📊 Customer Churn Prediction Report

📋 1. Executive Summary

Overview of the Project

Predicting customer churn remains a pressing concern for telecom firms looking to retain subscribers. This project homes in on identifying at-risk customers for a telco using an Artificial Neural Network model. The overarching goal is enabling proactive retention initiatives by singling out those most likely to terminate their service. Through sophisticated data analysis and machine learning methods, we aspire to craft a sturdy predictive system offering actionable guidance. This would bolster efforts to craft retention tactics better catering to subscribers on the cusp of churn. By tapping the ANN model's capabilities and the insights gleaned, management can reevaluate strategies to keep valued customers in the fold.

Key Outcomes

The ANN model developed in this project demonstrated strong predictive capabilities. Key performance metrics include:

- Accuracy measures how often the model was correct when predicting whether customers would churn or not churn. At 78%, it accurately categorized most customers.

- Precision reveals how frequently customers labeled as churners by the model ended up churning. At 68%, over half the customers the model predicted would churn did so.

- Recall displays the proportion of actual churners correctly identified by the model. At just 32%, less than half of churners were recognized as such.

- The F1-score balances precision and recall, judging the model's overall ability to forecast churning. Weighing both false positives and negatives, it achieved a score of 44%, indicating room for improvement in concurrently maximizing correctness and completeness.

The readings indicate that the model can effectively tell the predicted churn customer and those who are not, and now we can move further refinement for deployment in real life situations on this basis.

Insights Gained

The analysis found several key factors that influence customer churn: customer service interactions, types of contracts, monthly charges, and tenure. Understanding these critical attributes enhances our ability to develop targeted strategies for lowering churn rates.

- 📊 **Customer Service Interactions:** The vast amount of customer service contacts proved predictive of potential churn, pinpointing opportunities to better serve clients and strengthen loyalty. Frequent calls and queries exposed dissatisfaction demanding remedy if defection were to be prevented.

- 📈 **Contract Type:** Individuals bound by brief, month-by-month commitments demonstrated heightened churn risk relative to those settled under longer-term accords, implying incentives for extended engagements may stem the outflow. Strategically extending engagements' duration could curtail turnover's tide.

• 💰 **Monthly Charges:** Higher monthly charges were linked to higher churn rates. This may suggest that businesses need more competitive pricing strategies or added-value propositions.

• 🕒 **Tenure:** Customers with shorter tenure are more possible to churn. Thus, it is important to have well-thought-out approaches to engage customers within the first few months of the sale.

The results presented within this project offer necessary data that could be used to create better customer retention strategies. Such a framework would let companies ensure that their customers are more satisfied and, therefore, provide better profits.

📝 2. Introduction

2.1 Background

Overview of Customer Churn

Customer churn or customer attrition refers to when clients cease utilizing an enterprise's products and services, opting instead to switch their loyalty and trade to a rival company. In the telecommunications sphere, such churn presents a serious predicament owing to the aggressive competition between providers and the convenience with which subscribers can alter carriers. Exceedingly high rates of churn risk resulting in massive losses of earnings, augmented costs of attracting new customers, and diminished profitability. Grasping the triggers that provoke client churn and anticipating which patrons may potentially terminate their contracts has immense significance for retaining a stable customer base and ensuring the long haul success of the business venture.

Importance of Predictive Modeling

Predictive modeling helps find out which customers are likely to churn based on their historical data and patterns. Predictive models, in turn, leverage such capabilities by allowing very sophisticated machine learning algorithms like Artificial Neural Networks (ANNs) to analyze a plethora of customer behaviour and interaction characteristics to predict the probability that customer churn.

It helps business take preemptive steps to hold on high churn risk customers by providing tailor-made offers or it can work as a service quality improvement channel, customer engagement strategy enhancers etc. In addition to churn reduction, predictive modeling is widely applicable across many industries and functions from optimizing resource allocation to right-sizing marketing mix.

This introduction covers all the difficulties that there are around customer churn and brings in some idea just how critical predictive modeling can be to solve this problem before moving forward with what our project is about.

