Project Outline for IE500 Data Mining

Project Name: Customer Churn Prediction with Artificial Neural Network

Team number: Team 9

Team member: Zhiqi Yang, 2110635

Elise Wolf, 1828204

Xi Liu, 2113820

Shiqi Zhou, 2036915

Yenan Chen, 2113612

1 What is the problem you are solving?

1.1 Overview/Abstract

Customer churn prediction is crucial for businesses aiming to retain clients and reduce revenue loss. This project utilizes an Artificial Neural Network (ANN) to predict customer churn, a critical issue in industries like telecommunications. Using customer data, including demographics and service usage, the ANN model is trained to detect patterns that indicate churn. After data preprocessing, the model's performance is evaluated using accuracy, precision, and AUC-ROC. The results provide insights into key churn drivers, helping businesses improve retention strategies.

Figure 1. The general workflow of customer churn prediction based on double-compressed artificial neural network.

1.2 Problem Statement

The primary objective of this project is to develop a machine learning model that can predict whether a customer of a telecommunications company is likely to churn (i.e., leave the service). Customer churn is a critical issue for telecom companies, as acquiring new customers is often more expensive than retaining existing ones. By predicting which customers are likely to churn, telecom companies can take proactive measures to retain these customers and reduce overall churn rates.

We aim to address this problem from two perspectives:

- **Customer Retention for Telco Companies**: The company can use this information to focus their customer retention strategies on high-risk customers. This could involve offering special discounts, improved services, or personalized offers to customers who are likely to leave.
- **Business Decision-Making**: Identifying the factors that contribute most to customer churn (e.g., contract type, tenure, services used) will allow telecom companies to optimize their business strategies and product offerings based on the needs of their customers.

Key questions we aim to answer include:

- Which features (such as customer tenure, contract type, and monthly charges) are most strongly associated with customer churn?

- Can we accurately predict customer churn using machine learning models, and how effective are different models in this task?
- What actions can telecom companies take based on the predictions to reduce churn rates?

2 What data will you use?

We will use the **Telco Customer Churn** dataset, available on Kaggle, which contains data from a telecommunications company on customer demographics, account information, services used, and whether the customer has churned.

The dataset includes the following features:

- **Customer Information**: Gender, senior citizen status, partner and dependent status.
- **Account Information**: Contract type, tenure, payment method, paperless billing, and monthly charges.
- **Service Information**: Internet service, online security, online backup, device protection, technical support, and TV/movie streaming services.
- **Target Variable**: `Churn` – a binary variable indicating whether the customer has churned.

Data source:

- Telco Customer Churn Dataset: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Data Gathering:

- The dataset is readily available for download in **CSV format** from Kaggle. After downloading, we will inspect and preprocess the data for analysis. The dataset is already well-structured, but we will perform necessary preprocessing steps, such as handling missing values and encoding categorical variables.

3 How will you solve the problem?

3.1 Methods: Problem Solving Steps

We will approach the problem by building and evaluating multiple machine learning models that predict customer churn based on the available features. Our approach will consist of the following steps:

3.1. Preprocessing Steps Required

- Data Cleaning: We will first check for and handle any missing or inconsistent data entries. Missing values in numerical fields will be imputed using the median, while categorical fields may have missing values filled with the mode.

- Feature Encoding: Since the dataset contains categorical variables (e.g., gender, contract type), we will use techniques such as one-hot encoding to transform these into numerical representations.
- Scaling: Numerical features such as MonthlyCharges and TotalCharges will be scaled to ensure that all features contribute equally to the model.

3.2. Algorithms and model specifics

We will perform an exploratory data analysis (EDA) to identify which features are most strongly associated with customer churn. This will help us understand the relationships within the data and potentially reduce the number of features for model training. Techniques like correlation analysis and recursive feature elimination (RFE) will be used to select the most relevant features.

3.2.1. Naive Bayes

- **Difficulty**: Easiest
- **Explanation**: Naive Bayes is a very simple and fast model to implement. It works on the assumption that features are independent, which makes it computationally efficient and easy to train. It's often used as a baseline model for classification tasks.
- **Application**: Naive Bayes can be used for customer churn prediction by treating each feature (e.g., customer behavior, demographics, usage statistics) as independent. The model calculates the likelihood of a customer churning based on these features.
- **Pros**: Easy to implement and fast to train.
- **Cons**: Assumes feature independence, which might not hold in real-world churn prediction problems.

3.2.2. Logistic Regression

- **Difficulty**: Easy
- **Explanation**: Logistic regression is simple to understand and implement, especially for binary classification tasks. It assumes a linear relationship between the features and the log-odds of the target variable. It's computationally efficient and interpretable, making it a good starting point for many classification problems. As the target variable is binary, logistic regression is a natural choice for baseline classification. This interpretable model will give us insights into which features have the strongest impact on churn.
- **Application**: Logistic Regression is often used in customer churn prediction for binary classification (churn or no churn). It models the probability of a customer churning based on input features like the number of complaints, subscription length, and usage frequency.
- **Pros**: Simple, interpretable, and effective for smaller datasets. The coefficients provide insights into the importance of different factors.
- **Cons**: Assumes a linear relationship between the features and the churn probability, which might not capture complex patterns.

