SEPTEMBER 2 :2024 CHENNAI

Customer Churn Prediction

Telecom customer Churn Prediction Using ANN and deploy it in Azure and display result using Tableau Dashboard.

By

Radhin Krishna R

22376003

BSC Data Science

V th Semester

CONTENTS

Introduction

This project aims to develop a robust churn prediction system capable of providing real-time insights into customer and employee attrition. By integrating various components, including data extraction, transformation, loading (ETL), KPI calculation, data cleaning and preprocessing, feature selection, machine learning model development, and visualization, this system will enable organizations to make informed decisions and implement proactive retention strategies. Our goal is not to become experts in a single tool or method, but to excel as solution experts. That's why we've integrated both Machine Learning (ML) and Artificial Neural Networks (ANN) into this project. Although we select the one with superior accuracy, time, and space complexity, we acknowledge the potential of employing multiple ML algorithms through an ensemble model. Presently, each model and dashboard is constructed using a static dataset.

Methodology:-
We aim to create a project that is reusable and flexible for all types of churn analysis, whether it's customer, segment, or employee churn. Each dataset varies by attributes and individual characteristics, so a one-size-fits-all architecture isn't feasible. However, we can develop a model that can analyze the data, retrain itself when accuracy dips below a certain threshold, and operate in a cloud environment for efficient resource management. This approach will also allow the use of real-time data, making it a versatile tool for various churn predictions within an organization.

Abstract

Customer Churn Prediction and Real-Time Analysis

This project aims to develop a comprehensive solution for predicting customer churn in the telecommunications industry and providing real-time insights for effective resource management and policy evaluation. By leveraging machine learning techniques, the project seeks to identify customers at risk of churning and offer actionable recommendations to mitigate losses.

Key components of the project include:

- Data Extraction and Preprocessing: Extraction of relevant data from various sources followed by cleaning, normalization, and feature engineering to ensure data quality and consistency.
- Machine Learning Model Development: Training and evaluation of machine learning models, such as ANNs, to accurately predict customer churn based on historical data. The models will be carefully selected and optimized to achieve the highest possible predictive performance.
- Real-Time Dashboard: Creation of an interactive dashboard to visualize churn metrics, predictions, and trends segmented by region, department, gender, and other relevant factors. The dashboard will provide a clear and intuitive overview of customer churn patterns and enable timely decision-making.
- Cloud-Based Architecture: Design and implementation of a scalable cloud architecture to enable real-time data ingestion, model deployment, and prediction generation. The architecture will be designed to handle large volumes of data and ensure high availability and performance.
- Continuous Learning: Incorporation of mechanisms for model retraining and updates based on new data to ensure ongoing accuracy and relevance. The model will be continuously evaluated and refined to adapt to changing customer behavior and market dynamics.

The project has made significant progress in the ETL process, KPI calculation, data cleaning, and feature selection. The next steps involve integrating the components into a cloud environment, selecting the optimal ML model, and developing a user-friendly web application for accessing the predictions.

By successfully implementing this project, telecommunications companies can gain valuable insights into customer behavior, optimize resource allocation, and improve customer retention

strategies. The project will provide a powerful tool for making data-driven decisions and enhancing overall business performance

Project Overview

MODULE 1 : ETL

1.1 Data extraction

The provided **Python code** effectively extracts data from a remote API and saves it in a structured CSV format, making it suitable for subsequent analysis and machine learning tasks. It starts by importing essential libraries: requests for handling HTTP requests, csv for working with CSV files, and json for parsing JSON data

The script then configures the API endpoint and API key, which are crucial for authentication and identification. It constructs a dictionary of request parameters, including the API key and any other required parameters. A GET request is sent to the API using the requests.get() function, and the response is evaluated. If the request is successful (status code 200), the JSON data is parsed using the response.json() method, converting it into a Python dictionary for easier manipulation.

The extracted data is then written to a CSV file using the csv module. A CSV writer object is created, and the header row containing column names is written to the file. The loop iterates through each item (data point) in the parsed JSON data, extracting the desired fields and writing them as a row to the CSV file. This ensures that the data is organized in a tabular format for efficient analysis.

To handle APIs that return paginated results, the code can be extended to include a loop that iterates through multiple pages until there are no more results. This is achieved by updating the page number in the request parameters and checking for a "next_page" indicator in the API response.

The code incorporates error handling to gracefully deal with unsuccessful requests. If the API request returns an error (non-200 status code), an informative error message is printed, and the program execution is terminated. This prevents further processing of potentially corrupted or incomplete data.

1.2 Data Transformation

The initial phase of the project involved transforming the extracted telecom churn data within MS SQL Server. Four tables were created to effectively organize and manage the data, ensuring efficient analysis. A comprehensive data transformation process was conducted to ensure data quality and consistency. This included cleaning the data by removing duplicates, handling null values, and correcting errors, as well as converting data types to match their intended use. Additionally, data standardization was implemented to ensure consistent formats and units across different columns and tables. To enhance the predictive power of the model, feature engineering techniques were applied, creating new features or transforming existing ones.

Once the data was transformed and cleaned, ten key performance indicators (KPIs) were defined to gain valuable insights into the telecom churn dataset. These KPIs were designed to measure various aspects of customer behaviour, service usage, and churn patterns. Examples include churn rate, average revenue per user (ARPU), customer lifetime value (CLTV), contract length, usage patterns, service bundle popularity, and customer satisfaction. SQL queries were leveraged to calculate these KPIs within MS SQL Server, providing a comprehensive understanding of customer churn trends and identifying areas for improvement.

This Data is saved for creating the first ANN model and analyse which all components are related and formalize the ANN architecture and to be used as an initial phase of visualization in tableau.

Module 2: Data Pre-processing- Analysis

2.1 Feature engineering

Data Cleaning

- Removal of Irrelevant Column: The "Customer ID" column is dropped as it's unlikely to have a direct impact on churn prediction.
- Handling Missing Values: Rows with missing values in the "Total Charges" column are identified and removed to ensure data consistency and avoid potential errors in subsequent analysis.

Data Type Conversion

- Conversion of "Total Charges": To ensure compatibility with numerical operations, the "Total Charges" column, initially a string, is converted to a numeric data type. This enables calculations and statistical analysis on this feature.

Data Exploration and Visualization

- Churn Distribution: Histograms are created to visualize the distribution of tenure and monthly charges for both churning and non-churning customers. This helps identify potential patterns or differences between these groups.

Feature Encoding

- Handling Categorical Features:
    - Replacement of Categorical Values: The values "No internet service" and "No phone service" are replaced with "No" for consistency and to simplify feature encoding.
    - Encoding of Binary Features: Columns representing binary features (e.g., "Married", "Phone Service") are converted to numeric values (1 for "Yes", 0 for "No"). This allows the model to directly process these features.
    - One-Hot Encoding: For columns with more than two categories (e.g., "Offer"), one-hot encoding is applied. This creates separate binary columns for each unique category, preventing the model from assuming an ordinal relationship between categories.

Data Normalization

- Scaling Numerical Features: Features with a wide range of values, such as "Tenure in Months", "Monthly Charge", and "Total Revenue", are scaled using MinMaxScaler. This ensures that features with larger scales don't dominate the model's learning process, leading to more balanced feature contributions.

Impact of Feature Engineering

- Improved Data Quality: Cleaning and handling missing values ensure that the model trains on reliable and consistent data, reducing the risk of errors and biases.
- Enhanced Model Performance: Feature encoding and normalization contribute to improved model performance by allowing the model to effectively process categorical data and avoid biases due to varying feature scales.
- Better Model Interpretability: One-hot encoding makes it easier to understand the influence of different categories on churn prediction, providing insights into the factors driving customer churn.

2.2 Data Modeling & Analysis

The provided code performs a series of data visualizations, analyses, and feature engineering steps to prepare a telecom customer churn dataset for machine learning model training. Here's a detailed breakdown of these processes and their impact on understanding and predicting customer churn

Tenure vs. Churn: This visualization employs histograms to compare the distribution of "Tenure in Months" for customers who churned (typically shown in red) and those who stayed (usually in green).