📝 2.2 Objectives

Project Goals 🎯

The goal of this project is to construct and test an Artificial Neural Network (ANN) model that can predict customer churn in the telecommunications sector. The ANN model goal is to correctly identify by historical data on the customer which are more likely to leave from that service. The project aims to deliver actionable insights, based on the predictive ability of this model that can allow them to devise targeted customer retention strategies and prevent churn which will eventually lead increased customer satisfaction and business profitability.

Scope of the Project 📋

- **Included Aspects**:
  - **Data Preprocessing:** Cleaning, normalising and encoding the data which made it ready for model training. Steps involved in this process were:Handling missing valuesScaling of numerical featuresEncoding for categorical variables
  - **Model Development**: The project aimed to create an ANN model specifically developed for customer churn prediction. This included choosing the model architecture, hyper-parameters tuning and training the filtered data.
  - **Model Evaluation**: We analysed the ANN model through major dimensions success metrics like, accuracy, precision, recall and F1-score. The model's prediction analysis during the evaluation phase to understand what crucial factors determines churn.
- **Excluded Aspects**:
  - **Real-Time Deployment**: The project did not cover deploying the ANN model in a real-time production environment. The focus was solely on model development and evaluation.
  - **Integration with Business Systems**: There was no integration of the model with existing customer management systems or other business tools. The project scope was limited to developing a standalone predictive model.

📝 3. Dataset Description

Source of the Dataset 📊

The dataset used in this project was sourced from **Advanced Consultant Service**, specifically for a customer churn analysis in the telecommunications sector. This dataset includes detailed records of customer demographics, account information, and service usage patterns, all of which are critical for developing an effective predictive model.

Data Characteristics 🔍

- **Number of Records**: The dataset contains complete and client instances, which provides ample data for training and testing an Artificial Neural (ANN) Model. This comprehensive data set ensures that the model will work and gives credible forecasts of churn rate.

- **Nature of the Data**:
    - **Numerical Data**:
        - The dataset includes several numerical features that quantify customer attributes and behaviours. Examples of numerical data include:
            - **Tenure**: How long a customer has been with the company in months
            - **Monthly Charges**: The amount charged to the customer on a monthly basis.
            - **Total Charges**: The total amount the customer has been charged since they first signed up
    - **Categorical Data**:
        - The dataset also includes categorical features that describe various Aspects of customer accounts and services, such as:
            - **Contract Type**: Indicates the customer's contract duration (e.g. month-to-month, one-year, two years).
            - **Payment Method**: The way the customer Has been paying his bill (e.g. electronic check, mailed check, bank transfer, credit card).

- **Additional Services**: Whether the customer has subscribed to services like online security, tech support, streaming TV.

- **Important Attributes Related to Customer Churn**:

  - **Churn Status**: The major target variable, represented by two vectors (Churn_Yes and Churn_No),rows in or out of service. Churn status indicates whether a customer has churned or not. This binary classification is very important for predictive modeling tasks.

  - **Customer Demographics**: Attributes such as gender, senior citizen status and partner and dependent status, give us a better understanding of who our customers are and help us to find out what demographic trends are related churn.

  - **Service Usage**: These features relate to the services that customers use, e.g. internet and phone. They also include whether they obtain further add-ons like online back-up service or device insurance. Such attributes are helpful in identifying service usage patterns predictive of customer churn.

📝 4. Data Preprocessing

Data Cleaning 🧹

- **Handling Missing Values**:

  - We cleaned the dataset, and fine-tuned it to be used as an input for training. To handle missing values we followed the traditional approach of identifying and filling in those cells with suitable strategies like using mean for numerical data, mode for categorical. This makes sure that your data is neither corrupted nor biased towards a particular class and maintains the model efficacy.

- **Duplicate Records**:

  - **103 duplicate rows** were detected while data cleaning. We considered the set of duplicates listed below for a while, but ultimately chose **not to remove them** due to potential approval in regards valid repeat transaction Removing them can cause loss of crucial data particularly in instances where recurrent transactions may tell the way a customer makes use of these services.