3.2.3. K-Nearest Neighbors (KNN)

- **Difficulty**: Easy
- **Explanation**: KNN is easy to understand and implement because it does not require a training process. Predictions are made based on the majority vote of the nearest neighbors. However, KNN can become computationally expensive during prediction for large datasets since it stores all the data.
- **Application**: KNN can predict churn by finding the most similar customers (neighbors) and using their churn behavior to predict whether a current customer will churn. If a majority of similar customers have churned, the model predicts that the customer is likely to churn.
- **Pros**: Intuitive and easy to understand. No training required.
- **Cons**: Computationally expensive for large datasets, especially at prediction time. Doesn’t perform well with noisy data.

3.2.4. Decision Trees

- **Difficulty**: Moderate
- **Explanation**: Decision trees are easy to visualize and interpret. They split the dataset based on feature values and recursively partition the data. The algorithm is simple, but it can overfit, so tree pruning and hyperparameter tuning might be needed for better performance. Decision trees will help us identify feature interactions and provide a model that is easy to visualize and interpret. This method is also effective for handling non-linear relationships between features.
- **Application**: Decision trees can be used to predict customer churn by splitting data into branches based on decision rules (e.g., if a customer’s usage is below a certain threshold, predict churn). The tree is built by recursively selecting the feature that best separates churners from non-churners.
- **Pros**: Easy to visualize and interpret. Works well with categorical and numerical data.
- **Cons**: Can easily overfit, which might require pruning or other methods to prevent.

3.2.5. Random Forest

- **Difficulty**: Moderate
- **Explanation**: Random forests are an ensemble method that builds multiple decision trees and averages their predictions. While it’s more complex than a single decision tree, it’s still relatively easy to implement using libraries like scikit-learn. It provides better accuracy and reduces overfitting compared to decision trees. This model tends to generalize well and can handle large feature sets.
- **Application**: Random Forest uses an ensemble of decision trees to improve churn prediction accuracy. Each tree is built from a random sample of data and a random subset of features. The final prediction is made by averaging the predictions of all the trees.
- **Pros**: Reduces overfitting and improves generalization compared to a single decision tree. Can handle large datasets and complex relationships.
- **Cons**: Less interpretable than a single decision tree. More computationally intensive.

3.2.6. Support Vector Machines (SVM)

- **Difficulty**: Moderate to Hard
- **Explanation**: SVMs work well for linear and non-linear classification tasks. The math behind SVMs is more complex, especially when using different kernel functions to handle non-linear data. Tuning parameters like the kernel and the regularization term can be challenging. SVM is useful for creating a clear decision boundary between customers who churn and those who don't. We will experiment with both linear and non-linear kernels.
- **Application**: SVM can be applied to customer churn prediction by finding the best hyperplane that separates churners from non-churners. SVM can also be used with kernels to capture non-linear relationships in the data.
- **Pros**: Effective for high-dimensional spaces and when there's a clear margin of separation between classes.
- **Cons**: Computationally intensive, especially with large datasets. Requires careful tuning of kernel and regularization parameters.

3.2.7. XGBoost

- **Difficulty**: Hard
- **Explanation**: XGBoost is an advanced implementation of gradient boosting. It offers excellent performance, especially for large datasets, but requires careful tuning of hyperparameters. The model works by iteratively building decision trees that correct the errors of previous trees. It's more difficult to understand and optimize due to the number of parameters involved. By training weak learners sequentially, XGBoost can improve accuracy and potentially outperform simpler models.
- **Application**: XGBoost is a powerful model for customer churn prediction. It uses gradient boosting over decision trees, iteratively building models that correct the errors of previous models. XGBoost excels in handling large datasets and capturing complex relationships between features.
- **Pros**: High accuracy, fast training, and can handle missing data and complex interactions between variables.
- **Cons**: Requires careful tuning of hyperparameters and may not be as interpretable as simpler models.

3.2.8. Artificial Neural Networks (ANN)

- **Difficulty**: Hard
- **Explanation**: ANNs are biologically-inspired models that are flexible and capable of learning complex patterns in data. However, they require more data and computational resources than traditional models. The complexity increases with the number of layers and neurons, making them harder to tune and interpret.
- **Application**: ANNs can model complex relationships between input features (e.g., demographics, usage data, behavior patterns) and the likelihood of churn. It can capture non-linear relationships that other models might miss.
- **Pros**: Capable of handling complex, non-linear relationships in data. Can be applied to both structured and unstructured data (e.g., combining customer transaction history with text data from customer reviews).

- **Cons**: Requires a large amount of data and computational resources. Longer training time and hyperparameter tuning can be challenging.