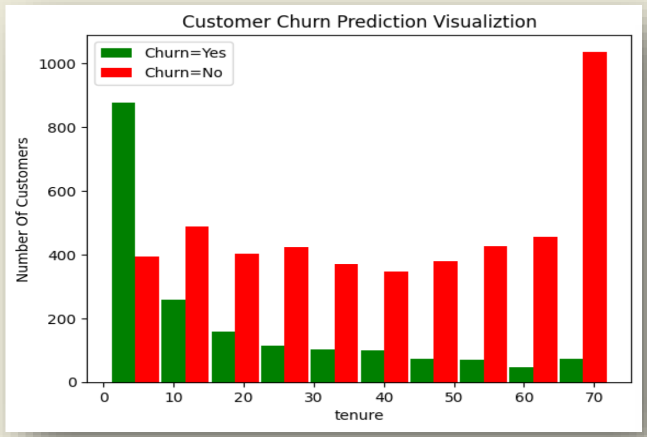
Analyzing this plot helps identify potential correlations between customer tenure and churn. For instance, a significantly higher churn rate at a specific tenure might indicate areas needing improvement in customer retention strategies. By visually comparing the distribution shapes, you can assess if churn is more likely for newer or more established customers.

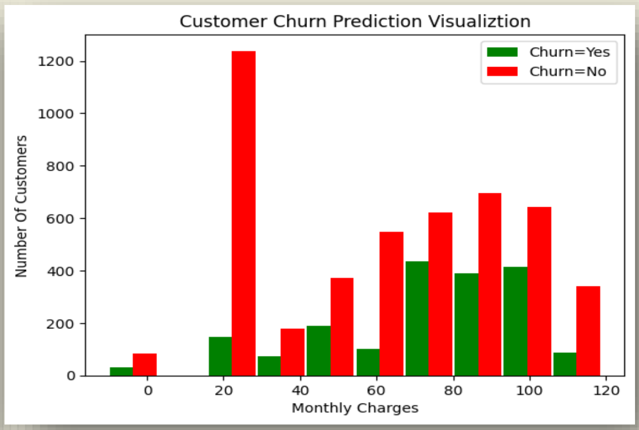
Impact: These visualizations provide an initial exploration of the data, revealing potential patterns related to churn. They are instrumental in understanding general trends but may not pinpoint specific factors influencing churn decisions. However, they can guide further analysis and feature selection for the machine learning model.

Feature Analysis:

Unique Values: The print_unique_col_values function identifies the unique values present in each categorical column. This helps understand the composition of the data and identify potential inconsistencies in how certain categories are represented (e.g., "No internet service" vs. a more standardized "No").

Impact: Feature analysis is crucial for determining how to handle categorical features during model training. By understanding the unique values within each category, you can decide on appropriate encoding techniques (e.g., one-hot encoding or label encoding) to transform these features into a format suitable for the machine learning model.





Unique Values: The print_unique_col_values function identifies the unique values present in each categorical column. This helps understand the composition of the data and identify potential inconsistencies in how certain categories are represented (e.g., "No internet service" vs. a more standardized "No").

Impact**:** Feature analysis is crucial for determining how to handle categorical features during model training. By understanding the unique values within each category, you can decide on appropriate encoding techniques (e.g., one-hot encoding or label encoding) to transform these features into a format suitable for the machine learning model.

Replacing Missing Values: The code replaces specific values like "No internet service" and "No phone service" with a simpler "No" to ensure consistency in the data. This step helps address potential issues that might arise during model training due to inconsistencies in how missing or non-standard values are represented within a category.

Converting Yes/No to Numeric: Columns representing binary features like "Married" are converted to numeric (typically 1 for Yes and 0 for No). This transformation allows the model to understand these features quantitatively. By converting these features from strings to numerical values, the model can learn the relationships between these binary features and the target variable (customer churn) more effectively.

more effectively.

One-Hot Encoding: Columns with more than two categories (e.g., "Offer") undergo one-hot encoding. This technique creates separate binary columns for each unique category within the original column. For instance, if "Offer" has three categories ("A", "B", and "C"), one-hot encoding would result in three new binary columns, where a value of 1 indicates belonging to that specific category and 0 indicates not belonging.

Impact: Feature encoding is a critical step in preparing categorical data for machine learning models. By converting features and handling missing values consistently, the model can learn from them more effectively. One-hot encoding, in particular, improves model interpretability as it allows you to understand the influence of each individual category within a multi-category feature on the model's predictions.

Module 3: Artificial Nueral network

The provided code defines a multi-layer perceptron (MLP) neural network for binary classification using the Keras library in TensorFlow. MLPs are powerful tools for various tasks, excelling in finding complex relationships between input features and a binary output (0 or 1).

Network Layers and Activation Functions:

- Input Layer:
  - The first layer, keras.layers.Dense(26, input_shape=(70,), activation='relu'), acts as the entry point for your data. It can handle input samples with 70 features (None, 70 shape). The None in the first dimension signifies that the batch size can be flexible during training. This layer has 26 neurons, which are essentially processing units that learn patterns from the data. The activation='relu' part introduces non-linearity through the rectified linear unit (ReLU) activation function. ReLU allows the network to learn more complex relationships between features compared to linear activation.
- Hidden Layer:
  - The subsequent layer, keras.layers.Dense(15, activation='relu'), receives the processed output from the input layer. It has 15 neurons, further refining the learned features. The ReLU activation is applied again to maintain non-linearity, crucial for modeling real-world problems that are rarely linear.
- Output Layer:
  - The final layer, keras.layers.Dense(1, activation='sigmoid'), takes the output from the hidden layer. This layer has a single neuron, responsible for generating the binary classification output. The activation='sigmoid' function is well-suited for this task because it maps any real number to a value between 0 and 1, representing the probability of an instance belonging to the positive class (typically class 1).

Training and Evaluation:

- Compilation:

  - The model.compile(...) section sets up the training process. The Adam optimizer is employed with a learning rate of 0.01 to efficiently adjust the network's weights during training. The loss='binary_crossentropy' function measures the difference between the predicted probabilities and the true labels (0 or 1). It's suitable for binary classification problems. Additionally, the metrics=['accuracy'] argument tracks the model's accuracy during training, providing a basic performance indicator.

- Training:

  - The model.fit(X_train, y_train, epochs=100) line trains the model on the provided training data (X_train for features and y_train for labels) for 100 epochs (iterations). During each epoch, the model iterates through the training data, calculates the loss, and updates its internal weights to minimize the loss gradually. This process helps the network learn the patterns in the data to make accurate predictions on unseen examples.

Prediction and Thresholding:

- yp = model.predict(X_test) instructs the trained model to make predictions on the unseen test data (X_test). The predictions are stored in the yp list. These predictions are initially in the form of probabilities between 0 and 1 (due to the sigmoid activation).

- The subsequent code iterates through yp and creates a new y_pred list. It applies a threshold of 0.5. Any value in yp greater than 0.5 is classified as 1 (positive class), and those below 0.5 are classified as 0 (negative class). This is a common approach to convert the model's probabilistic outputs into definitive class labels for binary classification tasks.

Evaluation Metrics:

- print(classification_report(y_test,y_pred)) and model.evaluate(X_test, y_test) assess the model's performance on the test data. The classification_report provides a detailed breakdown of metrics like precision, recall, F1-score, and support for each class, offering valuable insights into the model's strengths and weaknesses. The model.evaluate function calculates the built-in evaluation metrics (often accuracy and loss) on the test data, giving a more concise performance summary.

F1-Score, Recall, Accuracy Calculation:

- The code extracts F1-score, recall, and accuracy from the classification report for further analysis. These metrics provide complementary perspectives on the model's performance. F1-score considers both precision (correctly predicted positives) and recall (correctly identified positive cases), offering a balanced view. Recall focuses on how well the model identifies all positive cases, while accuracy measures the overall proportion of correct predictions.

Binary Crossentropy Calculation (using NumPy):

The code calculates binary crossentropy loss using NumPy:

- The predicted values (y_pred) are converted to a NumPy array for compatibility with TensorFlow's loss functions.
- The tf.keras.losses.BinaryCrossentropy(from_logits=True) function is used to calculate the binary crossentropy loss between the true labels (y_test) and the predicted probabilities (y_pred_array). The from_logits=True argument indicates that the input values are logits (pre-activation values) rather than probabilities.
- The numpy().mean() method calculates the average binary crossentropy loss across all samples.
-

Creating a evaluation DataFrame:

- The code creates a Pandas DataFrame to store the calculated metrics for easy analysis and visualization.
- The DataFrame includes columns for the predicted values, F1-score, recall, accuracy, and binary crossentropy.
- The values are populated with the corresponding calculated metrics.

This DataFrame provides a structured way to organize and interpret the model's performance, making it easier to evaluate the effectiveness of the neural network in the given binary classification task.