Feature Engineering 🔨

- **Creating New Features**:
    - New features that capture critical customer behaviours and traits were created through feature engineering to improve the models. For instance:
        - **Tenure Group**: Made by grouping the customers based on their tenure to provide more context from the customer longevity and how it affects churn.
        - **Total Service Count**: This variable simply represented the number of different services a customer subscribed to (e.g., internet, phone, streaming services) as mentioned before we wanted to know whether broader use of service was associated with higher churn.
    - **Choice of Features**:
        - The new features were selected based on their potential to add predictive power to the model. Features like **Tenure Group** help in segmenting customers by their tenure, which is often linked with churn. **Total Service Count** helps in identifying if customers with multiple services are more or less likely to churn.

Data Splitting ✂️

- **Training and Testing Split**:
    - The dataset was split into training and testing sets to evaluate the model's performance effectively. An **80/20 split** was used, meaning 80% of the data was allocated for training the model, and the remaining 20% was reserved for testing. This split ratio ensures that the model is trained on a substantial portion of the data while still leaving a significant portion for evaluating its performance.
    - **Consistency in Data Preparation**:

- Both training and testing datasets included all features necessary for model development and validation, ensuring consistency in the data used across different stages of the model lifecycle. This uniform approach helps in assessing the model's generalization capabilities more accurately.

📝 5. Model Development

Model Selection 🧠

- **Choosing an Artificial Neural Network (ANN)**:

    - An **Artificial Neural Network (ANN)** was chosen for its ability to capture intricate patterns within intricate data. Predicting customer churn involves deciphering subtle factors like behaviour, utilisation, and demographics which are frequently convoluted and codependent. ANNs, with their many layers and neutrons, are well-equipped to tackle such complexities. The network learns from information to deliver precise predictions, outperforming traditional models which sometimes overlook important interactions.

    - The decision to employ an ANN was based on the nature of the issue: anticipating whether clients will stay is understanding subtle and varied elements for example customer conduct, administration utilisation, and statistical profile subtleties, which regularly nonlinear and interconnected. Numerous layered ANNs have the capacity to handle such complexities and can gain from the information to give exact anticipations superior to anything customary direct models.

Defining the ANN Architecture 🏗️

- **Architecture Overview**:

- The architecture of the ANN model was carefully designed to balance complexity and performance:

  - **Input Layer**: The model starts with an input layer that takes in the features from the dataset, specifically tailored to the number of input features (14 in this case).

  - **Hidden Layers**:

    - **First Hidden Layer**: Comprises **64 units** with a **ReLU (Rectified Linear Unit)** activation function. This layer is designed to capture the initial complexity of the data by learning high-dimensional feature representations.

    - **Second Hidden Layer**: Contains **32 units** with a **ReLU** activation function, serving to refine the learned features and reduce dimensionality while preserving important information.

    - **Third Hidden Layer**: Features **16 units** with a **ReLU** activation function, further fine-tuning the model's understanding of the data.

    - **Dropout Layers**: Positioned after the first and second hidden layers with a dropout rate of **0.2**. These layers help prevent overfitting by randomly setting a fraction of the input units to zero during training, ensuring that the model generalizes well to unseen data.

  - **Output Layer**: Comprises a single unit with a **sigmoid** activation function, producing a probability score indicating the likelihood of a customer churning. This score is thresholded to classify customers into churned or not churned categories.

Compilation and Configuration ⚙️

- **Model Compilation**:

  - **Optimizer**: The **Adam** optimizer was chosen for its efficiency and adaptive learning rate capabilities, which help in converging to an optimal solution faster and more reliably than standard gradient descent.

  - **Loss Function**: **Binary Cross-Entropy** was used as the loss function, which is suitable for binary classification problems like churn prediction. It measures the dissimilarity between the predicted probability and the actual class label (churn or no churn), guiding the optimization process to minimize this difference.

- **Metrics**: **Accuracy** was selected as the primary evaluation metric during training to monitor how well the model correctly classifies customers. Additional metrics such as **Precision, Recall,** and **F1 Score** were later computed to provide a comprehensive assessment of the model's performance, especially in handling imbalanced classes often present in churn datasets.

📝 6. Training the Model

Training Process 🏋️‍♂️

- **Training the ANN Model**:
    - The **Artificial Neural Network (ANN)** model was trained using the preprocessed training dataset to predict customer churn. Training involves adjusting the weights of the model to minimize the error between predicted and actual outcomes.
    - **Number of Epochs**: The model was trained over **50 epochs**, which was determined to be sufficient for the model to learn from the data without excessive training that could lead to overfitting.
    - **Batch Size**: A **batch size of 32** was used, meaning the model processed 32 samples at a time before updating its weights. This size was chosen to provide a balance between the efficiency of training and the stability of the model's learning process.

