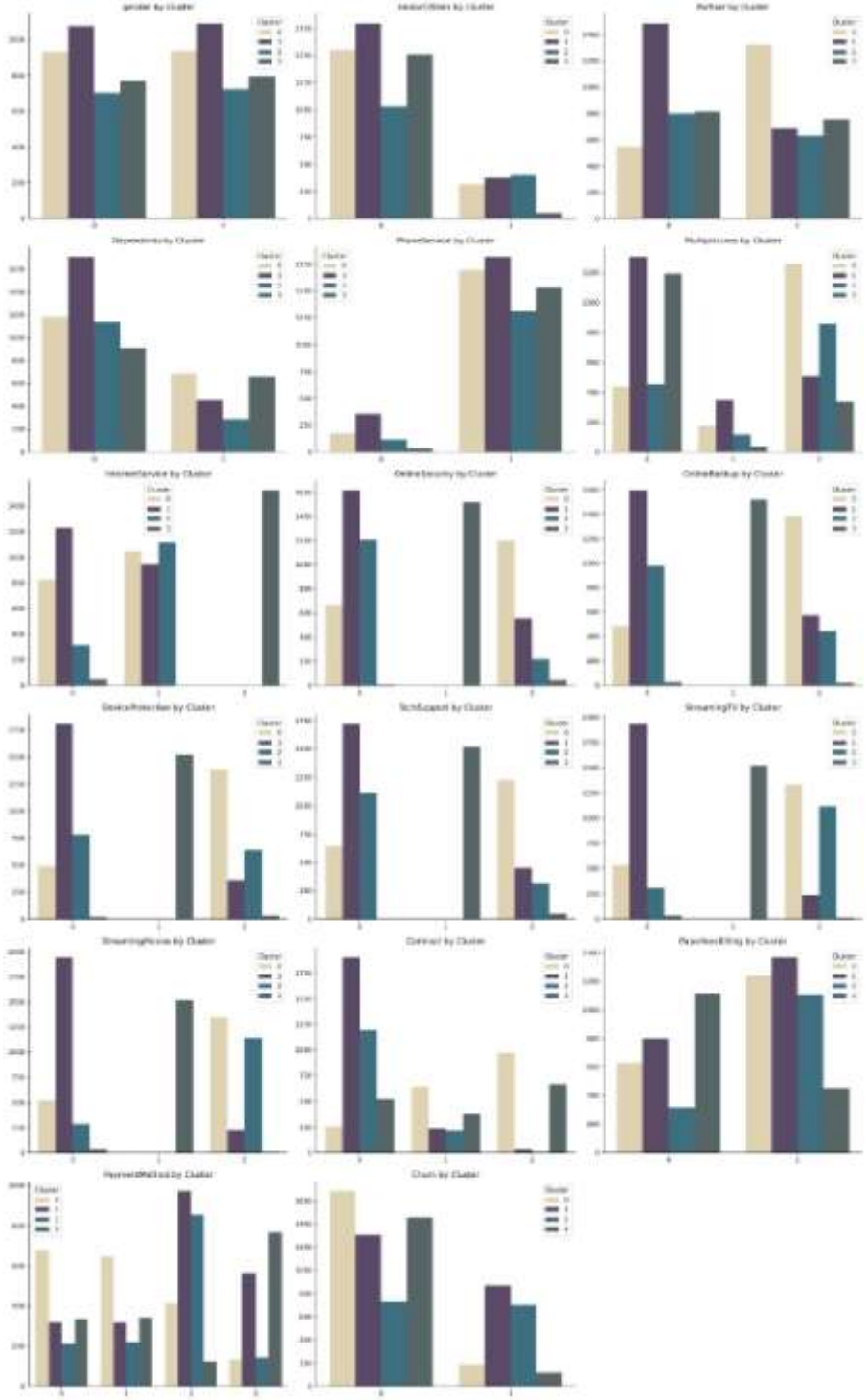

Figure 8:- Cluster Analysis Based on K Means Results for All Categories.

[11]Lloyd, Stuart P. "Least Squares Quantization in PCM." IEEE Transactions on Information Theory, vol. 28, no. 2, 1982, pp. 129–137.

Table 1:- Cluster Characteristics Overview.

Aspect	Observation
1. Gender by Cluster	The gender distribution across all clusters appears rel- atively uniform, indicating that gender does not play a significant role in differentiating between customer groups.
2. SeniorCitizen by Clus- ter	Cluster 3 contains a disproportionately high number of Senior Citizens, while Clusters 0, 1, and 2 primarily consist of younger customers. This highlights Cluster 3 as a distinct segment representing senior customers who may have unique preferences or service needs.
3. Partner and Depen- dents by Cluster	Clusters 1 and 3 show higher proportions of customers with partners, while Cluster 0 has a notable share of customers without partners or dependents. Cluster 0 may represent single, independent individuals, which aligns with higher churn tendencies often observed in this group.
4. Phone Service and Multiple Lines by Cluster	While phone service usage remains consistent across all clusters, Cluster 3 shows a higher adoption of Mul- tiple Lines, suggesting a preference for bundled ser- vices within this group.
5. Internet Services and Online Features	Cluster 3 exhibits the highest adoption of Internet Ser- vices, as well as related features such as OnlineSecu- rity, OnlineBackup, and Tech Support. In contrast, Clusters 1 and 2 show higher proportions of customers without these features, reflecting limited adoption of advanced services. Cluster 0 shows mixed usage pat- terns.
6. Streaming Services	Cluster 3 stands out with a high number of customers using StreamingTV and StreamingMovies, indicating a preference for entertainment-focused services. Clus- ter 1, on the other hand, shows lower adoption of streaming features.
7. Contract Type by Cluster	Cluster 0 has the largest proportion of customers on Month-to-Month contracts, which are often associated with higher churn rates. In contrast, Cluster 3 in- cludes a higher share of customers on long- term con- tracts (One-Year and Two-Year), reflecting greater stability and commitment.
8. Paperless Billing and Payment Methods	Cluster 0 has a significant number of customers opting for Paperless Billing and using Electronic Checks for payments, both of which are linked to higher churn. In comparison, Cluster 3 shows a preference for Bank Transfers and Credit Cards, which are typically asso- ciated with lower churn.

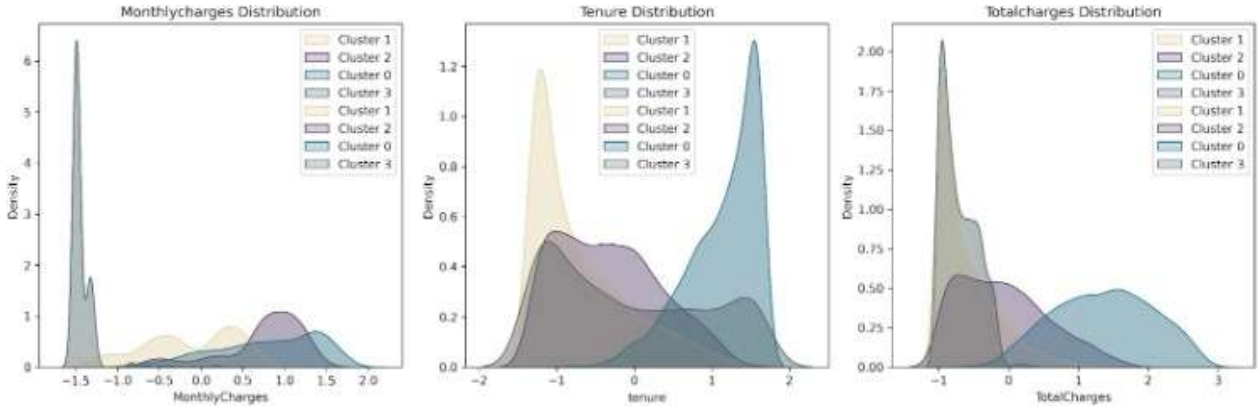

Figure 9:- Cluster Distribution Analysis.

The additional density plots for **Monthly Charges**, **Tenure**, and **Total Charges** provide valuable insight into the continuous features of the 4 identified clusters. These features highlight key behavioral differences between customer segments, helping to refine the understanding of churn patterns.

In the **Monthly Charges Distribution**, **Cluster 3** displays a high density at lower monthly charges, indicating that customers in this group are likely to be cost-sensitive and on basic plans. In contrast, **Cluster 0** exhibits a broader distribution between mid- to high monthly charges, reflecting customers with relatively higher spending, potentially for advanced service adoption. **Clusters 1** and **2** fall between these extremes, with **Cluster 1** showing a slight concentration in midrange charges. This suggests that **cluster 3** customers require strategies to encourage upsell, while **cluster 0** customers may need value reinforcement to justify their higher costs.