Regularization Techniques

To prevent overfitting, apply regularization techniques such as L1, L2 regularization, or dropout. Dropout randomly disables a fraction of neurons during training, which helps the model generalize better by preventing it from relying too heavily on any single neuron.

Hyperparameter Tuning

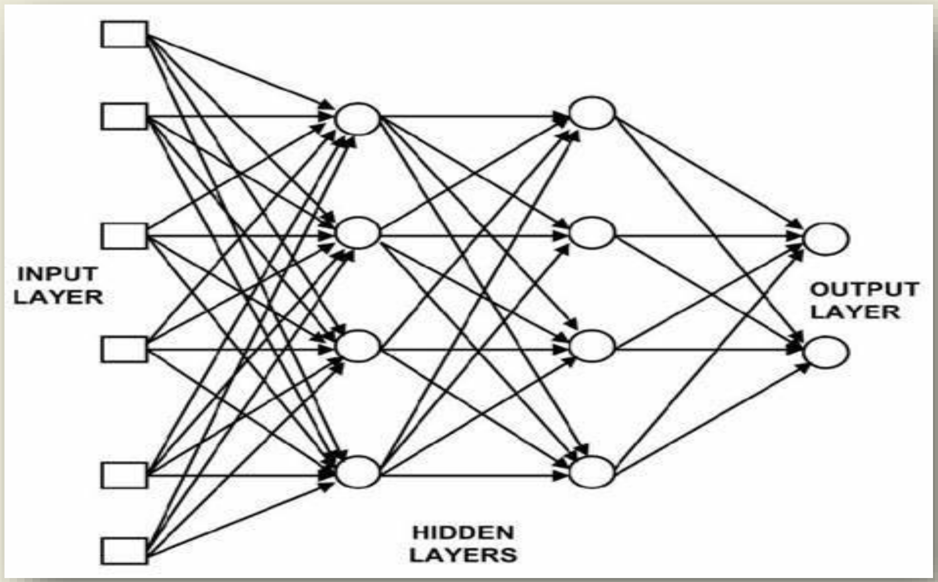
Hyperparameters like learning rate, batch size, and the number of epochs can significantly affect model performance. Use techniques like grid search or random search to find the optimal hyperparameters. Tools like Keras Tuner can automate this process and help you find the best configuration.

Optimization Algorithms

The choice of optimization algorithm can also impact the model's accuracy. While stochastic gradient descent (SGD) is a common choice, other algorithms like Adam, RMSprop, or Adagrad might converge faster and lead to better performance. Experiment with different optimizers to see which one works best for your model.

Data Augmentation

If you're working with image data, data augmentation can help improve model accuracy by artificially increasing the size of your training dataset. Techniques like rotation, scaling, and flipping can create new training examples, helping the model generalize better



Module 4: Dash Board

The objective of this dashboard is to provide a comprehensive overview of customer churn, focusing on various aspects like customer information, service usage, churn predictions, and account details. The dashboard will be built using Tableau and will leverage data from a SQL server hosted on Azure for real-time updates.
The dashboard will consist of five primary pages:
Overview Page:

- Display key performance indicators (KPIs) such as total churn rate, customer retention rate, and average revenue per user (ARPU).
- Visualize the overall churn trend over time using a line chart.
- Analyze churn rates across different customer segments (e.g., demographic, geographic, or service plan) using a bar chart
- Explore the correlation between churn and various factors (e.g., service usage, customer satisfaction) using scatter plots or correlation matrices.

Customer Page:

- Show customer information like age, gender, and location using bar charts or pie charts.
- Visualize Customer Lifetime Value (CLTV) distribution using a histogram or density plot.
- Display customer segmentation based on clustering or other techniques using scatter plots or color-coded maps.
- Track customer interactions and behaviors over time to identify potential churn indicators.

Service Page:

- Analyze service usage patterns (e.g., minutes used, data consumed) using line charts or area charts.
- Track service outages and their impact on customer churn using bar charts or heatmaps.
- Display customer satisfaction ratings for different services using star ratings or bar charts.
- Examine the effectiveness of service bundles or promotions in reducing churn.

Prediction Page:

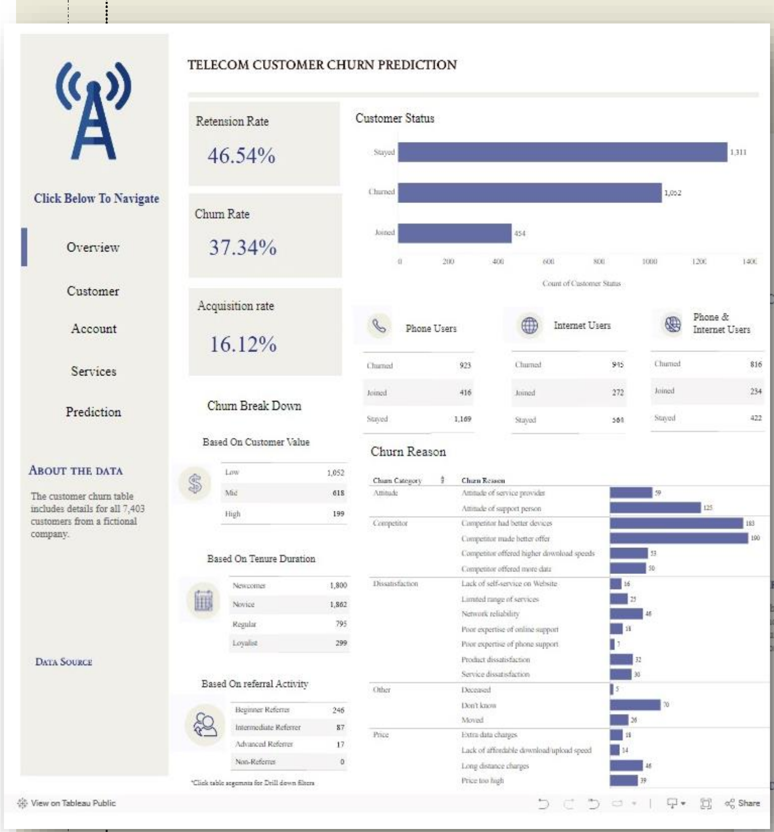
- Integrate the results from the neural network model into the dashboard.
- Visualize the predicted churn probability for each customer using a histogram or scatter plot.
- Rank customers based on their predicted churn risk using a bar chart or table.
- Evaluate the accuracy of the prediction model using metrics like precision, recall, and F1-score.

Account Page:

- Analyze account-level activities (e.g., logins, payments) using line charts or time series visualizations.
- Track contract expiration dates and potential churn risks using a calendar or timeline.
- Visualize customer support interactions (e.g., tickets, calls) using bar charts or heatmaps.
- Examine billing and payment patterns to identify potential issues that may lead to churn.

To ensure the dashboard displays the most current information, data will be extracted from the SQL server on Azure in real-time. This can be achieved using Tableau's live data connection feature or by scheduling data refreshes.

- Customer Segmentation: Experiment with different customer segmentation methods to identify target groups for retention efforts.
- Customer Lifetime Value (CLTV) Analysis: Calculate and analyze CLTV to prioritize customer retention efforts.
- Customer Journey Mapping: Visualize the customer journey to identify pain points and areas for improvement.
- Geo-Spatial Analysis: If applicable, visualize data based on geographic location to identify regional trends or patterns.
- Integration with Other Tools: Consider integrating the dashboard with other tools, such as CRM systems or marketing automation platforms, for a more holistic view of customer interactions.
- Automated Alerts: Set up automated alerts to notify relevant stakeholders of significant changes in churn rates or other key metrics.



TELECOM CUSTOMER CHURN PREDICTION

Retension Rate 46.54%

Churn Rate 37.34%

Acquisition rate 16.12%

About the data

The customer churn table includes details for all 7,403 customers from a fictional company.