Training Results 📊

The training of the Artificial Neural Network (ANN) model was conducted over 50 epochs with early stopping to prevent overfitting. During training, the model demonstrated a steady improvement in both training and validation accuracy.

- **Training Accuracy**: The model achieved a training accuracy of approximately **79%** by the 20th epoch. The accuracy continued to improve during early epochs, but early stopping halted the training after no significant improvement was observed beyond this point.

- **Training Loss**: The training loss consistently decreased during the training process. The final training loss stabilized at around **0.44**, demonstrating that the model was able to effectively minimize the error between predictions and actual outcomes.

- **Validation Results**: Throughout the training, the model's validation accuracy also improved but at a slower rate compared to the training accuracy. The **validation loss** showed some fluctuations but generally decreased. The final validation accuracy was **78%**, and the validation loss was **0.45**. These figures indicate that the model was able to generalize reasonably well without significant overfitting.

- **Early Stopping**: The model employed early stopping with a patience of 10 epochs, halting the training at epoch 20. This approach ensured that the model did not continue training beyond the point where it stopped improving on validation data.

Graphs for the training and validation loss, as well as accuracy over epochs, are included in the appendices (as shown in **training_results.txt** and **training_visualization.png**). These visualizations clearly depict the trend of model convergence over time.

Overfitting Prevention 🚫

- **Strategies to Prevent Overfitting**:

    - **Dropout Layers**: As part of the model architecture, **dropout layers** with a rate of **0.2** were implemented after each hidden layer. Dropout randomly deactivates a fraction of the neurons during training, which helps prevent the model from becoming too reliant on any one neuron. This encourages the network to learn more robust features that generalize better to new data.

    - **Early Stopping**: An **early stopping** mechanism was employed to halt training if the model's performance on the validation set did not improve for several consecutive epochs. This strategy helps prevent overfitting by stopping the training process when the model starts to learn noise from the training data instead of useful patterns. It ensures that the final model is the one that performed best on the validation set.

📝 7. Model Evaluation

Performance Metrics 📏

- **Metrics Used to Evaluate Model Performance**:
  - **Accuracy**: This metric represents the proportion of correct predictions made by the model out of all predictions. It provides a general sense of how well the model is performing but can be misleading if the dataset is imbalanced.
  - **Precision**: Precision measures the accuracy of positive predictions. It is calculated as the number of true positives divided by the sum of true positives and false positives. A high precision indicates that the model is effective in minimizing false positives.
  - **Recall**: Also known as sensitivity, recall measures the ability of the model to correctly identify all positive instances. It is calculated as the number of true positives divided by the sum of true positives and false negatives. High recall ensures that most actual positives are identified, minimizing false negatives.
  - **F1-Score**: The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is particularly useful when dealing with imbalanced datasets where both precision and recall are important.
  - **ROC-AUC (Receiver Operating Characteristic - Area Under Curve)**: The ROC-AUC score evaluates the model's ability to distinguish between classes. A higher AUC indicates better performance across various threshold settings.
  - **Confusion Matrix**: This is a summary of prediction results on a classification problem. It shows the number of true positives, false positives, true negatives, and false negatives, providing a detailed breakdown of model performance.

Evaluation Results 📈

- **Results on the Test Dataset**:
  - **Confusion Matrix**:
    - The confusion matrix revealed how well the model distinguished between churn and non-churn customers:
      - **True Positives (TP)**: 121 Correctly predicted churn customers.

- **True Negatives (TN)**: 978 Correctly predicted non-churn customers.

- **False Positives (FP)**: 57 Non-churn customers incorrectly predicted as churn.

- **False Negatives (FN)**: 253 Churn customers incorrectly predicted as non-churn.

- **ROC Curve**:

  - The ROC curve visualizes the trade-off between true positive rate (recall) and false positive rate. The **AUC** was observed to be **0.79**, indicating good model discrimination between churn and non-churn customers.

- **Precision-Recall Curve**:

  - This curve illustrates the trade-off between precision and recall for different thresholds. It is especially useful when dealing with class imbalance, providing a more comprehensive view of the model's performance across different decision thresholds.