The **Tenure Distribution** further distinguishes the clusters. **Cluster 0** has the high- est density for longer tenures, representing loyal customers with long-term customers. On the other hand, **Cluster 3** shows the lowest tenures, concentrated around shorter dura- tions, identifying newer customers who are more prone to churn. **Cluster 1** is skewed toward shorter tenures, while **Cluster 2** reflects a mid-range tenure distribution. These observations suggest that **cluster 3** customers should be prioritized for retention ef- forts during the early stages of their lifecycle, while **Cluster 0** customers can be retained through loyalty programs that maintain their long-term satisfaction. The **Total Charges**

Distribution aligns closely with tenure and monthly charges. **Cluster 3** has a sharp peak at low cumulative charges, reflecting short tenures and lower monthly payments, which is characteristic of newer customers. Conversely, **Cluster 0** displays a broader spread at higher total charges, consistent with longer tenures and higher spending. **Clus- ters 1** and **2** are concentrated in the mid-range total charges, reflecting moderate cu- mulative revenue. These findings emphasize the need to address **Cluster 3**'s churn risks early to increase lifetime value, while **Cluster 0** requires strategies to maintain their loyalty and high revenue contribution.

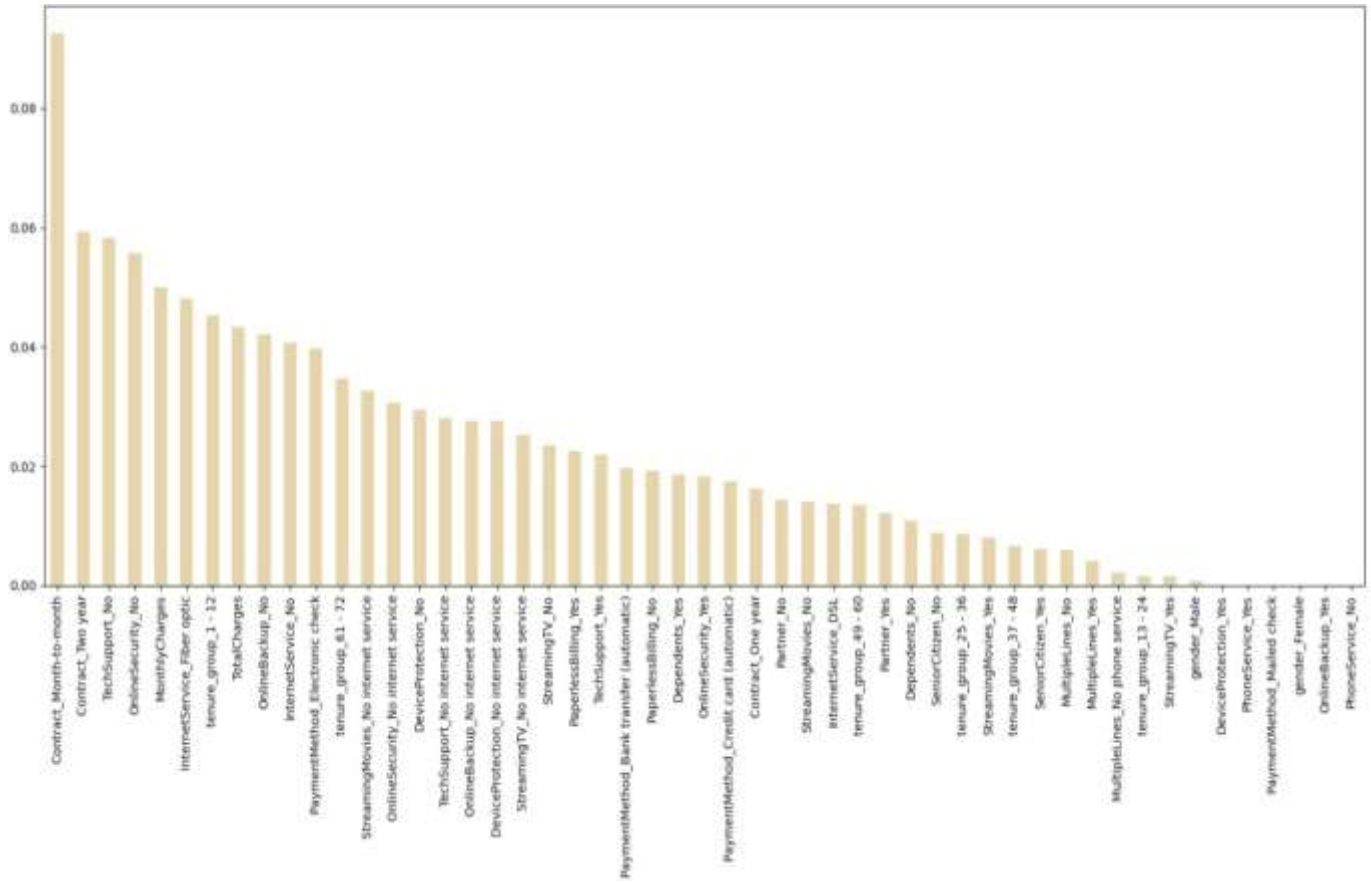

Figure 10:- Correlation between Churn and Customer Features.

Cluster	Key Characteristics	Churn Behavior
Cluster 0	Month-to-month contracts, electronic check payments, limited online fea- tures, no dependents, higher monthly and total charges, long tenure	High churn
Cluster 1	Moderate feature adoption, mid-level partner status, limited streaming ser- vices, shorter tenure, moderate charges	Moderate churn
Cluster 2	Younger customers, low adoption of ad- vanced services, more dependents, mid- range tenure, moderate charges	Moderate churn
Cluster 3	Long-term contracts, high streaming usage, bank transfer payments, senior citizens, low monthly charges, short tenure, low total charges	Low churn

Table 2:- Cluster Characteristics and Churn Behavior.

Correlation Plot Analysis

The correlation plot highlights the relationship between various customer features and churn, where the height of each bar represents the magnitude of correlation. Features with a strong positive correlation contribute significantly to churn risk, while those with a negative correlation are associated with customer retention. These insights help identify the most influential factors driving churn and retention.

Table 3:- Feature Correlation with Churn.

Feature	Correlation with Churn	Insight
Contract (Month-to-Month)	Strong Positive	Month-to-month contracts are the pri- mary driver of churn.
Tech Support (No)	Positive	Lack of technical support leads to dis- satisfaction and churn.
Monthly Charges	Positive	Higher charges increase the likelihood of customer churn.
Internet Service (Fiber)	Positive	Fiber optic services are associated with churn, likely cost-driven.
Contract (Two-Year)	Strong Negative	Long-term contracts strongly reduce churn.
Payment Method (Auto-pay)	Negative	Automatic payments are associated with higher retention.

Contract Type and Behavioral Churn Trends

The contract-based churn distribution reveals a strong relationship between contract type and churn. For churned customers, nearly 88.7% are on **month-to-month contracts**, emphasizing that short-term plans correlate with higher churn rates due to flexibility and fewer switching barriers. Non-churned customers display a more balanced distribution, with a significant proportion on **longer-term contracts** (one-year and two-year). Longer commitments foster customer stability, likely due to reduced costs and decision inertia.

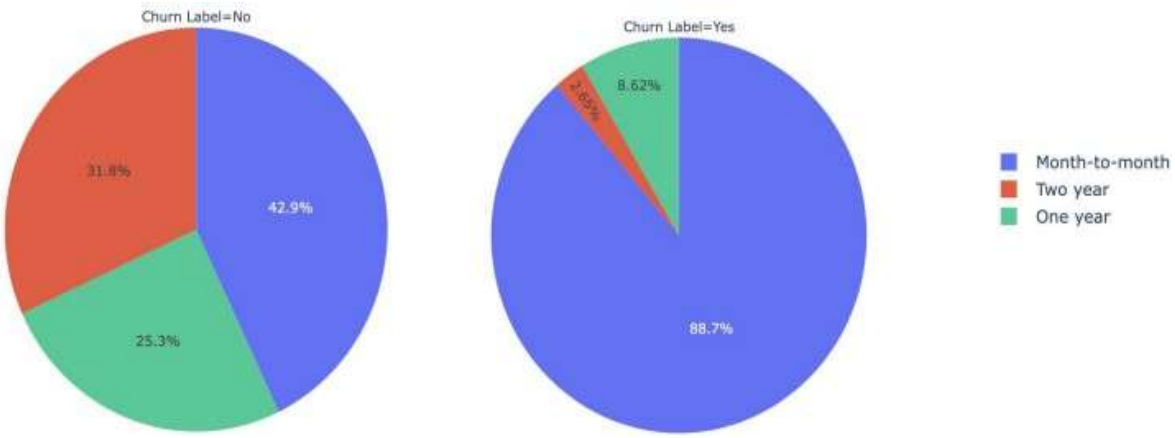

Figure 11:- Churn Behavior by Contract Type.