Customer Status

Churn Break Down

Based On Customer Value

Low	1,052
Mid	618
High	199

Based On Tenure Duration

Newcomer	1,800
Novice	1,862
Regular	795
Loyalist	299

Based On referral Activity

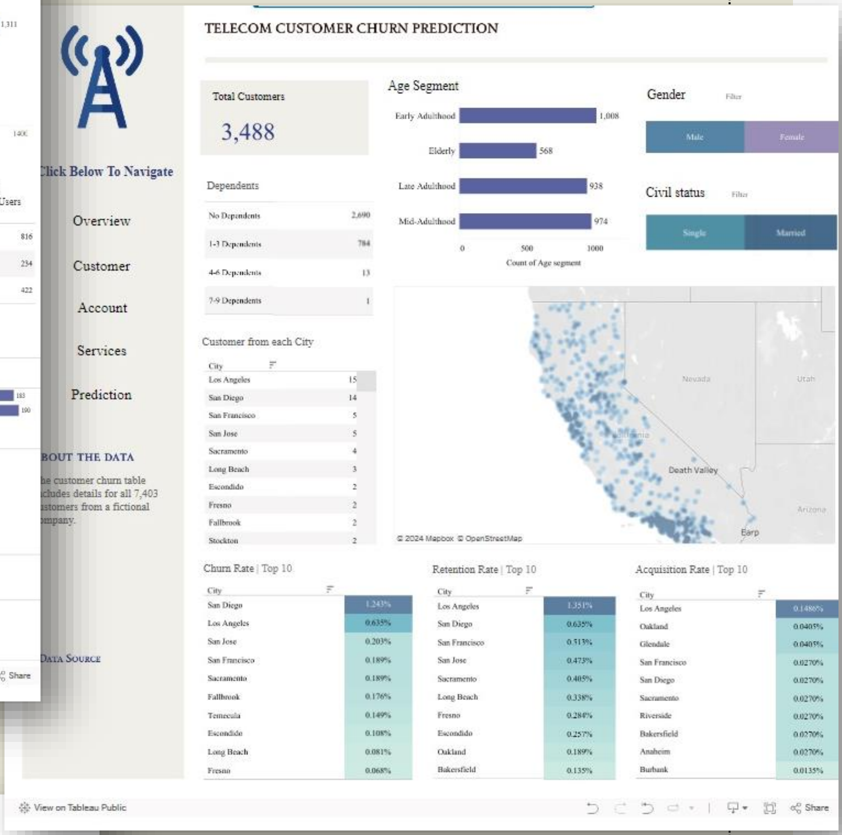
Beginner Referrer	246
Intermediate Referrer	87
Advanced Referrer	17
Non-Referrer	0

*Click table segments for Drill down filters

	Phone Users	Internet Users	Phone & Internet Users
Churned	923	945	816
Joined	416	272	234
Stayed	1,169	564	422

Churn Reason

Churn Category	Churn Reason	
Attitude	Attitude of service provider	59
	Attitude of support person	125
Competitor	Competitor had better devices	183
	Competitor made better offer	190
	Competitor offered higher download speeds	53
	Competitor offered more data	50
Dissatisfaction	Lack of self-service on Website	16
	Limited range of services	25
	Network reliability	46
	Poor expertise of online support	18
	Poor expertise of phone support	7
	Product dissatisfaction	32
	Service dissatisfaction	30
Other	Deceased	5
	Don't know	70
	Moved	26
Price	Extra data charges	18
	Lack of affordable download/upload speed	14
	Long distance charges	46
	Price too high	39



TELECOM CUSTOMER CHURN PREDICTION

Total Customers 3,488

Dependents

No Dependents	2,690
1-3 Dependents	784
4-6 Dependents	13
7-9 Dependents	1

Age Segment

Gender Filter

Civil status Filter

Customer from each City

City	
Los Angeles	15
San Diego	14
San Francisco	5
San Jose	5
Sacramento	4
Long Beach	3
Escondido	2
Fresno	2
Fallbrook	2
Stockton	2

Churn Rate | Top 10

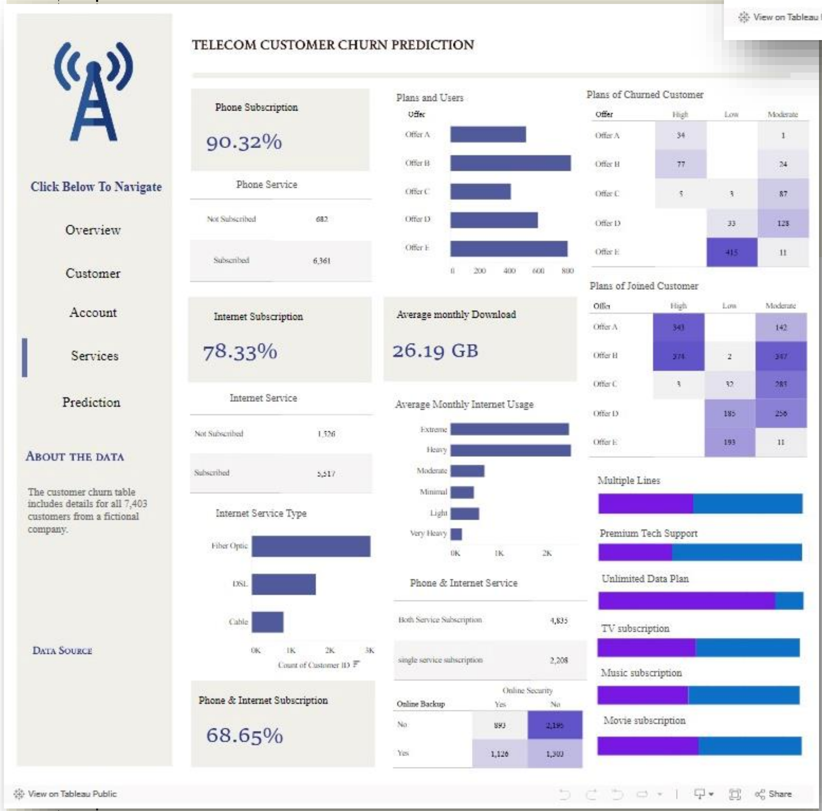
City	
San Diego	1.243%
Los Angeles	0.635%
San Jose	0.203%
San Francisco	0.189%
Sacramento	0.189%
Fallbrook	0.176%
Temecula	0.149%
Escondido	0.108%
Long Beach	0.081%
Fresno	0.068%

Retention Rate | Top 10

City	
Los Angeles	1.351%
San Diego	0.635%
San Francisco	0.513%
San Jose	0.473%
Sacramento	0.405%
Long Beach	0.338%
Fresno	0.284%
Escondido	0.257%
Oakland	0.189%
Bakersfield	0.135%

Acquisition Rate | Top 10

City	
Los Angeles	0.1486%
Oakland	0.0405%
Glendale	0.0405%
San Francisco	0.0270%
San Diego	0.0270%
Sacramento	0.0270%
Riverside	0.0270%
Bakersfield	0.0270%
Anaheim	0.0270%
Burbank	0.0135%



TELECOM CUSTOMER CHURN PREDICTION

Phone Subscription 90.32%

Phone Service

Not Subscribed	682
Subscribed	6,361

Internet Subscription 78.33%

Internet Service

Not Subscribed	1,526
Subscribed	5,517

Internet Service Type

Phone & Internet Subscription 68.65%

About the data

The customer churn table includes details for all 7,403 customers from a fictional company.

Plans and Users

Average monthly Download 26.19 GB

Average Monthly Internet Usage

Phone & Internet Service

Both Service Subscription	4,835
single service subscription	2,208

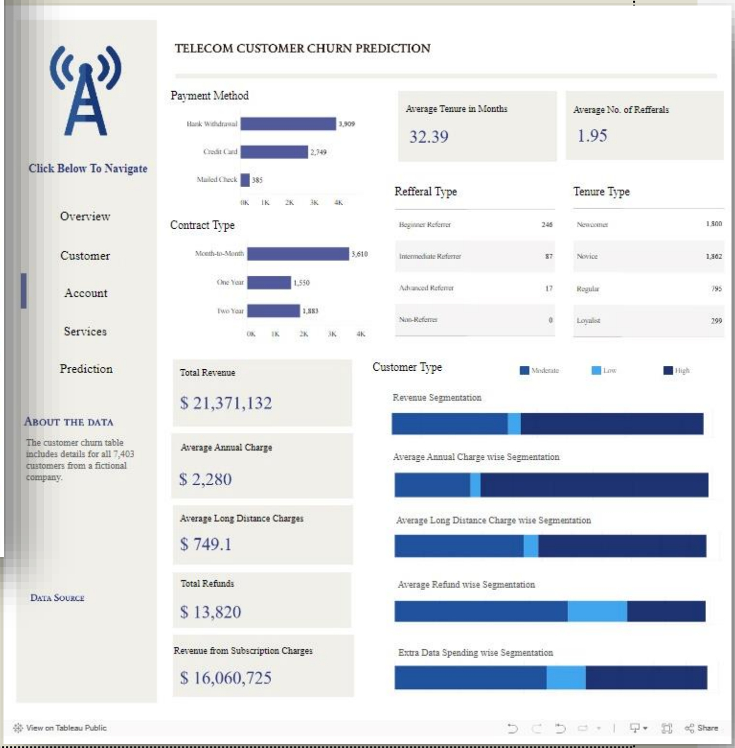
	Online Security	
Online Backup	Yes	No
No	893	2,195
Yes	1,126	1,303

Plans of Churned Customer

Offer	High	Low	Moderate
Offer A	34		1
Offer B	77		24
Offer C	5	3	87
Offer D		33	128
Offer E		415	11

Plans of Joined Customer

Offer	High	Low	Moderate
Offer A	343		142
Offer B	374	2	347
Offer C	3	32	283
Offer D		185	256
Offer E		193	11



TELECOM CUSTOMER CHURN PREDICTION

Payment Method

Contract Type

Average Tenure in Months 32.39

Average No. of Refferals 1.95

Refferal Type

Beginner Referrer	246
Intermediate Referrer	87
Advanced Referrer	17
Non-Referrer	0

Tenure Type

Newcomer	1,800
Novice	1,862
Regular	795
Loyalist	299

Total Revenue $ 21,371,132

Average Annual Charge $ 2,280

Average Long Distance Charges $ 749.1

Total Refunds $ 13,820

Revenue from Subscription Charges $ 16,060,725

About the data

The customer churn table includes details for all 7,403 customers from a fictional company.

Customer Type

Moderate
Low
High
Revenue Segmentation
Average Annual Charge wise Segmentation
Average Long Distance Charge wise Segmentation
Average Refund wise Segmentation
Extra Data Spending wise Segmentation

Modules 5: Azure Pipeline

stream Data to Event Hub
Azure Event Hubs is a big data streaming platform and event ingestion service. It can receive and process millions of events per second.

- Create an Event Hub: In the Azure portal, create an Event Hub namespace and an Event Hub within it.
- Send Data to Event Hub: Use an SDK or Azure CLI to send your streaming data to the Event Hub.

Process Data with Stream Analytics
Azure Stream Analytics is a real-time analytics and complex event-processing engine.

- Create a Stream Analytics Job: In the Azure portal, create a Stream Analytics job.
- Input Configuration: Set the Event Hub as the input source for the Stream Analytics job.
- Query Definition: Write a Stream Analytics query to process the incoming data. For example, you can filter, aggregate, or join data streams.
- Output Configuration: Set the output to Azure Synapse Analytics.

Store Processed Data in Synapse Analytics
Azure Synapse Analytics is a limitless analytics service that brings together big data and data warehousing.

- Create a Synapse Workspace: In the Azure portal, create a Synapse workspace.
- Create a Dedicated SQL Pool: This will be used to store the processed data.
- Connect Stream Analytics to Synapse: Configure the Stream Analytics job to output data to the Synapse SQL pool.

Archive Data in Blob Storage
Azure Blob Storage is Microsoft's object storage solution for the cloud.

- Create a Storage Account: In the Azure portal, create a storage account and a blob container within it.
- Configure Synapse to Archive Data: Set up a pipeline in Synapse to periodically move data to Blob Storage for archival purposes.

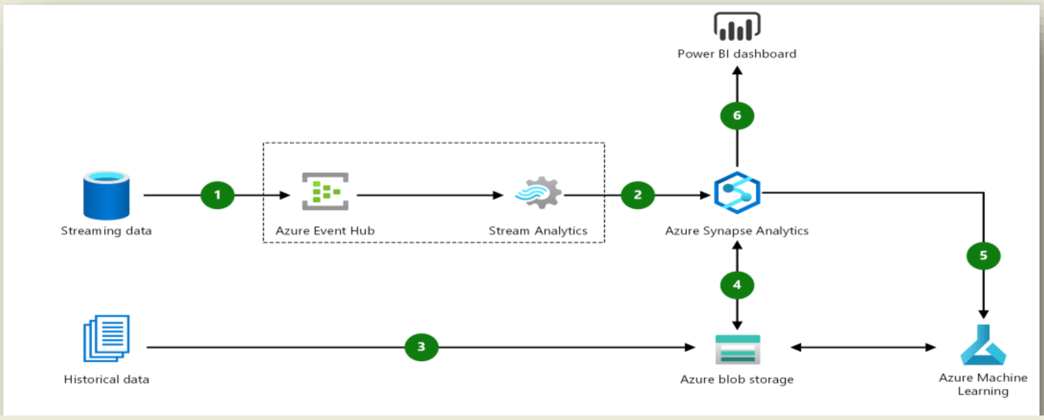
Use Azure ML for Predictions
Azure Machine Learning is a cloud-based environment you can use to train, deploy, automate, and manage machine learning models.

- Register Your Model: Register your trained ANN model in the Azure ML workspace.
- Create an Inference Pipeline: Set up an inference pipeline to use the model for predictions on the incoming data.

- Deploy the Model: Deploy the model as a web service or a real-time endpoint.

Visualize Data with Power BI
Power BI is a business analytics service that provides interactive visualizations and business intelligence capabilities.

- Connect Power BI to Synapse: Use the Power BI desktop to connect to your Synapse SQL pool.
- Create Reports and Dashboards: Build interactive reports and dashboards to visualize the customer churn predictions and other insights.



Module 6: web App

First, ensure you have Python and Streamlit installed on your machine. Streamlit is a powerful tool for creating web apps with minimal effort, making it perfect for this task.

Preparing Your Model

You already have a trained ANN model for predicting customer churn. Save this model using a library like joblib or pickle. This allows you to load the model later in your Streamlit app for making predictions.

Creating the Streamlit App

1. Import Libraries: Start by importing the necessary libraries, including Streamlit, pandas for handling data, and joblib for loading your model.

2. App Layout: Design the layout of your app. Begin with a title and a section to display some sample data. This helps users understand the kind of data your model works with. Load a sample dataset and display the first few rows.

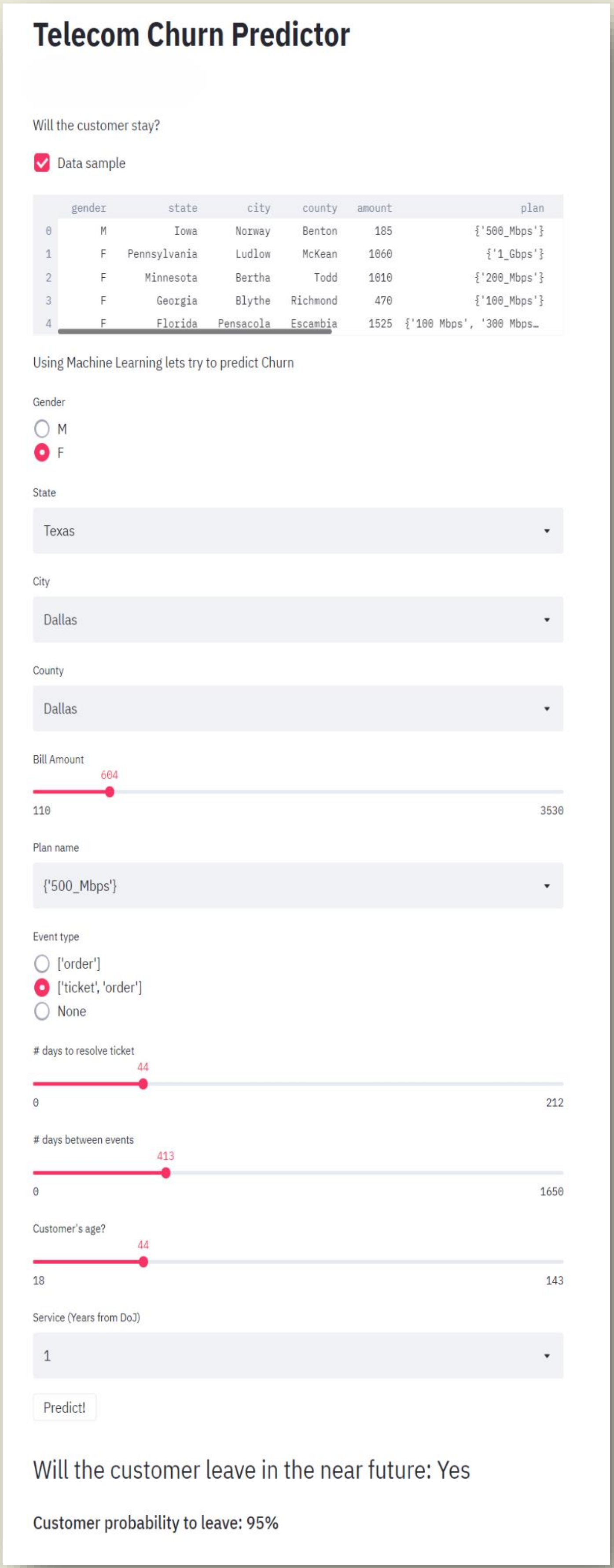
3. User Input: Create input fields for users to enter customer details. For example, you might ask for age, income, and tenure. Streamlit provides easy-to-use widgets for this purpose, such as sliders and number input fields.

4. Making Predictions: Add a button that, when clicked, takes the user input, processes it, and uses your ANN model to make a prediction. The model will output the probability of customer churn.

5. Displaying Results: Show the prediction result to the user. This could be the probability of churn expressed as a percentage, giving users a clear understanding of the risk.

Running the App

Once your app is set up, you can run it using a simple command in your terminal. This will start a local server, and you can interact with your app through a web browser.



Telecom Churn Predictor

Will the customer stay?

☑ Data sample

	gender	state	city	county	amount	plan
0	M	Iowa	Norway	Benton	185	{'500_Mbps'}
1	F	Pennsylvania	Ludlow	McKean	1060	{'1_Gbps'}
2	F	Minnesota	Bertha	Todd	1010	{'200_Mbps'}
3	F	Georgia	Blythe	Richmond	470	{'100_Mbps'}
4	F	Florida	Pensacola	Escambia	1525	{'100 Mbps', '300 Mbps…

Using Machine Learning lets try to predict Churn

Gender: M [✓ F]

State: Texas

City: Dallas

County: Dallas

Bill Amount: 604 (110 – 3530)

Plan name: {'500_Mbps'}

Event type: ['order'] [✓ ['ticket', 'order']] None

days to resolve ticket: 44 (0 – 212)

days between events: 413 (0 – 1650)

Customer's age?: 44 (18 – 143)

Service (Years from DoJ): 1

Predict!

Will the customer leave in the near future: Yes

Customer probability to leave: 95%

Project Significance

Current Methodologies in Churn Prediction

Churn prediction is a critical aspect of customer relationship management, aiming to identify customers who are likely to discontinue using a service or product. Traditional methodologies predominantly utilize machine learning models such as logistic regression, decision trees, support vector machines (SVM), and ensemble learning techniques like random forests and gradient boosting. These models analyze a plethora of historical customer data, including usage patterns, transaction history, and demographic information, to pinpoint key indicators of churn. For example, logistic regression might be employed to estimate the probability of a customer churning based on their past behavior, while decision trees can elucidate the decision paths leading to churn. Despite their effectiveness, these models often encounter significant challenges. Handling large volumes of data, integrating data from multiple disparate sources, and providing real-time predictions are some of the common hurdles. Additionally, these traditional models may struggle with capturing complex, non-linear relationships within the data, potentially leading to less accurate predictions.

Leveraging Cloud-Based ANN and Dashboards

Your innovative cloud-based Artificial Neural Network (ANN) and dashboard system effectively addresses several limitations inherent in traditional churn prediction methods. By harnessing the power of cloud computing, your system can effortlessly manage vast amounts of data from various sources, ensuring that the data remains up-to-date and is accessible from any location. The ANN, with its advanced capability to learn intricate patterns and relationships within the data, offers more precise and nuanced predictions compared to traditional models. This is particularly beneficial in capturing complex, non-linear interactions that are often missed by conventional methods. Furthermore, the integrated dashboard provides real-time insights and visualizations, enabling stakeholders to monitor churn indicators continuously and take proactive measures swiftly. This combination not only enhances the accuracy of predictions but also significantly improves decision-making efficiency. By offering a comprehensive, real-time view of customer behavior, your system empowers businesses to implement timely interventions, thereby reducing churn rates and enhancing customer retention.

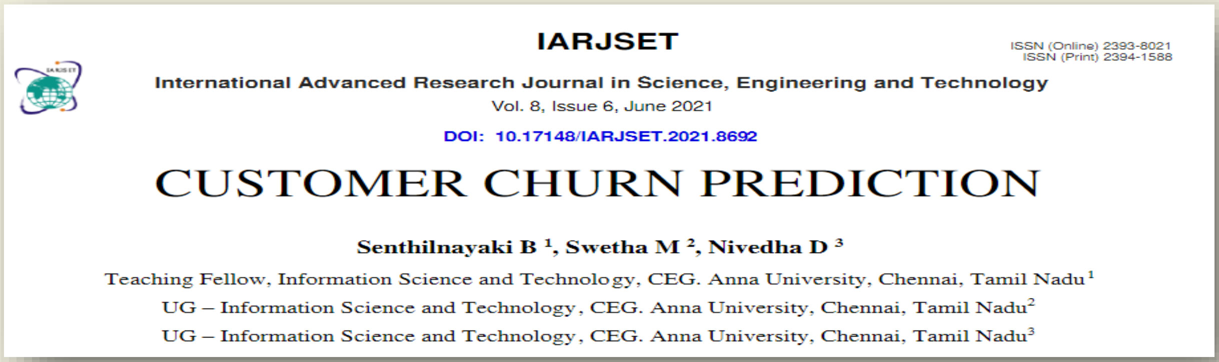
Literature Survey

Customer Churn Prediction: A Comparative Analysis of Machine Learning Algorithms

Customer churn analysis and prediction in the telecom sector is an important area of research due to the high cost of acquiring new customers compared to retaining existing ones (e.g., Senthilnayaki B, Swetha M, Nivedha D). Machine learning techniques such as Logistic Regression, Random Forest, and K-Nearest Neighbors are commonly used to build churn prediction models

Authors: **Senthilnayaki B, Senthilnayaki B, Senthilnayaki B.**

This study aims to develop and evaluate the performance of various machine learning models for predicting customer churn in the telecom industry. By comparing the accuracy, precision, recall, and F1-score of Logistic Regression, Random Forest, and K-Nearest Neighbors algorithms, we aim to identify the most effective model for predicting customer churn and providing valuable insights for targeted retention strategies.



IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 8, Issue 6, June 2021

DOI: 10.17148/IARJSET.2021.8692

CUSTOMER CHURN PREDICTION

Senthilnayaki B [1], Swetha M [2], Nivedha D [3]

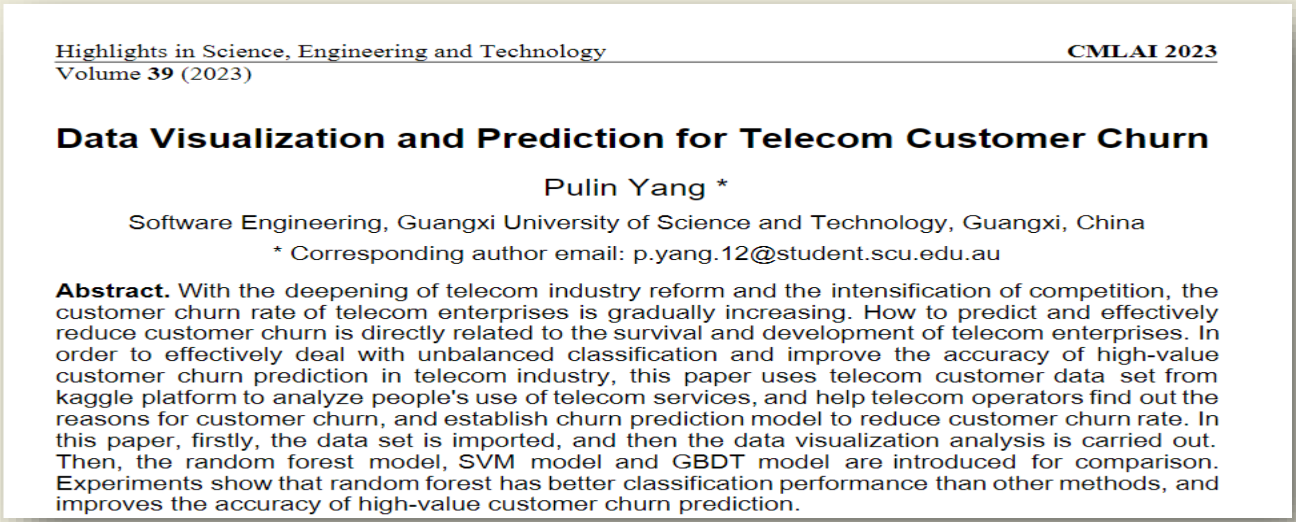
Teaching Fellow, Information Science and Technology, CEG. Anna University, Chennai, Tamil Nadu[1]
UG – Information Science and Technology, CEG. Anna University, Chennai, Tamil Nadu[2]
UG – Information Science and Technology, CEG. Anna University, Chennai, Tamil Nadu[3]

Enhancing Customer Loyalty in the Telecom Industry: A Machine Learning Approach

This paper, presented at the Highlights in Science, Engineering and Technology CMLAI conference (Volume 39, **2023**), investigates the application of machine learning algorithms for predicting customer churn in the telecom industry.

Author:**Pulin Yang, from Guangxi University of Science and Technology**, China, emphasizes the growing importance of customer retention strategies due to rising competition and increasing churn rates. The study focuses on developing a churn prediction model that not only identifies customers at risk of leaving but also prioritizes high-value customers for targeted retention efforts.

The research utilizes a telecom customer dataset from Kaggle to analyze usage patterns and identify factors contributing to churn. By employing data visualization techniques, the study explores trends and relationships within the data. Subsequently, the paper compares the performance of three machine learning models: Random Forest, Support Vector Machine (SVM), and Gradient Boosting Decision Tree (GBDT). The results demonstrate that the Random Forest model achieves superior classification accuracy, particularly in predicting churn among high-value customers. This finding suggests that Random Forest offers a valuable tool for telecom operators to optimize their customer retention strategies and minimize churn rates.


Highlights in Science, Engineering and Technology CMLAI 2023
Volume 39 (2023)

Data Visualization and Prediction for Telecom Customer Churn

Pulin Yang *

Software Engineering, Guangxi University of Science and Technology, Guangxi, China

* Corresponding author email: p.yang.12@student.scu.edu.au

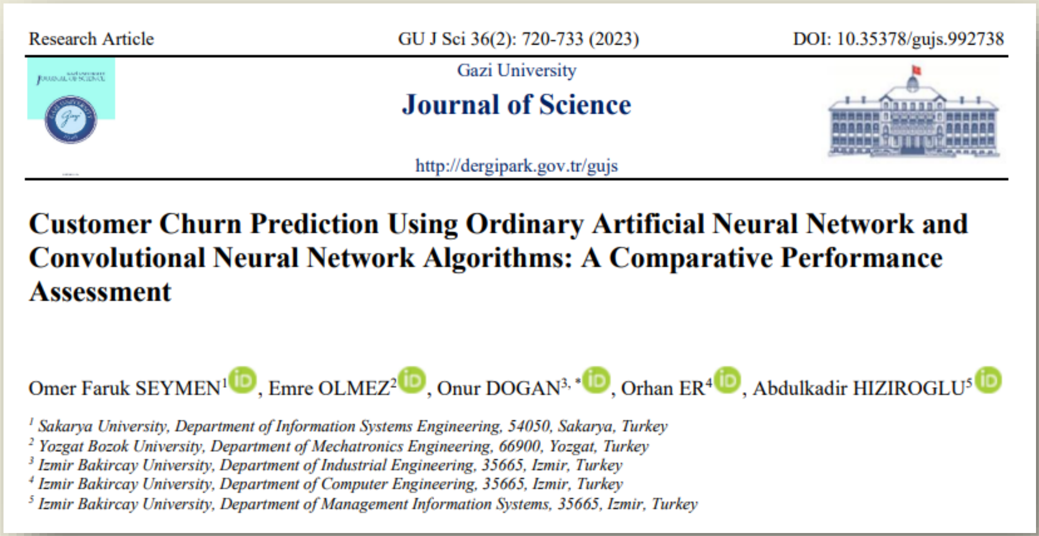
Abstract. With the deepening of telecom industry reform and the intensification of competition, the customer churn rate of telecom enterprises is gradually increasing. How to predict and effectively reduce customer churn is directly related to the survival and development of telecom enterprises. In order to effectively deal with unbalanced classification and improve the accuracy of high-value customer churn prediction in telecom industry, this paper uses telecom customer data set from kaggle platform to analyze people's use of telecom services, and help telecom operators find out the reasons for customer churn, and establish churn prediction model to reduce customer churn rate. In this paper, firstly, the data set is imported, and then the data visualization analysis is carried out. Then, the random forest model, SVM model and GBDT model are introduced for comparison. Experiments show that random forest has better classification performance than other methods, and improves the accuracy of high-value customer churn prediction.

Customer Churn Prediction: A Comparative Analysis of Ordinary Artificial Neural Networks and Convolutional Neural Networks

Authors: Omer Faruk Seymen, Abdulkadir Hiziroglu İzmir ,Orhan Er

This paper, published in the Gazi University Journal of Science, explores the application of both ordinary artificial neural networks (ANNs) and convolutional neural networks (CNNs) for predicting customer churn in the retail sector. The study aims to compare the performance of these two deep learning approaches and evaluate their effectiveness in handling the challenges associated with imbalanced datasets and complex customer behavior patterns.

By transforming customer transaction and demographic data into an image format, the authors leverage the unique capabilities of CNNs for feature extraction and classification. The experimental results demonstrate that the CNN-based model outperforms traditional churn prediction methods, highlighting the potential of deep learning techniques in enhancing customer retention strategies.


Research Article GU J Sci 36(2): 720-733 (2023) DOI: 10.35378/gujs.992738

Gazi University
Journal of Science
http://dergipark.gov.tr/gujs

Customer Churn Prediction Using Ordinary Artificial Neural Network and Convolutional Neural Network Algorithms: A Comparative Performance Assessment

Omer Faruk SEYMEN[1], Emre OLMEZ[2], Onur DOGAN[3,*], Orhan ER[4], Abdulkadir HIZIROGLU[5]

[1] Sakarya University, Department of Information Systems Engineering, 54050, Sakarya, Turkey
[2] Yozgat Bozok University, Department of Mechatronics Engineering, 66900, Yozgat, Turkey
[3] Izmir Bakircay University, Department of Industrial Engineering, 35665, Izmir, Turkey
[4] Izmir Bakircay University, Department of Computer Engineering, 35665, Izmir, Turkey
[5] Izmir Bakircay University, Department of Management Information Systems, 35665, Izmir, Turkey

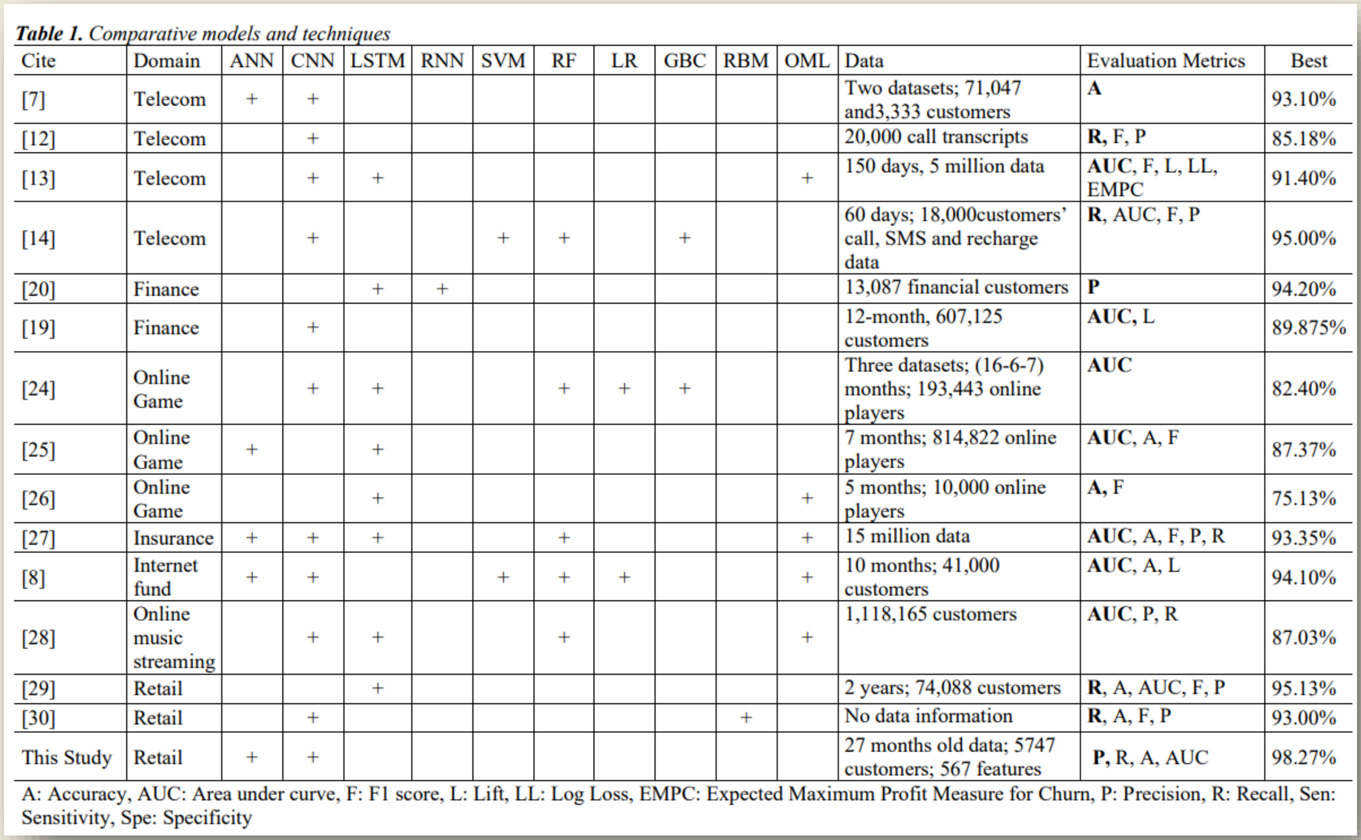


Table 1. *Comparative models and techniques*

Cite	Domain	ANN	CNN	LSTM	RNN	SVM	RF	LR	GBC	RBM	OML	Data	Evaluation Metrics	Best
[7]	Telecom	+	+									Two datasets; 71,047 and3,333 customers	**A**	93.10%
[12]	Telecom		+									20,000 call transcripts	**R,** F, P	85.18%
[13]	Telecom		+	+							+	150 days, 5 million data	**AUC**, F, L, LL, EMPC	91.40%
[14]	Telecom		+			+	+		+			60 days; 18,000customers' call, SMS and recharge data	**R**, AUC, F, P	95.00%
[20]	Finance			+	+							13,087 financial customers	**P**	94.20%
[19]	Finance		+									12-month, 607,125 customers	**AUC,** L	89.875%
[24]	Online Game		+	+			+	+	+			Three datasets; (16-6-7) months; 193,443 online players	**AUC**	82.40%
[25]	Online Game	+		+								7 months; 814,822 online players	**AUC**, A, F	87.37%
[26]	Online Game			+							+	5 months; 10,000 online players	**A,** F	75.13%
[27]	Insurance	+	+	+			+				+	15 million data	**AUC**, A, F, P, R	93.35%
[8]	Internet fund	+	+			+	+	+			+	10 months; 41,000 customers	**AUC**, A, L	94.10%
[28]	Online music streaming		+	+			+				+	1,118,165 customers	**AUC**, P, R	87.03%
[29]	Retail			+								2 years; 74,088 customers	**R**, A, AUC, F, P	95.13%
[30]	Retail		+							+		No data information	**R**, A, F, P	93.00%
This Study	Retail	+	+									27 months old data; 5747 customers; 567 features	**P,** R, A, AUC	98.27%

A: Accuracy, AUC: Area under curve, F: F1 score, L: Lift, LL: Log Loss, EMPC: Expected Maximum Profit Measure for Churn, P: Precision, R: Recall, Sen: Sensitivity, Spe: Specificity

Table 1 summarizes the related work in the field of customer churn prediction, highlighting the performance metrics used and the best-performing models. As noted, most deep learning-based studies in this area have focused on the telecommunications sector, where contractual relationships exist between customers and companies.

Deep Learning Applications:

- Feedforward Neural Networks (FNNs): [7] compared FNNs (both small and large versions) with ANNs and CNNs for telecom churn prediction, using two different datasets. Accuracy was the primary metric used for comparison.
- Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM): [13] developed a daily churn prediction model using CNN and LSTM, outperforming RFM-based models. The models were evaluated using AUC, F1 Score, Lift, Log Loss, and EPMC.
- CNN for Image-Based Prediction: [14] transformed call details, SMS, Recharge, and Data usage records into images and used CNN for churn prediction. Precision, recall, and F1-scores were compared against SVM, Random Forest, and Gradient Boosting Classifier.
- CNN with Natural Language Processing (NLP): [12] combined CNN with NLP to analyze customer call transcripts and predict churn. Precision, recall, and F1-score were used for evaluation.

Other Deep Learning Studies:

- CNN and Deep CNN: [15, 17] used CNN and Deep CNN for telecom customer churn prediction. They evaluated performance using Confusion Matrix outputs, accuracy, AUC, loss margin, maximum dice coefficient, and Jaccard coefficient.
- Vector Embedding Model: [18] used a Vector Embedding Model for loss estimation in a telecom dataset. Accuracy and F1-score were reported.
- CNN in Financial Services: [19, 20] applied CNN models to predict churn in the financial service domain. [19] used textual and structural data, while [20] compared RNN and LSTM models. Evaluation metrics included Precision, AUC, and TDL.
- Deep Neural Network (DNN): [21] compared DNN with MPL for banking customer churn prediction. Accuracy and loss margin were used, along with different activation functions and training algorithms.

Other Machine Learning Approaches:

- SVM with Bayesian Optimization: [22] optimized SVM kernels using Bayesian Optimization. Linear, Polynomial, Radial, and Sigmoid kernels were compared using accuracy, precision, recall, and F1-score.
- Extended Convolutional Decision Tree (ECDT): [23] applied ECDT for retail employee churn prediction. Grid Search Optimization was used to improve accuracy, which was compared against ECDT, KNN, CNN, SVM, NB, and DT.
- Machine Learning Models for Online Gaming: [24-26] used various machine learning models to predict churn in the online gaming industry, based on player behavior data. Accuracy, AUC, and F1-score were used for evaluation.

CNN Enhancements:

- Shallow-Based CNN: [27] combined deep and shallow learning models for churn prediction.
- LSTM Integration: [28] added LSTM output as an input to a CNN model.
- Static Data Integration: [8] integrated static data (demographics) with dynamic features (RFM and relationship length) in a CNN model.

Retail Domain Studies:

- LSTM for Smartphone Buyer Churn: [29] used LSTM to predict churn based on questionnaire responses from smartphone buyers. AUC, Precision, and Recall were used for evaluation.
- CNN and Restricted Boltzmann Machine: [30] compared CNN and Restricted Boltzmann Machine for customer churn prediction in the retail domain. Various performance metrics were used for evaluation.
- While there are studies on data mining and machine learning in the retail domain, deep learning applications are still relatively limited.

Conclusion

This project successfully developed a sophisticated real-time customer churn prediction system, demonstrating the power of Artificial Neural Networks (ANNs) and the scalability of Azure cloud infrastructure. The system, equipped with a comprehensive dashboard and a user-friendly web application, accurately identifies at-risk customers, empowering businesses to proactively address churn and enhance customer retention.

By leveraging the capabilities of ANNs, the model effectively predicts customer churn based on a diverse set of factors, including historical usage patterns, demographic data, and recent customer interactions. The Azure platform provides a scalable and resilient environment for real-time data processing and model deployment, ensuring the system's ability to handle increasing workloads and adapt to evolving customer behaviors.

The integrated dashboard offers a clear visualization of key performance indicators, enabling stakeholders to monitor the system's effectiveness and identify emerging trends. The web application provides a user-friendly interface for accessing churn predictions, empowering businesses to take timely actions to retain their valuable customers.

Beyond its immediate benefits, this project serves as a proof of concept for the application of advanced machine learning techniques in customer retention. The system's success highlights the potential of AI-driven solutions to transform business operations and drive growth. Future enhancements could include incorporating additional features, such as customer sentiment analysis and predictive marketing campaigns, to further refine the churn prediction capabilities and provide even more valuable insights to businesses.

In conclusion, this project demonstrates the efficacy of ANNs in predicting customer churn and the advantages of deploying such models on cloud platforms like Azure. The system has the potential to significantly enhance customer retention rates and drive business growth. By harnessing the power of AI and leveraging the scalability of cloud technology, businesses can gain a competitive edge and deliver exceptional customer experiences