Analysis of Results 🔍

- **Insights Gained from Evaluation Metrics**:

  - **Model Strengths**:

    - The **high accuracy** and **F1-score** suggest that the model performs well in both identifying churn and non-churn customers. It indicates that the model can handle both classes effectively, which is crucial for business decisions.

    - The **ROC-AUC score** of **0.79** demonstrates that the model has a strong ability to differentiate between churn and non-churn customers. This is particularly valuable for targeted retention strategies, as it ensures that most at-risk customers are correctly identified.

  - **Areas for Improvement**:

    - The model exhibited some **false positives** and **false negatives**, indicating areas where predictions could be refined. False positives might lead to unnecessary retention efforts, while false negatives could result in lost customers.

- Further tuning of the decision threshold based on business priorities (such as maximizing recall to reduce churn or precision to reduce costs) could enhance the model's alignment with business objectives.

📝 8. Prediction and Analysis

Prediction on Test Data 📊

- **Process of Making Predictions on the Test Dataset**:
  - The model, after being trained and validated, was used to make predictions on the test dataset. This dataset consists of data that the model has never seen before, ensuring an unbiased evaluation of its predictive capabilities.
  - Predictions were made using the **trained ANN model** to assess whether each customer would churn or not. The model outputs probabilities for each customer belonging to the churn or non-churn class.
  - A **threshold of 0.5** was applied to these probabilities to convert them into binary predictions (churn or no churn). This means that if the model predicts a probability higher than 0.5 for a customer to churn, it is classified as a churn customer.

Critical Attributes Influencing Predictions 🔍

- **Analysis of Features Impacting Model Predictions**:
  - During the training phase, the model learned patterns and relationships between the features and the target variable (churn). Some features have a more significant influence on the model's predictions than others.
  - The most critical attributes influencing the predictions include:
    - **Tenure**: Customers with shorter tenure were more likely to churn, suggesting dissatisfaction or unmet expectations in the early phases of the service.
    - **Monthly Charges**: Higher monthly charges correlated with higher churn rates, possibly indicating price sensitivity among customers.

- **Contract Type**: Customers with month-to-month contracts were more prone to churn than those with longer-term contracts, as they have more flexibility to leave without penalties.

- **Support Calls**: An increase in the number of support calls was associated with higher churn, possibly indicating unresolved issues or dissatisfaction with customer service.

- These attributes are significant as they directly impact customer satisfaction and retention. Understanding these factors allows businesses to develop targeted strategies for reducing churn.

Visualizations 📉

- **Importance of Visual Representations**:

  - Visualizations provide an intuitive understanding of which features are most influential in the model's decision-making process.

  - **Feature Importance Graphs**:

    - These graphs display the relative importance of each feature used by the model to make predictions. Features like **Tenure** and **Monthly Charges** were among the top predictors, aligning with our analysis.

  - **SHAP Values (SHapley Additive exPlanations)**:

    - SHAP values offer a detailed view of how each feature contributes to the model's prediction for a specific instance. They help in understanding the impact of each feature on the model's output, providing both global and local interpretability.

    - For example, **higher SHAP values** for the feature **Contract Type** suggest that month-to-month contracts significantly increase the likelihood of churn.

  - **Confusion Matrix**:

    - A confusion matrix was generated to visualize the performance of the model. It provides a detailed breakdown of the true positives, false positives, true negatives, and false negatives, offering insights into where the model excels and where it might need improvement.

📝 9. Conclusion

Summary of Findings 📊

- The project aimed to develop and evaluate an **Artificial Neural Network (ANN) model** for predicting customer churn in the telecommunications industry. The model was trained on a dataset comprising various customer attributes, including tenure, monthly charges, and contract type, among others.
- **Effectiveness of the ANN Model**:
    - The ANN model demonstrated robust performance in predicting customer churn, achieving a **test accuracy of 78%**. Key metrics such as **precision (68%)**, **recall (32%)**, and **F1-score (44%)** indicate that the model is fairly balanced in identifying both churn and non-churn customers.
    - The model effectively captured patterns within the data, utilizing critical features like **tenure**, **monthly charges**, and **contract type** to make accurate predictions.