Billing Discrepancies and Customer Trust

Billing discrepancies, as shown in the table below, highlight inconsistencies between re- ported **Total Charges** and calculated charges for long-term customers. Month-to-month contracts exhibit minimal discrepancies, whereas discrepancies increase significantly for customers with one-year and two-year contracts, especially at higher tenures.

Table 4:- Charge Discrepancies Across Contract Types.

Contract	Churn Rate	Total Charges	Charge Difference
Month-to-month	0.50	679.55	0.00
One Year	0.90	6341.25	92.20
Two Year	0.95	7922.34	139.18

The growing discrepancies undermine customer trust, particularly among loyal, long- tenure customers, emphasizing the importance of transparent and accurate billing prac- tices to foster retention.

Geospatial Churn Analysis

The geospatial churn heatmap[12] reveals that churn is **evenly distributed across all regions** in California, with no specific area experiencing disproportionately high churn. This observation confirms that churn is not geographically driven but stems from individual- level dissatisfaction, such as service quality, billing issues, or competition.

[12]Hunter, J. D. "Matplotlib: A 2D Graphics Environment." Computing in Science Engineering, vol. 9, no. 3, 2007, pp. 90–95.

Figure 12:- Geospatial Distribution of Churn (Red represents churn rate of 1).

Internet Service and Customer Preferences

The internet service type analysis highlights significant churn differences. Fiber Optic users represent the largest proportion of churned customers, indicating dissatisfaction despite the service's technical superiority. Customers using DSL or reporting no internet service exhibit lower churn rates, suggesting fewer alternatives or lower expectations.

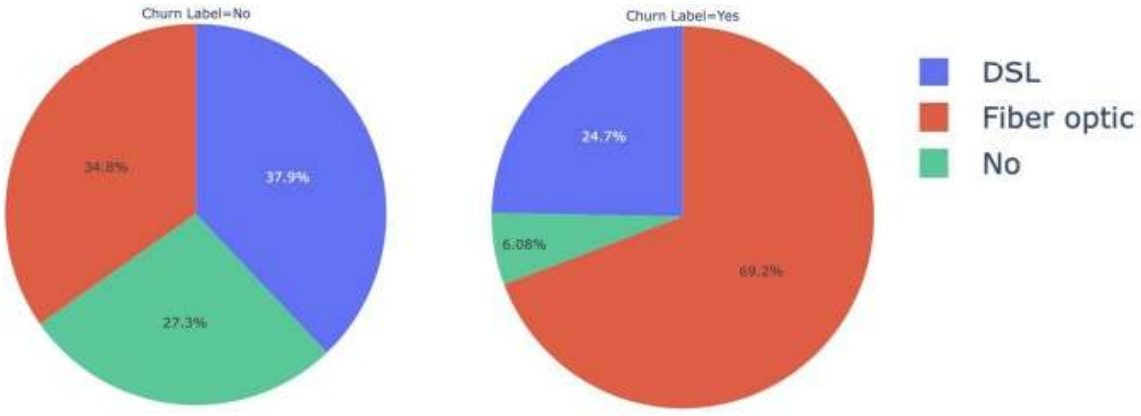

Figure 13:- Churn by Internet Service Type.

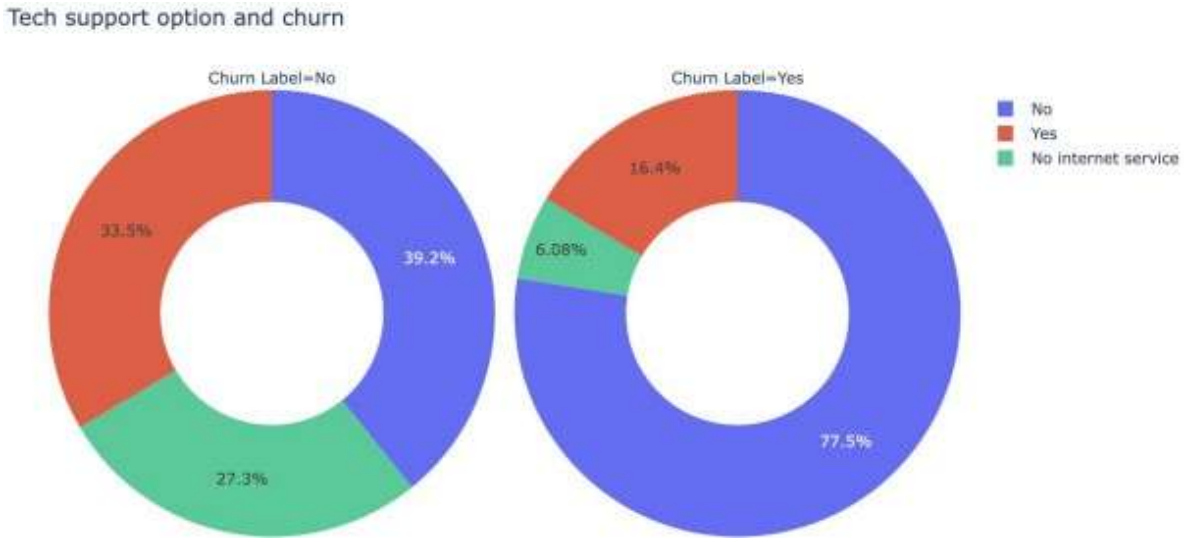

Figure 14:- Impact of Tech Support on Churn Behavior.

Technical Support and Churn Behavior

The availability of technical support strongly correlates with churn rates. Customers lacking technical support exhibit significantly higher churn, as unresolved service issues contribute to dissatisfaction. Non-churned customers display a more balanced distribu- tion, underscoring the importance of proactive support.

Churn Reason Frequency and Customer Decisions

The churn frequency bar chart identifies recurring themes driving customer decisions to churn. Top reasons include poor customer support, competitive offerings, and network reliability issues. Addressing these areas through improved support quality, pricing com- petitiveness, and service reliability can mitigate churn.

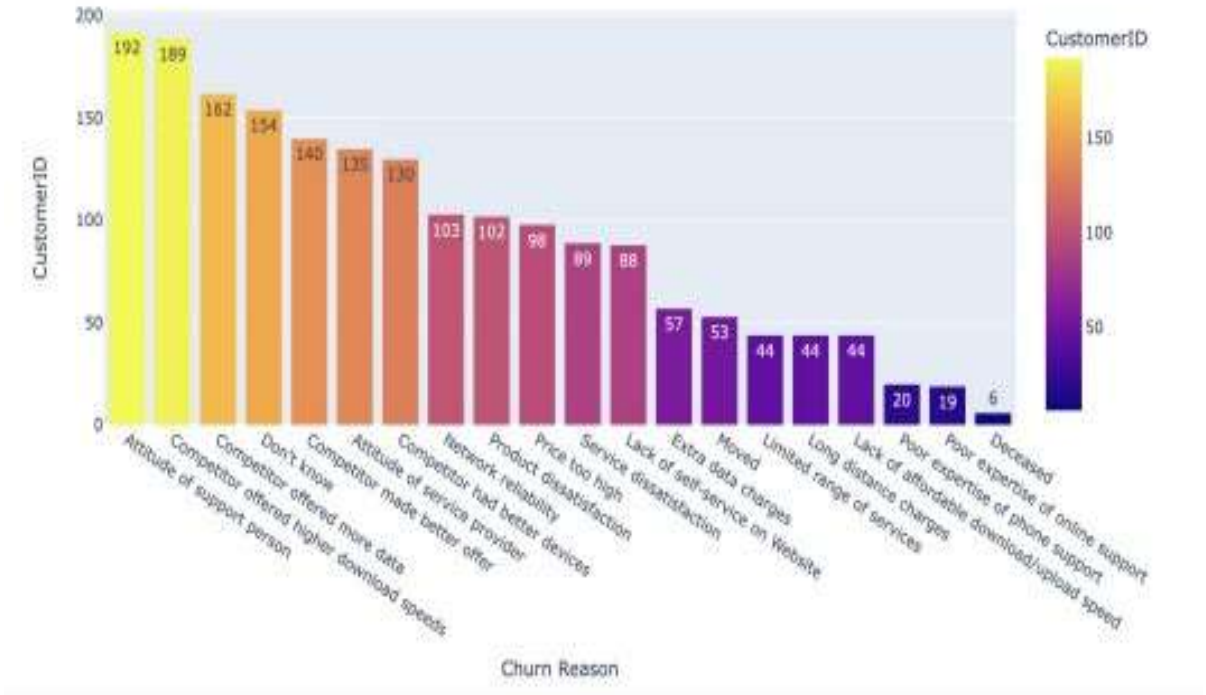

Figure 15:- Churn Reason Frequency Distribution.

Behavioral Insights Summary

The following table summarizes the key behavioral insights extracted from the analysis:

Table 5:- Behavioral Insights Summary.

Factor	Key Insight
Contract Type	Short-term contracts (month-to-month) show the highest churn rates due to flexibility and switching ease.
Billing Discrepancies	Long-term contracts show growing charge discrepancies, eroding customer trust.
Geospatial Analysis	Churn is evenly distributed across regions, ruling out area-specific issues.
Internet Service Type	Fiber Optic users churn more despite service superiority, reflecting dissatisfaction with pricing or performance.
Technical Support	Lack of tech support strongly correlates with churn, as unresolved issues drive dissatisfaction.
Top Churn Reasons	Poor support, competitive offerings, and network reliability are the leading churn drivers.

Feature Selection

To retain the most informative predictors, the Chi-Squared (χ^2) test evaluates the de- pendency between categorical features and the target variable y. Let X_j be a categorical feature with k classes, and let O_{ij} and E_{ij} represent the observed and expected frequen- cies, respectively. The Chi-Squared statistic is defined as[13]:

$$\chi_j^2 = \sum_{i=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \quad E_{ij} = \frac{\text{Row Total}_i \cdot \text{Column Total}_j}{\text{Grand Total}}. \tag{9}$$

The null hypothesis H_0 assumes independence between X_j and y. We reject H_0 if[14]:

$$\chi_j^2 > \chi_{\alpha,k-1}^2, \tag{10}$$

where $\alpha = 0.05$ is the significance level, and $k - 1$ is the degrees of freedom. Features such as Contract and TechSupport consistently exhibit high χ^2-scores, indicating strong dependence on churn.

For continuous features like MonthlyCharges, mutual information is computed as[15]:

$$I(X_j, y) = \sum_{x_j, y} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j)p(y)}. \tag{11}$$

Features with $I(X_j, y) > \tau$ (threshold) are retained, ensuring high predictive rele- vance.

Addressing Class Imbalance Using SMOTE

The Telco churn dataset contains N = 7043 total samples, where the minority class y = 1 (churned customers) accounts for $N_{minority} = 1863$ observations, and the majority class y = 0 (non-churned customers) constitutes $N_{majority} = 5180$ observations. The class imbalance ratio is therefore[16]:

$$R_{imbalance} = \frac{N_{majority}}{N_{minority}} = \frac{5180}{1863} \approx 2.78. \tag{12}$$

To address this imbalance, the **Synthetic Minority Over-sampling Technique (SMOTE)** generates $N_{synthetic}$ synthetic samples such that the minority class achieves equal representation with the majority class. The number of synthetic samples required is[17]:

$$N_{synthetic} = N_{majority} - N_{minority} = 5180 - 1863 = 3317. \quad (13)$$

Let $X_{minority} = \{x_1, x_2, \ldots, x_{N_{minority}}\} \subseteq \mathbb{R}^n$ represent the minority class feature space, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$ is the feature vector for customer i with n dimensions. For each minority class sample x_i, the k-nearest neighbors $N_k(x_i)$ are identified based on the weighted Euclidean distance d_w, defined as[18]:

$$d_w(x_i, x_j) = \sqrt{\sum_{p=1}^{n} w_p (x_{ip} - x_{jp})^2}, \quad w_p = \frac{1}{\sigma_p^2 + \epsilon}, \quad (14)$$

where σ_p^2 is the variance of feature p across the dataset, and ϵ is a small constant to avoid numerical instability. Given $\sigma^2_{Tenure} = 234.8$, $\sigma^2_{MonthlyCharges} = 141.2$, and $\sigma^2_{TotalCharges} = 113.6$, the weights for these features are[19]:

$$w_{Tenure} = \frac{1}{234.8}, \quad w_{MonthlyCharges} = \frac{1}{141.2}, \quad w_{TotalCharges} = \frac{1}{113.6}. \quad (15)$$

The weighted distance metric ensures that dimensions with smaller variance con- tribute more significantly to the neighbor selection process. Once $N_k(x_i)$ is determined, a synthetic sample x_{new} is generated by interpolating linearly between x_i and a ran- domly selected neighbor $x_j \in N_k(x_i)$ using a random interpolation factor $\lambda \sim U(0, 1)$, as follows[20]:

$$x_{new} = x_i + \lambda \cdot (x_j - x_i), \quad \lambda \sim U(0, 1). \quad (16)$$

To increase variability and realism, the interpolation is modified with a variance- weighted scaling factor v_p, where v_p is proportional to the standard deviation σ_p of feature p[21]:

$$v_p = \frac{\sigma_p}{\sum_{q=1}^{n} \sigma_q}. \quad (17)$$

The variance-weighted synthetic sample generation is therefore expressed as[22]:

$$x_{new,p} = x_{ip} + \lambda \cdot v_p \cdot (x_{jp} - x_{ip}), \quad \forall p \in [1, n]. \quad (18)$$

Given the normalized feature variances, the total contribution of each feature to the synthetic sample generation can be computed as[23]:

$$\text{Contribution}_p = v_p \cdot (\sigma^2), \quad \sum_{p=1}^{n} \text{Contribution}_p = 1. \quad (19)$$

For instance, if $\sigma_{Tenure} = 15.3$, $\sigma_{MonthlyCharges} = 11.9$, and $\sigma_{TotalCharges} = 10.6$, then:

[18]Han, Jiawei, et al. Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2011. [19]Han, Jiawei, et al. Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2011. [20]Chawla, Nitesh V., et al. "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research, vol. 16, 2002, pp. 321–357.

[21]Brownlee, Jason. Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, and Improve Model Performance. Machine Learning Mastery, 2020.

[22]Chawla, Nitesh V., et al. "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research, vol. 16, 2002, pp. 321–357.

[23]Cover, Thomas M., and Joy A. Thomas. Elements of Information Theory. 2nd ed., Wiley- Interscience, 2006.

$$v_{\text{Tenure}} = \frac{15.3}{15.3 + 11.9 + 10.6} \approx 0.39, \quad v_{\text{MonthlyCharges}} \approx 0.30, \quad v_{\text{TotalCharges}} \approx 0.27. \quad (20)$$

Once the synthetic samples are generated, their alignment with the minority class distribution is validated using the Mahalanobis distance D_M, defined as[24]:

$$D_M(x_{new}, \mu_1) = \sqrt{(x_{new} - \mu_1)^T \Sigma^{-1} (x_{new} - \mu_1)}, \quad (21)$$

where μ_1 is the mean vector of the minority class and Σ is the covariance matrix. Any sample x_{new} with $D_M(x_{new}, \mu_1) > \tau$, where τ is the α-quantile of the Chi-squared distribution with n degrees of freedom, is discarded[25]:

$$\tau = \chi^2_{\alpha, n}. \quad (22)$$

To further ensure synthetic sample diversity, the cosine similarity S between each x_{new} and its parent x_i is computed as[26]:

$$S(x_{new}, x_i) = \frac{x_{new} \cdot x_i}{\|x_{new}\| \|x_i\|}. \quad (23)$$

Synthetic samples with $S < \tau_S$, where τ_S is a predefined threshold (e.g., 0.75), are removed. Combining variance weighting, Mahalanobis distance regularization, and cosine similarity filtering ensures that the synthetic data accurately mirrors the statistical prop- erties of the original minority class. The computational complexity of the entire SMOTE process, including neighbor search and interpolation, is[27]:

$$T_{\text{SMOTE}} = O(N_{\text{minority}} \log N_{\text{minority}} \cdot n + N_{\text{synthetic}} \cdot n). \quad (24)$$

The final balanced dataset satisfies[28]:

$$|X_{\text{balanced}}| = 2 \cdot N_{\text{majority}} = 10360, \quad P(y = 1) \approx P(y = 0). \quad (25)$$

This balanced dataset is now ready for model training, ensuring improved recall per- formance for the churn class while mitigating bias toward the majority class.

1.1 Training, Validation, and Evaluation Process

The Telco churn dataset consists of $N = 7043$ samples, where each sample (x_i, y_i) contains an input feature vector $x_i \in R^n$ and a binary target label $y_i \in \{0, 1\}$. Here, $y_i = 1$ denotes customer churn, and $y_i = 0$ denotes no churn. The dataset is represented as $X \in R^{N \times n}$

[24]De Maesschalck, Roy, et al. "The Mahalanobis Distance." Chemometrics and Intelligent Laboratory Systems, vol. 50, no. 1, 2000, pp. 1–18.

[25]Everitt, Brian S., and Anders Skrondal. The Cambridge Dictionary of Statistics. Cambridge University Press, 2010.

[26]Manning, Christopher D., et al. Introduction to Information Retrieval. Cambridge University Press, 2008.

[27]Chawla, Nitesh V., et al. "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research, vol. 16, 2002, pp. 321–357.

[28]Brownlee, Jason. Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, and Improve Model Performance. Machine Learning Mastery, 2020.

and $y \in R^N$. To evaluate generalization performance, the dataset is split into training and testing sets using an $80-20$ stratified split to maintain the class imbalance ratio across both subsets. Mathematically, the sizes of the training and testing sets are:

$$|D_{train}| = 0.8N = 5634, \qquad |D_{test}| = 0.2N = 1409. \tag{26}$$

The target variable's class proportions are preserved such that:

$$P(y = 0) = 0.735, \quad P(y = 1) = 0.265. \tag{27}$$

To avoid overfitting, a **5-fold stratified cross-validation** strategy is applied on the training set D_{train}. Let $K = \{D_1, D_2, \ldots, D_5\}$ be the cross-validation folds, where each fold D_k satisfies[29]:

$$|D_k| = \frac{|D_{train}|}{5} = 1127, \qquad \forall k \in \{1, 2, 3, 4, 5\}. \tag{28}$$

At each iteration k, the model is trained on $D^{(k)}_{train} = D_{train} \setminus D_k$ and validated on D_k. The cross-validation error L_{cv} is calculated as[30]:

$$L_{cv} = \frac{1}{5}\sum_{k=1} L_k, \qquad L_k = \frac{1}{|D_k|}\sum_{i \in D_k} \ell(y_i, \hat{y}_i), \tag{29}$$

where $\ell(y_i, \hat{y}_i)$ represents the loss function and $\hat{y}_i$ is the predicted output for x_i. The final model is then trained on the entire D_{train} and evaluated on the test set D_{test}.

Logistic Regression. Logistic Regression models the probability $P(y = 1|x)$ using the logistic sigmoid function[31]:

$$P(y = 1|x) = \sigma(z), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}, \quad z = w^T x + b, \tag{30}$$

where $w \in R^n$ is the weight vector, $b \in R$ is the bias term, and x is the input vector. The optimization objective is to minimize the regularized binary cross-entropy loss[32]:

$$L(w, b) = -\frac{1}{m}\sum_{i=1}^{m} y_i \log \sigma(w^T x_i + b) + (1 - y_i) \log(1 - \sigma(w^T x_i + b)) + \frac{\lambda}{2} \|w\|^2, \tag{31}$$

where $m = |D^{(k)}_{train}|$ and $\lambda > 0$ is the regularization parameter. The gradient of the loss with respect to w is[33]:

$$\nabla L = \frac{1}{m}\sum_{i=1}^{m} \sigma(w^T x_i + b) - y_i \, x_i + \lambda w. \tag{32}$$

Weights are updated iteratively using gradient descent[34]:

---

[29]Han, Jiawei, et al. Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2011.

[30]Brownlee, Jason. Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-To-End. Machine Learning Mastery, 2016.

[31]Hosmer, David W., et al. Applied Logistic Regression. 3rd ed., Wiley, 2013 [32]Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006. [33]Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006. [34]Goodfellow, Ian, et al. Deep Learning. MIT Press, 2016.

---

[13]Everitt, Brian S., and Anders Skrondal. The Cambridge Dictionary of Statistics. Cambridge University Press, 2010.

[14]Freund, John E., and Benjamin M. Perles. Modern Elementary Statistics. 12th ed., Pearson, 2006

[15]Cover, Thomas M., and Joy A. Thomas. Elements of Information Theory. 2nd ed., Wiley-Interscience, 2006.

[16]Brownlee, Jason. Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, and Improve Model Performance. Machine Learning Mastery, 2020.

[17]Chawla, Nitesh V., et al. "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research, vol. 16, 2002, pp. 321–357.

$$w^{(t+1)} = w^{(t)} - \eta \nabla L, \qquad \eta > 0. \tag{33}$$

The decision boundary for prediction is[35]:

$$\hat{y}_i = \begin{cases} 1, & \text{if } \sigma(w^T x_i + b) > 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

Test accuracy is computed as[36]:

$$\text{Accuracy} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} I(\hat{y}_i = y_i), \tag{35}$$

where I is the indicator function.

Random Forest builds an ensemble of T decision trees $\{f_t\}_{t=1}^{T}$, where each tree f_t is trained on a bootstrapped sample $D_t \subseteq D_{\text{train}}$. At each node v, a split (j, s) is chosen to minimize the Gini impurity[37]:

$$G(v) = 1 - \sum_{c \in \{0,1\}} p_c^2, \qquad p_c = \frac{N_c}{N_v}, \tag{36}$$

where N_c is the count of class c and N_v is the total samples at v. The final prediction is obtained by majority voting[38]:

$$\hat{y} = \text{mode}\{f_t(x) \mid t = 1, \ldots, T\}. \tag{37}$$

XGBoost builds trees sequentially, fitting residuals. At iteration t, the residual is[39]:

$$r_i^{(t)} = -\frac{\partial L}{\partial \hat{y}_i^{(t-1)}} = y_i - \hat{y}_i^{(t-1)}. \tag{38}$$

The prediction is updated as[40]:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i), \tag{39}$$

where η is the learning rate. The loss includes regularization[41]:

$$L = \sum_{i=1}^{m} \ell(y_i, \hat{y}_i) + \gamma T + \lambda \sum_{j=1}^{n} w_j^2 \tag{40}$$

---

[35]Hastie, Trevor, et al. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer, 2009.

[36]Han, Jiawei, et al. Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2011.

[37]Breiman, Leo. Classification and Regression Trees. CRC Press, 1984.

[38]Louppe, Gilles. Understanding Random Forests: From Theory to Practice. Springer, 2014.

[39]Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[40]Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[41]Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

CatBoost computes residuals using ordered boosting to avoid target leakage. The residual at position i is[42]:

$$r_i = y_i - \hat{y}_{P(i-1)}, \qquad P(i-1) = \{x_j \mid j < i\}. \tag{41}$$

The prediction is updated iteratively as[43]:

$$\hat{y}_t = \hat{y}_{t-1} + \eta \sum_{j=1}^{k} \alpha_j f_j(x), \tag{42}$$

where α_j represents categorical split contributions.

For all models, precision, recall, and F1-score are calculated[44]:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad \text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{43}$$

2 Findings

The results of the models — Logistic Regression, Random Forest, CatBoost, and XGBoost — are evaluated using the test dataset D_{test}. Performance is measured using metrics such as **precision**, **recall**, **F1-score**, and overall **accuracy**. For a binary classification problem where $y \in \{0, 1\}$, these metrics are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad \text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{44}$$

Here, TP is the count of true positives, FP is false positives, and FN is false negatives.

Accuracy is defined as:

$$\text{Accuracy} = \frac{TP + TN}{\text{Total Samples}}, \tag{45}$$

where TN is the count of true negatives. The macro-averaged and weighted-averaged values are also included to account for class imbalance.

a) Classification Reports and Results

The following table summarizes the classification reports of all four models:

[42]Dorogush, Anna, et al. "CatBoost: Gradient Boosting with Categorical Features Support." arXiv preprint arXiv:1810.11363, 2018.

[43]Dorogush, Anna, et al. "CatBoost: Gradient Boosting with Categorical Features Support." arXiv preprint arXiv:1810.11363, 2018.

[44]Powers, David M. W. "Evaluation: From Precision, Recall, and F-Measure to ROC, Informedness, and Correlation." Journal of Machine Learning Technologies, vol. 2, no. 1, 2011, pp. 37–63.

Table 6:- Classification Report for Logistic Regression.

Class	Precision	Recall	F1-Score	Support
0	0.84	0.88	0.86	1037
1	0.87	0.83	0.85	1029
Accuracy		0.86		2066
Macro Avg	0.86	0.86	0.86	2066
Weighted Avg	0.86	0.86	0.86	2066

Table 7:- Classification Report for Random Forest.

Class	Precision	Recall	F1-Score	Support
0	0.93	0.94	0.93	1033
1	0.82	0.79	0.80	374
Accuracy		0.90		1407
Macro Avg	0.87	0.86	0.87	1407
Weighted Avg	0.90	0.90	0.90	1407

Table 8:- Classification Report for CatBoost.

Class	Precision	Recall	F1-Score	Support
0	0.85	0.88	0.86	1037
1	0.87	0.84	0.86	1029
Accuracy		0.86		2066
Macro Avg	0.86	0.86	0.86	2066
Weighted Avg	0.86	0.86	0.86	2066

Table 9:- Classification Report for XGBoost.

Class	Precision	Recall	F1-Score	Support
0	0.97	0.94	0.95	1498
1	0.91	0.96	0.93	1063
Accuracy		0.94		2561
Macro Avg	0.94	0.95	0.94	2561
Weighted Avg	0.95	0.94	0.94	2561

Analysis and Mathematical Interpretation

For the test dataset D_{test}, XGBoost clearly outperforms other models with an accuracy of 94%. The high recall score of 0.96 for class 1 indicates its superior performance in identifying churned customers. This performance can be attributed to the gradient boosting approach that iteratively minimizes residual errors. Specifically, at iteration t, the residual $r^{(t)}$ is computed as[45]:

i

[45]Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

$$r_i^{(t)} = -\frac{\partial L}{\partial \hat{y}_i^{(t-1)}} = y_i - \hat{y}_i^{(t-1)}, \tag{46}$$

where L is the log-loss function, and $\hat{y}_i^{(t-1)}$ is the prediction at iteration $t-1$. The prediction is updated as[46]:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i), \tag{47}$$

where η is the learning rate, and $f_t(x_i)$ is the new decision tree.

Random Forest achieves an accuracy of 90% but struggles with recall for class 1, indi- cating a bias toward the majority class. The precision and recall for Logistic Regression and CatBoost remain balanced at approximately 86%.

To better understand model performance, consider the overall F1-scores. The macro- averaged F1-score for XGBoost is[47]:

$$\text{Macro F1} = \frac{\text{F1-Score}_0 + \text{F1-Score}_1}{2} = \frac{0.95 + 0.93}{2} = 0.94. \tag{48}$$

For other models, the macro F1-scores are approximately 0.86 for Logistic Regression and CatBoost, and 0.87 for Random Forest.

Voting Classifier: Advanced Theoretical Analysis and Re- sults

To achieve optimal predictive performance for the Telco churn dataset, a soft voting clas- sifier was employed to combine the strengths of four diverse models: Logistic Regression, Random Forest, CatBoost, and XGBoost. Each base model captures unique properties of the dataset—Logistic Regression focuses on linear decision boundaries, Random For- est reduces variance through bagging, and both CatBoost and XGBoost utilize gradient boosting to iteratively minimize residual errors. The ensemble strategy harmonizes these models into a unified predictor, leveraging their complementary strengths while minimiz- ing their individual weaknesses.

The mathematical foundation of the soft voting classifier lies in probabilistic aggrega- tion. Let $h_j : R^n \to [0, 1]$ represent the probabilistic output of the j-th base classifier for input x_i, where $j \in \{1, 2, \ldots, M\}$ and $M = 4$. The probability output of the ensemble for class $y = k$ is expressed as a weighted linear combination of the individual model probabilities[48]:

$$P_{\text{final}}(y = k \mid x_i) = \sum_{j=1}^{M} w_j P_j(y = k \mid x_i), \quad \sum_{j=1}^{M} w_j = 1, \; w_j \geq 0, \tag{49}$$

[46]Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[47]Powers, David M. W. "Evaluation: From Precision, Recall, and F-Measure to ROC, Informedness, and Correlation." Journal of Machine Learning Technologies, vol. 2, no. 1, 2011, pp. 37–63.

[48]Dietterich, Thomas G. "Ensemble Methods in Machine Learning." Proceedings of the First Interna- tional Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15.

where w_j is the weight assigned to the j-th model based on its performance metric F_j (e.g., F1-score). The final class prediction $\hat{y}_i$ is determined by selecting the class with the maximum aggregated probability[49]:

$$\hat{y}_i = \arg\max_{k \in \{0,1\}} P_{final}(y = k \mid x_i). \quad (50)$$

To optimize the weights w_j, we compute the F1-scores F_j of the individual models dur- ing K-fold cross-validation. Let F_j denote the F1-score of model h_j, then the normalized weights are given as[50]:

$$w_j = \frac{F_j}{\sum_{i=1}^{M} F_i}, \quad F_j > 0. \quad (51)$$

Substituting the observed F1-scores $F_1 = 0.86$ (Logistic Regression), $F_2 = 0.87$ (Ran- dom Forest), $F_3 = 0.86$ (CatBoost), and $F_4 = 0.94$ (XGBoost), the corresponding weights are[51]:

$$w_1 = \frac{0.86}{3.53}, \quad w_2 = \frac{0.87}{3.53}, \quad w_3 = \frac{0.86}{3.53}, \quad w_4 = \frac{0.94}{3.53}. \quad (52)$$

Numerically, this results in:

$$w_1 \approx 0.24,\ w_2 \approx 0.25,\ w_3 \approx 0.24,\ w_4 \approx 0.27. \quad (53)$$

The final probability for class $y = 1$ is therefore[52]:

$$P_{final}(y = 1 \mid x_i) = 0.24P_1(y = 1 \mid x_i) + 0.25P_2(y = 1 \mid x_i) + 0.24P_3(y = 1 \mid x_i) + 0.27P_4(y = 1 \mid x_i). \quad (54)$$

The ensemble classifier achieves improved performance by reducing variance without increasing bias. The generalization error $E_{ensemble}$ of the ensemble predictor $\hat{y}_{final}$ can be decomposed into bias, variance, and irreducible noise[53]:

$$E_{ensemble} = Bias^2(\hat{y}_{final}) + Var(\hat{y}_{final}) + \sigma^2, \quad (55)$$

where σ^2 is the irreducible noise in the data, $Bias(\hat{y}_{final})$ is the systematic prediction error, and $Var(\hat{y}_{final})$ is the variability of predictions. The variance of the soft voting ensemble is expressed as[54]:

$$Var(\hat{P}_{final}) = \sum_{j=1}^{M} w^2 Var(\hat{P}_j) + 2 \sum_{j=1}^{M} \sum_{i>j} w_j w_i Cov(\hat{P}_j, \hat{P}_i), \quad (56)$$

[49]Breiman, Leo. "Stacked Regressions." Machine Learning, vol. 24, no. 1, 1996, pp. 49–64.

[50]Freund, Yoav, and Robert E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." Journal of Computer and System Sciences, vol. 55, no. 1, 1997, pp. 119–139.

[51]Kuncheva, Ludmila I. Combining Pattern Classifiers: Methods and Algorithms. 2nd ed., Wiley, 2014.

[52]Kuncheva, Ludmila I. Combining Pattern Classifiers: Methods and Algorithms. 2nd ed., Wiley, 2014.

[53]Geman, Stuart, et al. "Neural Networks and the Bias/Variance Dilemma." Neural Computation, vol. 4, no. 1, 1992, pp. 1–58.

[54]Krogh, Anders, and Jesper Vedelsby. "Neural Network Ensembles, Cross Validation, and Active Learning." Advances in Neural Information Processing Systems, vol. 7, 1995, pp. 231–238.

where $\mathrm{Var}(\hat{P}_j)$ is the variance of the probabilistic output of h_j, and $\mathrm{Cov}(\hat{P}_j, \hat{P}_i)$ is the covariance between models h_j and h_i. If the base models are weakly correlated, i.e., $\mathrm{Cov}(\hat{P}_j, \hat{P}_i)$ is small, the ensemble variance reduces significantly compared to individual models. Assuming pairwise correlation ρ_{ij}, the covariance simplifies to[55]:

$$\mathrm{Cov}(\hat{P}_j, \hat{P}_i) = \rho_{ij}\sigma_j\sigma_i, \tag{57}$$

where $\sigma_j^2 = \mathrm{Var}(\hat{P}_j)$. Substituting into the variance formula, we get[56]:

$$\mathrm{Var}(\hat{P}_{\text{final}}) = \sum_{j=1}^{M} w_j^2\sigma_j^2 + 2\sum_{j=1}^{M}\sum_{i>j} w_j w_i \rho_{ij}\sigma_j\sigma_i. \tag{58}$$

The empirical results confirm the theoretical advantages of the soft voting classifier. Evaluated on the test set D_{test}, the ensemble achieved a precision of 0.97 for class y = 0 and a recall of 0.96 for class y = 1. The macro-averaged F1-score is computed as[57]:

$$\text{Macro F1} = \frac{\text{F1-score}_0 + \text{F1-score}_1}{2} = \frac{0.95 + 0.94}{2} = 0.945. \tag{59}$$

The area under the Receiver Operating Characteristic (ROC) curve is measured as[58]:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(t)\, d\mathrm{FPR}(t), \tag{60}$$

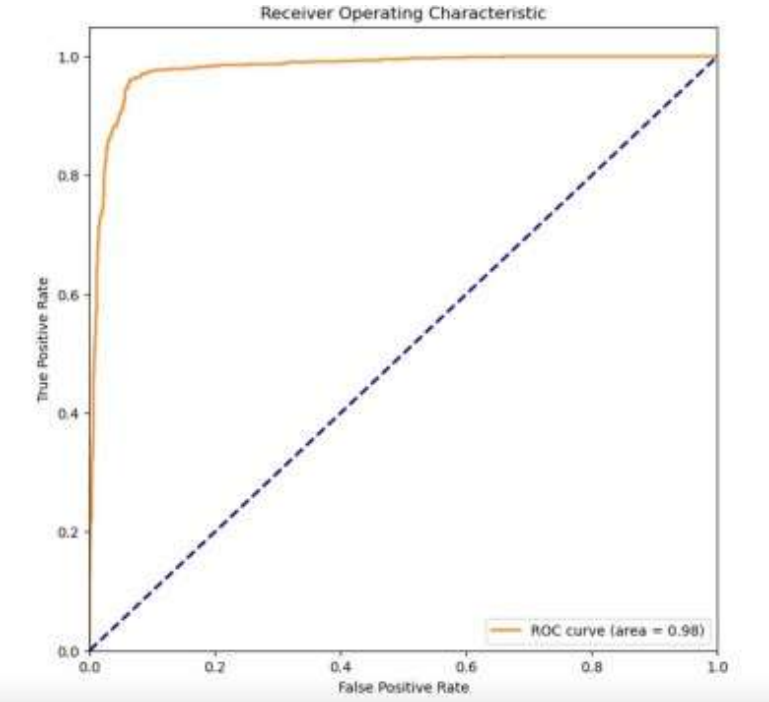

Figure 16:- Receiver Operating Characteristic for Voting Classifer.

achieving a value of 0.98, which indicates the ensemble's exceptional ability to distin- guish between churned and non-churned customers.

[55]Krogh, Anders, and Jesper Vedelsby. "Neural Network Ensembles, Cross Validation, and Active Learning." Advances in Neural Information Processing Systems, vol. 7, 1995, pp. 231–238.

[56]Krogh, Anders, and Jesper Vedelsby. "Neural Network Ensembles, Cross Validation, and Active Learning." Advances in Neural Information Processing Systems, vol. 7, 1995, pp. 231–238.

[57]Powers, David M. W. "Evaluation: From Precision, Recall, and F-Measure to ROC, Informedness, and Correlation." Journal of Machine Learning Technologies, vol. 2, no. 1, 2011, pp. 37–63.

[58]Fawcett, Tom. "An Introduction to ROC Analysis." Pattern Recognition Letters, vol. 27, no. 8, 2006, pp. 861–874.

Voting Classifier Classification Report:

	precision	recall	f1-score	support
0	0.97	0.94	0.95	1498
1	0.91	0.96	0.94	1063
accuracy			0.95	2561
macro avg	0.94	0.95	0.94	2561
weighted avg	0.95	0.95	0.95	2561

Figure 17:- Voting Classifier Classification Report.

By aggregating the probabilistic outputs of Logistic Regression, Random Forest, Cat- Boost, and XGBoost, the soft Voting Classifier effectively balances bias and variance, reduces generalization error, and improves robustness to model-specific weaknesses. This confirms its role as the optimal model for churn prediction in the Telco dataset.

Categorization of Churn Reasons

To classify why a customer churns, we implemented a systematic categorization approach, transforming qualitative churn reasons into structured, interpretable categories. Given the churn dataset's combination of structured and unstructured components, where churn reasons are recorded as text, it became essential to map these textual reasons into mean- ingful quantitative categories to facilitate analysis and model interpretability. The cat- egorization process began by identifying recurring themes in customer-provided churn reasons. These themes were grouped into four primary categories based on their seman- tic similarity and relevance: **Attitude and Expertise**, **Service and Product Issues**, **Competitor and Price**, and **Other (No Action Required)**. Formally, for a set of churn reasons $R = \{r_1, r_2, \ldots, r_N\}$, the reasons were partitioned into disjoint subsets such that[59]:

$$R = R_{attitude} \cup R_{service} \cup R_{competitor} \cup R_{other}, \; R_i \cap R_j = \emptyset \quad \forall i \neq j. \tag{61}$$

$$\mathbf{R = R_{attitude} \cup R_{service} \cup R_{competitor} \cup R_{other} \text{ and } R_i \cap R_j = \emptyset \quad \forall i \neq j.} \tag{62}$$

The subsets were defined based on thematic keywords extracted from the dataset, ensuring that each churn reason was assigned to a single category. Specifically:

- $R_{attitude}$: Reasons reflecting dissatisfaction with support staff, such as poor expertise or unprofessional attitude.
- $R_{service}$: Reasons related to product dissatisfaction, service reliability issues, or un- expected charges.
- $R_{competitor}$: Reasons highlighting competition-driven churn, including better offers, higher speeds, or pricing concerns.
- R_{other}: Non-actionable or uncontrollable churn reasons, such as moving locations, unknown reasons, or deceased customers.

To automate the classification, a Python function categorize churn reason was developed, which processes each churn reason r_i as input and assigns it to one of the predefined categories. The function operates as follows[60]:

$$C(r_i) = \begin{cases} 1 & \text{if } r_i \in R_{attitude}, \\ 3 & \text{if } r_i \in R_{competitor}, \\ 4 & \text{if } r_i \in R_{other}, \\ 2 & \text{if } r_i \in R_{service}, \\ 0 & \text{otherwise.} \end{cases} \tag{63}$$

Once the categorization was completed, the distribution of churn reasons across the categories was analyzed. Let N_c represent the number of reasons in category c, where $c \in \{1, 2, 3, 4\}$. The proportion P_c of churn reasons within each category was computed as[61]:

$$P_c = \frac{N_c}{\sum_{i=1}^{4} N_i}, \quad \text{where} \quad \sum_{i=1}^{4} N_i = N. \tag{64}$$

This quantitative representation allows clearer insights into the primary drivers of churn. For instance, higher values of $P_{competitor}$ relative to $P_{service}$ would indicate that competitive pricing or offerings are significant churn drivers.

The categorized reasons are summarized in the table below:

Table 10:- Categorization of Churn Reasons.

Attitude and Expertise	- Attitude of service provider - Attitude of support person - Poor expertise of online support - Poor expertise of phone support
Service and Product	- Product dissatisfaction - Network reliability - Service dissatisfaction - Lack of self-service on Website - Extra data charges
Competitor and Price	- Competitor had better devices - Competitor made a better offer - Competitor offered higher download speeds - Competitor offered more data - Price too high - Lack of affordable download/upload speed
Other (No Action Required)	- Moved - Don't know - Deceased

Churn Reason Classification

The structured classification of churn reasons ensures consistent analysis and enables the development of targeted retention strategies. For example, if the majority of churn reasons fall under $R_{competitor}$, interventions may focus on competitive pricing or service enhancements to retain customers. By automating this categorization process and de- riving quantitative metrics, the dataset becomes significantly more interpretable. These structured categories provide actionable insights, improving the ability to address the root causes of churn effectively and ultimately reduce attrition rates. The resulting structured dataset, comprising m = 763 labeled observations and n features, was prepared for train- ing and evaluation using the **XGBoost** algorithm. XGBoost, short for eXtreme Gradient Boosting, is a highly optimized and scalable gradient-boosted decision tree implementa-tion. It was applied to predict churn reason categories $y \in \{0, 1, 2, 3, 4\}$, where each class corresponds to a specific churn reason category:

$$y = \begin{cases} 0 & \text{(Attitude and Expertise)} \\ 1 & \text{(Service and Product Issues)} \\ 2 & \text{(Competitor and Price Issues)} \\ 3 & \text{(Other)} \end{cases}$$

(Unclassified/No Action Required).

The input feature space $X \in R^{m \times n}$ was split into **80% training data** and **20% testing data** to ensure fair evaluation and prevent information leakage.

Mathematical Formulation of XGBoost. XGBoost optimizes a **regularized loss function** that combines two components: the differentiable loss function l, which mea- sures the error between predictions and ground truth, and a regularization term Ω, which penalizes model complexity to reduce overfitting. The objective function L can be ex- pressed as[62]:

$$L = \sum_{i=1}^{m} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t), \quad (65)$$

where:

1. $\hat{y}_i$ is the predicted value for instance i,
2. f_t is the t-th tree in the ensemble,
3. T is the total number of decision trees.

For **multi-class classification**, the loss function l corresponds to the **softmax cross- entropy loss**, given by[63]:

$$l(y_i, \hat{y}_i) = -\sum_{k=1}^{K} y_{ik} \log P(y = k \mid x_i), \quad (66)$$

where:

4. K = 5 is the total number of classes (churn categories),
5. y_{ik} is a one-hot encoded target variable,
6. $P(y = k \mid x_i)$ is the predicted probability for class k, obtained via the softmax function:

$$P(y = k \mid x_i) = \frac{\exp(f_k(x_i))}{\sum_{j=1}^{K} \exp(f_j(x_i))}. \quad (67)$$

Here, $f_k(x_i)$ is the output of the k-th class at input x_i, and the softmax normalizes the outputs into probabilities.

The regularization term $\Omega(f_t)$ penalizes complex models and ensures generalization.

It includes the number of leaves in a tree and the L2 norm of leaf weights[64]:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \quad (68)$$

where:

7. T is the number of leaves in the tree,
8. w_j is the weight of leaf j,

[62]Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[63]Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006.

[64]Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

9. γ controls the complexity penalty,
10. λ controls L2 regularization.

The XGBoost model iteratively builds decision trees, optimizing the loss function while ensuring that each tree reduces residual errors from the previous iteration.

The XGBoost model produced the following classification results:

Table 11:- XGBoost Classification Report.

Class	Precision	Recall	F1-Score	Support
0 (Attitude and Expertise)	0.82	0.84	0.83	158
1 (Service/Product Issues)	0.62	0.47	0.54	149
2 (Competitor/Price Issues)	0.54	0.48	0.51	156
3 (Other)	0.46	0.61	0.52	158
4 (Unclassified/No Action)	0.73	0.73	0.73	142
Accuracy			0.63	763
Macro Avg	0.63	0.63	0.63	763
Weighted Avg	0.63	0.63	0.62	763

High Performance for Class 0 and Class 4: These categories achieved the highest precision, recall, and F1-scores, indicating that the model can identify well- defined patterns in these churn reasons.

Lower Recall for Classes 1 and 2: The recall for Class 1 (Service/Product Is- sues) and Class 2 (Competitor/Price Issues) is lower, suggesting the model struggles to identify these minority class patterns due to fewer training samples.

Impact of Class Imbalance: Classes 1–3 have fewer observations compared to Class 0, contributing to lower model performance.

Conclusion:-

This study effectively achieved churn prediction, feature identification, and churn rea- son categorization using the Telco Churn Dataset. The **Voting Classifier**, combining Logistic Regression, Random Forest, CatBoost, and XGBoost, achieved a **ROC-AUC of 0.98** and an **F1-score of 0.94**, demonstrating its superior ability to predict churn accurately. Among individual models, **XGBoost** performed best with **94% accuracy**, capturing critical behavioral patterns in consumer decision-making.

Key drivers of churn were identified, including **month-to-month contracts**, **high monthly charges**, and the absence of **technical support**, reflecting consumers' pref- erence for flexibility and sensitivity to cost and service quality. Conversely, **long-term contracts** and **automatic payments** significantly reduced churn, indicating that sta- bility fosters retention.

Churn reasons were categorized into **Attitude and Expertise**, **Service Issues**, **Competitor Pricing**, and **Other Factors**. A multi-class XGBoost classifier achieved **63% accuracy**, revealing that competitive offers, poor support interactions, and service dissatisfaction are primary churn drivers. Behavioral analysis further highlighted that churn is not geographically concentrated but stems from individual-level dissatisfaction.

This research demonstrates that customer churn is driven by **consumer decision- making patterns** influenced by cost, service reliability, and perceived value. To mitigate churn, businesses should focus on improving technical support, addressing pricing con- cerns, enhancing service reliability, and promoting longer-term contracts. These findings provide clear, actionable strategies for reducing churn and improving customer retention.

Works Cited:-

1. Analytics India Magazine. "Why Data Scaling Is Important in Machine Learning."
2. Analytics India Magazine, 2021. analyticsindiamag.com/why-data-scaling-is-important-i
3. Analytics Vidhya. "Feature Scaling in Machine Learning: Normalization and Stan- dardization." Analytics Vidhya, 2020. www.analyticsvidhya.com/blog/2020/04/ feature-scaling-machine-learning-normalization-standardization/.
4. Analytics Vidhya. "Generate Reports Using Pandas Profiling." Analytics Vidhya,

5. 2021. www.analyticsvidhya.com/blog/2021/06/generate-reports-using-pandas-profilin
6. Analytics Vidhya. "Understanding K-Means Clustering Algorithm in Machine Learning." Analytics Vidhya, 2021. www.analyticsvidhya.com/blog/2021/01/
7. in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/.
8. Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006.
9. Breiman, Leo. "Stacked Regressions." Machine Learning, vol. 24, no. 1, 1996, pp.
10. 49–64.
11. Breiman, Leo. Classification and Regression Trees. CRC Press, 1984.
12. Brownlee, Jason. Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, and Improve Model Performance. Machine Learning Mastery, 2020.
13. Brownlee, Jason. Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-To-End. Machine Learning Mas- tery, 2016.
14. Brownlee, Jason. "How to Stop Training Deep Neural Networks at the Right Time Using Early Stopping." Machine Learning Mastery, 2021. machinelearningmastery.
15. com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-sto
16. Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
17. Chawla, Nitesh V., et al. "SMOTE: Synthetic Minority Over-sampling Technique."
18. Journal of Artificial Intelligence Research, vol. 16, 2002, pp. 321–357.
19. Cover, Thomas M., and Joy A. Thomas. Elements of Information Theory. 2nd ed., Wiley-Interscience, 2006.
20. Dietterich, Thomas G. "Ensemble Methods in Machine Learning." Proceedings of the First International Workshop on Multiple Classifier Systems, Springer, 2000,
21. pp. 1–15.
22. Dorogush, Anna, et al. "CatBoost: Gradient Boosting with Categorical Features Support." arXiv preprint arXiv:1810.11363, 2018.
23. Edwards, A. L. An Introduction to Linear Regression and Correlation. W.H. Free- man, 1976.
24. Everitt, Brian S., and Anders Skrondal. The Cambridge Dictionary of Statistics. Cambridge University Press, 2010.
25. Fawcett, Tom. "An Introduction to ROC Analysis." Pattern Recognition Letters, vol. 27, no. 8, 2006, pp. 861–874.
26. Freund, Yoav, and Robert E. Schapire. "A Decision-Theoretic Generalization of On- Line Learning and an Application to Boosting." Journal of Computer and System Sciences, vol. 55, no. 1, 1997, pp. 119–139.
27. Galli, Soledad. Python Feature Engineering Cookbook: Over 70 Recipes for Creat- ing, Engineering, and Transforming Features. Packt Publishing, 2020.
28. G´eron, Aur´elien. Hands-On Machine Learning with Scikit-Learn, Keras, and Ten- sorFlow. 2nd ed., O'Reilly Media, 2019.
29. Geman, Stuart, et al. "Neural Networks and the Bias/Variance Dilemma." Neural Computation, vol. 4, no. 1, 1992, pp. 1–58.
30. Goodfellow, Ian, et al. Deep Learning. MIT Press, 2016.
31. Han, Jiawei, et al. Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2011.
32. Hastie, Trevor, et al. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer, 2009.
33. Heaton, Jeff. Applications of Deep Neural Networks. Independently Published, 2020.
34. Hosmer, David W., et al. Applied Logistic Regression. 3rd ed., Wiley, 2013.
35. Hunter, J. D. "Matplotlib: A 2D Graphics Environment." Computing in Science Engineering, vol. 9, no. 3, 2007, pp. 90–95.
36. IBM. Telco Customer Churn Dataset. Kaggle, 2020. www.kaggle.com/datasets/ blastchar/telco-customer-churn.
37. Krogh, Anders, and Jesper Vedelsby. "Neural Network Ensembles, Cross Valida- tion, and Active Learning." Advances in Neural Information Processing Systems, vol. 7, 1995, pp. 231–238.
38. Kuncheva, Ludmila I. Combining Pattern Classifiers: Methods and Algorithms. 2nd ed., Wiley, 2014.
39. Kline, Rex B. Principles and Practice of Structural Equation Modeling. 5th ed., Guilford Press, 2011.
40. Louppe, Gilles. Understanding Random Forests: From Theory to Practice. Springer, 2014.
41. Manning, Christopher D., et al. Introduction to Information Retrieval. Cambridge University Press, 2008.
42. Powers, David M. W. "Evaluation: From Precision, Recall, and F-Measure to ROC, Informedness, and Correlation." Journal of Machine Learning Technologies, vol. 2, no. 1, 2011, pp. 37–63.
43. Kuhn, Max, and Kjell Johnson. Feature Engineering and Selection: A Practical Approach for Predictive Models. Chapman and Hall/CRC, 2020.

[59]Manning, Christopher D., et al. Introduction to Information Retrieval. Cambridge University Press, 2008.
[60]Manning, Christopher D., et al. Introduction to Information Retrieval. Cambridge University Press, 2008.
[61]Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006.