3.2.9. Multilayer Perceptron (MLP)

- **Difficulty**: Hardest
- **Explanation**: MLP is a type of deep neural network with multiple layers of neurons (hence "multilayer"). While it can model highly complex data, it’s difficult to train and tune. It requires large amounts of data, and the choice of hyperparameters like the number of layers, learning rate, and regularization terms significantly affects the model’s performance.
- **Application**: MLP is a type of ANN with multiple layers of neurons, which can be used to predict customer churn by learning from complex, high-dimensional data. The MLP architecture is well-suited for capturing non-linear patterns in the data.
- **Pros**: Flexible and powerful for modeling complex relationships. Can handle interactions between many features.
- **Cons**: Requires careful tuning, a large amount of data, and significant computational power. More difficult to interpret compared to simpler models.

4 How will you measure success? (Evaluation method)

4.1 Measure Success: Key Classification Evaluation Metrics

We will evaluate our models using standard classification metrics:

- Accuracy: The proportion of correctly classified customers.
- Precision and Recall: Precision measures how many of the predicted churn customers actually churn, while recall measures how many of the actual churn customers were correctly identified.
- F1-Score: A balanced measure of precision and recall, especially useful when the dataset is imbalanced.
- ROC-AUC Score: The area under the ROC curve will provide insight into how well our model distinguishes between churn and non-churn customers across various threshold values.

The success of the project will be measured through multiple approaches:

1. **Train-Test Split**: The dataset will be split into 80% training and 20% testing sets. The models will be trained on the training data, and performance will be evaluated on the test set.
2. **Cross-Validation**: We will apply k-fold cross-validation (likely using 5 or 10 folds) to ensure that our models generalize well to unseen data and avoid overfitting.

3. **Imbalance Handling**: Given that customer churn is a relatively rare event, we expect the dataset to be imbalanced. Therefore, we may employ techniques such as **SMOTE (Synthetic Minority Over-sampling Technique)** to balance the training set and improve model performance.
4. **Model Interpretability**: In addition to accuracy and performance metrics, we will prioritize models that provide actionable insights into why customers churn. For example, decision trees and logistic regression can be used to highlight the key factors that drive customer behavior.

5 What do you expect your results to look like?

5.1 Provide Key Insights into Churn Drivers

- **A predictive model:** The output will include a classification model that predicts the likelihood of churn for each customer based on their features.
- **Feature Importance Analysis:** Using models like Random Forest or XGBoost, we will be able to rank the features that most influence customer churn. This will give the telecom company actionable insights into which factors they should address to reduce churn.
- **Churn Probability Dashboard:** We may create a simple interface or dashboard that visualizes the probability of churn for individual customers, along with the top contributing factors.
- **Actionable Recommendations:** Based on our findings, we will provide recommendations to the telecom company about which customer segments are most at risk of churn and what strategies they might employ to retain these customers.

5.2 Analysis: Build a High-performing Predictive Model

One of the outputs of our project will be the development of a **Churn Probability Dashboard**. This dashboard will be designed to support **customer service representatives** (CSRs) during interactions with customers, specifically focusing on identifying at-risk customers and providing personalized retention strategies. Below is an outline of the functionality, structure, and impact of this dashboard.

The **Churn Probability Dashboard** will display real-time predictions of whether a customer is likely to churn, based on their personal data, account history, and interactions with the telecom services. In addition to the churn probability, the dashboard will offer **actionable insights** and **recommendations** that the CSR can use to retain the customer.

Key Features of the Dashboard:

1. **Customer Information Panel**: Displays key customer details such as:
    - Customer ID

        - Tenure (how long they have been with the company)
        - Monthly charges and contract type
        - Services subscribed to (e.g., internet, streaming, phone services)
2. **Churn Probability Score**: The model's prediction of the **probability** that the customer will churn, expressed as a percentage (e.g., "Churn Probability: 75%"). The churn score will be **color-coded** to reflect the urgency:
        - Green (0-30%): Low risk of churn.
        - Yellow (30-60%): Medium risk of churn.
        - Red (60-100%): High risk of churn.
3. **Top Contributing Factors**: A list of the **most influential factors** contributing to the churn prediction for the specific customer. This feature gives the CSR context for why the customer may be dissatisfied or considering leaving, allowing for more **targeted conversation points**. For example, the dashboard might show:
        - "High monthly charges compared to similar customers."
        - "Short tenure—customer has been with the company for less than 6 months."
        - "Lack of bundled services—customer only has one service subscription."
4. **Retention Recommendations**: Based on the churn prediction and contributing factors, the dashboard will provide **customized retention strategies**. Examples include:
        - "Offer a discount on their monthly charges to match competitors' pricing."
        - "Upgrade the customer to a premium internet service at no additional cost for the first 3 months."
        - "Recommend a contract change to a more flexible option (e.g., moving from annual to monthly)."
5. **Customer Sentiment History**: If available, the dashboard will display past **customer sentiment** based on previous interactions with customer service (e.g., whether previous calls have resulted in complaints or neutral/positive feedback).