Business Implications 💼

The results of the model indicate a reasonably strong predictive ability, especially with an accuracy of **79%** and a precision score of **68%**. However, the relatively low recall of **32%** presents some challenges for business decisions, particularly for customer retention strategies.

· **Impact of Precision**: The model's precision score suggests that when it predicts a customer will churn, it is correct about 68% of the time. This can be valuable in prioritizing targeted interventions or retention campaigns for customers predicted to churn. Precision is essential in this context to minimize the risk of misidentifying non-churn customers as churners, thus avoiding unnecessary retention efforts and costs.

· **Recall and its Limitations**: With a recall score of **32%**, the model is identifying only about a third of the actual churners. This means that a significant number of customers who are likely to churn may not be flagged by the model, potentially resulting in missed opportunities to retain these customers. From a business perspective, improving recall is crucial to effectively addressing customer churn and reducing overall churn rates.

· **AUC and Overall Prediction Power**: The model's **AUC score of 0.79** demonstrates that it can distinguish between churned and non-churned customers with reasonable accuracy. However, a focused effort on improving recall could lead to more effective retention efforts and a more balanced model overall.

· **Actionable Insights**: While the model performs well in terms of precision, a key business implication is the need for more targeted strategies to address customers who are at risk of being misclassified as non-churners. Additional data or adjustments to the classification threshold could help capture more of these at-risk customers.

Future Work 🔮

Based on the current performance of the model, several areas of improvement and further exploration are recommended:

1. **Improving Recall**: The model's recall score of 32% suggests that a large portion of churned customers are not being accurately predicted. Future iterations should focus on improving this metric to ensure a more comprehensive identification of churned customers. Approaches to consider include:
    - **Adjusting the Classification Threshold**: Fine-tuning the probability threshold for classifying customers as churned or non-churned may help balance precision and recall based on business priorities.
    - **Class Imbalance Handling**: Using techniques such as **SMOTE (Synthetic Minority Over-sampling Technique)** or **undersampling** the majority class could improve the model's sensitivity to churn cases.
2. **Feature Engineering**: Further exploration into feature selection and engineering could uncover additional patterns in the data. For instance, creating interaction terms between customer tenure, monthly charges, and contract types might reveal more complex relationships that are not currently being captured.
3. **Hyperparameter Tuning**: While the current model performs well, experimenting with additional hyperparameter tuning techniques such as **grid search** or **random search** could improve the model's performance. Parameters like the number of hidden layers, learning rate, and dropout rates could be adjusted for better optimization.

4. **Exploring Alternative Models**: The current model uses a standard feedforward ANN. However, other models such as **Gradient Boosting**, **Random Forests**, or **XGBoost** could be explored to compare performance, particularly in improving recall without sacrificing precision.

5. **Explainability and Interpretability**: As the model will likely influence customer retention strategies, improving the explainability of the predictions could provide more actionable insights to the business. Using techniques like **SHAP (SHapley Additive exPlanations)** or **LIME (Local Interpretable Model-agnostic Explanations)** can help identify which features are most important in predicting churn.

📚 10. Appendix

Code Snippets 🖥️

Below are the key code snippets used throughout the project for building, training, evaluating, and analyzing the Artificial Neural Network (ANN) model:

1. **Data Preprocessing**:

```
# Importing necessary libraries

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler


# Loading the dataset

data = pd.read_csv('data/raw/Dataset (ATS)-1.csv')  # Corrected file name based on your dataset


# Data cleaning and feature engineering

# Handling missing values and converting data types as needed

data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
```

```
data = data.fillna(0)


# Splitting the data into features and target

X = data.drop(['Churn_Yes', 'Churn_No'], axis=1)  # Using correct target columns

y = data['Churn_Yes']  # Assuming 'Churn_Yes' as the binary target


# Splitting the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Scaling the features

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)
```

2. **ANN Model Architecture**:

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout


# Defining the ANN model architecture

model = Sequential()

model.add(Dense(units=64, activation='relu', input_shape=(X_train_scaled.shape[1],)))

model.add(Dropout(0.2))

model.add(Dense(units=32, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(units=1, activation='sigmoid'))  # Output layer with sigmoid activation for
    binary classification
```

Compiling the model

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

3. **Training the Model**:

Training the model

history = model.fit(X_train_scaled, y_train, epochs=50, batch_size=32, validation_split=0.2, verbose=1)

Saving the model architecture and weights

model.save('Predictive_Modeling/trained_model/best_model.h5')

4. **Model Evaluation**:

from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc, precision_recall_curve

Making predictions on the test set

y_pred_prob = model.predict(X_test_scaled)

y_pred = (y_pred_prob > 0.5).astype(int)

Generating evaluation metrics

print(classification_report(y_test, y_pred))

conf_matrix = confusion_matrix(y_test, y_pred)

fpr, tpr, _ = roc_curve(y_test, y_pred_prob)

roc_auc = auc(fpr, tpr)

Plotting confusion matrix, ROC curve, and precision-recall curve

Saving these visualizations to the evaluation folder in ‘Predictive_Modeling/visualization'

Additional Figures and Tables 📊

1. **Confusion Matrix**:
   - A confusion matrix has been generated to visualize the model’s performance on the test dataset, providing a clear view of true positives, true negatives, false positives, and false negatives.
2. **ROC Curve**:
   - The ROC curve illustrates the trade-off between sensitivity and specificity for various threshold settings, helping to evaluate the classifier’s performance.
3. **Precision-Recall Curve**:
   - This curve provides insights into the model’s performance on imbalanced data, indicating how well it distinguishes between classes.

📖 References

1. **Advanced Consultant Service Internal Data (2024).** Customer dataset used for training and testing predictive models. Provided by the project’s stakeholders at Advanced Consultant Service.

2. **Chaudhary, B. (2024).** data_preprocessing.py: Code for preprocessing data, handling missing values, and encoding categorical variables. Internal project script by Bhavesh Chaudhary, Data Engineer.

3. **Chaudhary, B. (2024).** feature_engineering.py: Code for creating new features to enhance model performance. Internal project script by Bhavesh Chaudhary, Data Engineer.

4. **Basyal, K. (2024).** clustering_analysis.py: Code for performing clustering analysis as part of the customer segmentation efforts. Internal project script by Kushal Basyal, Data Analyst for Clustering Analysis.

5. **Ali, M. (2024).** predictive_modeling.py: Code for developing and training an Artificial Neural Network (ANN) model for predicting customer churn. Internal project script by Mujtaba Ali, Data Analyst for Predictive Modeling.

6. **Ali, M. (2024).** predict_churn.py: Code for predicting customer churn using the trained ANN model. Internal project script by Mujtaba Ali, Data Analyst for Predictive Modeling.

7. **Ali, M. (2024).** test_model.py: Code for testing the performance of the ANN model on the test dataset. Internal project script by Mujtaba Ali, Data Analyst for Predictive Modeling.

8. **Ali, M. (2024).** EvaluateModelPerformance.py: Code for evaluating the ANN model's performance and analyzing predictions. Internal project script by Mujtaba Ali, Data Analyst for Predictive Modeling.

9. **Python Software Foundation (2023).** Python Language Reference, version 3.8. Available at https://www.python.org/.

10. **Abadi, M., et al. (2016).** TensorFlow: Large-scale machine learning on heterogeneous systems. Available at https://www.tensorflow.org/.

11. **Pedregosa, F., et al. (2011).** Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830. Available at https://scikit-learn.org/.

12. **Pandas Development Team (2023).** Pandas: Python Data Analysis Library. Available at https://pandas.pydata.org/.

13. **Waskom, M. L. (2021).** Seaborn: Statistical Data Visualization. Journal of Open Source Software, 6(60), 3021. Available at https://seaborn.pydata.org/.

14. **Oliphant, T. E. (2006).** A guide to NumPy. Trelgol Publishing. Available at https://numpy.org/.

15. **Brownlee, J. (2020).** Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.

16. **Chollet, F. (2018).** Deep Learning with Python. Manning Publications.

17. **Kingma, D. P., & Ba, J. (2015).** Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR). Available at https://arxiv.org/abs/1412.6980.

18. **Han, J., Kamber, M., & Pei, J. (2012).** Data Mining: Concepts and Techniques. Elsevier.

19. **Raschka, S., & Mirjalili, V. (2019).** Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing.