## A1:PROPOSAL OF QUESTION

The research question I will be investigating is: *What factors contribute most to customer tenure?* Understanding the key drivers behind customer tenure can provide valuable insights to a company, helping it develop strategies to retain customers for the long term. The benefit of this analysis is that by identifying the primary factors that influence tenure, a company can design tailored incentives or introduce product enhancements that encourage long-term customer relationships. Reducing customer churn and extending tenure lead directly to cost savings, as the organization can focus less on expensive customer acquisition strategies. For this project, I will answer the question by applying decision tree analysis. This method will allow me to assess the hierarchy and interaction of various customer attributes—such as monthly charges, service usage, and product preferences—and pinpoint which factors play the most significant roles in customer retention. The goal is to uncover actionable insights for optimizing retention strategies and maximizing the long-term value of existing customers, potentially enhancing overall business performance.

## A2:DEFINED GOAL

The goal of this data analysis is to identify the most influential factors that impact customer tenure using the DecisionTreeRegressor from the sklearn library. By focusing on variables that were used in previous models, we aim to uncover the key characteristics that contribute to how long customers remain with the company. Our primary objective is to predict potential churning points by analyzing these factors, which will allow for more proactive and targeted customer service interventions. Through predictive modeling, we can better anticipate customer behavior, enabling the company to customize services, offers, or incentives that encourage long-term retention. Identifying these important factors also helps improve resource allocation and strategic planning, allowing the company to focus its efforts on the highest impact areas, ultimately bolstering customer satisfaction, retention, and solidifying our competitive advantage in the industry.

## B1:EXPLANATION OF PREDICTION

For this task, I chose to use decision trees as the primary method for predicting customer tenure. The decision tree algorithm offers several key advantages, making it suitable for this analysis. Firstly, it is highly interpretable and allows for easy visualization of the results. This means that the decision-making process behind each prediction is transparent and can be easily explained to non-technical stakeholders. Decision trees can handle both numerical and categorical data, which is beneficial given the mixed data types present in this dataset, such as monthly charges (numerical) and service preferences (categorical). The core strength of decision trees lies in their ability to segment the data based on the most significant variables, creating a flow of "if-then" conditions that lead to a prediction. This structure makes it easy to see which variables (e.g., service usage or monthly fees) have the most substantial influence on

customer tenure. Each branch of the tree represents a decision based on a specific feature, and the paths lead to different outcomes or predictions about tenure. The ability to visualize the final model helps in identifying clear patterns and trends within the data.

One reason I chose decision trees over more complex methods, like random forests, is the simplicity of interpretation. A single decision tree can be fully visualized, providing a clear understanding of how different customer characteristics impact tenure. This transparency is critical when communicating findings to stakeholders who may prioritize actionable insights over model complexity. However, decision trees are prone to overfitting, especially when applied to small datasets, as they tend to capture noise and anomalies in the data. This may lead to less accurate predictions. While random forests—an ensemble method combining multiple decision trees—can mitigate overfitting and often provide more accurate results, I chose decision trees because the goal is to identify and clearly explain the most important factors influencing tenure. In cases where prediction accuracy is prioritized over interpretability, a random forest would indeed be more suitable, but in this scenario, the decision tree's ease of use and visual clarity outweigh its limitations.

The expected outcomes of this analysis will include a clear understanding of which customer attributes have the most significant impact on tenure, enabling the company to make data-driven decisions to enhance customer retention. The decision tree will highlight potential points of customer churn, helping in designing strategies to keep customers engaged for a longer duration.

B2:SUMMARY OF METHOD ASSUMPTION

One key assumption of decision tree models is the assumption of feature independence. This means that the model treats each feature individually when making splits at each node, assuming that there is no inherent relationship or correlation between the features. In other words, it assumes that the predictive power of one feature is not influenced by the presence or value of another feature. However, in real-world datasets, this assumption often does not hold true. Features may exhibit high correlation with one another, yet decision trees can still perform effectively even in the presence of these correlations. For example, customer-related variables like monthly charges and bandwidth usage might be correlated, but the decision tree will still analyze them separately when determining how to split the data. Despite this, the model can still achieve good predictive accuracy, as decision trees inherently handle complex, non-linear relationships between variables. While decision trees don’t explicitly account for feature interactions during the split process, their hierarchical structure can still capture complex interactions implicitly, which is why they remain robust even when the assumption of independence is violated. This flexibility makes decision trees a versatile tool for real-world predictive modeling tasks, even in cases where feature interdependence is present.

B3:PACKAGES OR LIBRARIES LIST

The following Python libraries have been chosen for this analysis, each serving a specific purpose to support the modeling and evaluation process:

1. **Pandas (`pd`)**: This library is essential for data manipulation and analysis. It allows for efficient handling of structured data, such as importing the dataset, exploring its contents, and performing operations like filtering, aggregating, and transforming the data. In this analysis, Pandas will help clean and prepare the dataset for modeling.
2. **NumPy (`np`)**: NumPy is a fundamental package for numerical computing in Python. It provides support for working with arrays and offers a variety of mathematical functions. This library is used to handle numerical operations on the dataset and is particularly useful for manipulating data into arrays and performing mathematical computations, which are essential for model training.
3. **Seaborn (`sns`)**: Seaborn is a powerful data visualization library built on top of Matplotlib. It simplifies the creation of informative and aesthetically pleasing visualizations. Seaborn will be used to create exploratory data visualizations such as correlation heatmaps or distribution plots, helping to better understand the relationships between variables and the structure of the data.
4. **Matplotlib (`plt`)**: Matplotlib is a versatile plotting library used for creating static, animated, and interactive visualizations. In this analysis, Matplotlib will be used alongside Seaborn to plot decision trees, visualize model performance, and produce customized charts to better explain the outcomes of the analysis.
5. **Scikit-learn (`train_test_split`, `GridSearchCV`, `r2_score`, `accuracy_score`, `mse`, `DecisionTreeRegressor`, `tree`)**: Scikit-learn is the core machine learning library in Python, providing tools for building and evaluating models.
    - `train_test_split`: This function helps in splitting the data into training and testing sets to validate the model's performance.
    - `GridSearchCV`: This tool is used to fine-tune the model's hyperparameters, ensuring that the decision tree is optimized for better performance.
    - `r2_score`, `accuracy_score`, `mse`: These metrics are used to evaluate the performance of the model. They help assess the goodness of fit, accuracy, and error in predicting customer tenure.
    - `DecisionTreeRegressor`: This is the main algorithm used to create the regression model, predicting the factors contributing to customer tenure.
    - `tree`: The `tree` module is used to visualize the structure of the decision tree, enabling easy interpretation of the model's decision-making process.

C1:DATA PREPROCESSING

One key data preprocessing goal relevant to the decision tree prediction method is the conversion of categorical variables into numerical format. In this case, the `OnlineSecurity` feature, which initially contains categorical values ('Yes' and 'No'), was converted into numerical

format by mapping 'Yes' to 1 and 'No' to 0. This step is essential because decision trees in Scikit-learn require numerical input for processing and making predictions. Converting categorical values into a binary format enables the decision tree to correctly evaluate the impact of features like `OnlineSecurity` on the target variable, `Tenure`. Additionally, renaming the columns to follow Python naming conventions (`snake_case`) ensures clarity and consistency throughout the code. This preprocessing ensures the dataset is in a usable format for the decision tree model, which relies on numerical data to evaluate the relationships between features and the target variable effectively.

C2:DATA SET VARIABLES

The initial dataset variables used to perform the analysis for predicting customer tenure, with the dependent variable included, are as follows:

## Numeric Variables (Independent Variables):

1. Children: Represents the number of children the customer has.
2. Age: The age of the customer.
3. MonthlyCharge: The monthly charge the customer pays.
4. Bandwidth_GB_Year: The total bandwidth used by the customer in a year.
5. OnlineSecurity: converted to a binary numeric format where:
    - 'Yes' = 1
    - 'No' = 0

## Dependent Variable (Target):

1. Tenure: Represents the length of time (in months) the customer has been with the company. This is the variable we are trying to predict using the other features.

The combination of these independent variables was used to model the Tenure of customers and identify key factors affecting it.

C3:STEPS FOR ANALYSIS

The data preparation for the analysis involved several key steps, each implemented with a corresponding code segment. Below are the steps used, along with the associated code:

## 1. Load the Dataset:

The dataset is read into a pandas DataFrame, specifying only the relevant columns required for the analysis.

**Code Segment**:

```
df = pd.read_csv('/content/drive/My Drive/D209/churn_clean.csv',
usecols=['Children', 'Age', 'OnlineSecurity', 'Tenure',
'MonthlyCharge', 'Bandwidth_GB_Year'])
```

## 2. Rename Columns for Consistency:

To ensure consistency and follow Python naming conventions, column names are converted to snake_case. This step simplifies later references to these columns in the code.

**Code Segment**:

```
col_head = {
    'Children': 'children',
    'Age': 'age',
    'OnlineSecurity': 'online_security',
    'Tenure': 'tenure',
    'MonthlyCharge': 'monthly_charge',
    'Bandwidth_GB_Year': 'bandwidth_gb_year'
}
df.rename(columns=col_head, inplace=True)
```

## 3. Convert Categorical Variables to Numeric:

The categorical variable `OnlineSecurity`, which originally contains 'Yes' and 'No' values, is converted to binary numeric values (1 for 'Yes', 0 for 'No'). This conversion is necessary because the decision tree model can only process numerical input.

**Code Segment**:

```
df['online_security'] = df['online_security'].map({'Yes': 1, 'No': 0})
```

## 4. Check for Correlations:

A correlation matrix is generated to understand the relationships between the features. This helps to identify any highly correlated variables and is useful for exploratory data analysis before model building.

**Code Segment**:

```
df.corr().abs().style.background_gradient(cmap='coolwarm')
```

## 5. Split the Data into Train and Test Sets:

The dataset is split into independent variables (features) and the dependent variable (`tenure`). Then, the data is further divided into training and test sets, with 80% of the data used for training and 20% for testing. This step is essential for validating the model's performance on unseen data.

**Code Segment**:

```
X = df.drop(columns=['tenure'])
y = df['tenure']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

## 6. Save Processed Data:

The processed training and testing datasets are saved as CSV files for future use. This allows for easy retrieval and prevents the need to reprocess the data in subsequent analyses.

**Code Segment**:

```
X_train.to_csv('X_train_T2.csv')
X_test.to_csv('X_test_T2.csv')
y_train.to_csv('y_train_T2.csv')
y_test.to_csv('y_test_T2.csv')
```

These preprocessing steps ensure that the data is clean, consistent, and in a format suitable for decision tree modeling.

C4:CLEANED DATA SET

See attached.

D1:SPLITTING THE DATA

See attached.

D2:OUTPUT AND INTERMEDIATE CALCULATIONS

### Steps in the Analysis:

1. Feature Selection and Data Preprocessing: The dataset was preprocessed by selecting relevant features and transforming categorical variables into numeric form, ensuring the model can effectively process the data. Features such as `Children`, `Age`,

`OnlineSecurity`, `MonthlyCharge`, and `Bandwidth_GB_Year` were used as input to predict the target variable, tenure.

2. Model Splitting: The dataset was split into training (80%) and testing (20%) sets to allow the model to learn from the training data and be evaluated on unseen test data.

Decision Tree Model Training: The DecisionTreeRegressor from Scikit-learn was employed to build the regression model. The model was trained using the training set to learn the relationships between the independent variables and the target variable (tenure). During the training process, the algorithm split the data into branches by selecting the optimal feature at each node based on reducing the variance in the target variable.

Code:

```
dt = DecisionTreeRegressor(random_state=42)
dt.fit(X_train, y_train)
```

Hyperparameter Tuning: To optimize the model, GridSearchCV was used to perform hyperparameter tuning. The `max_depth` (maximum depth of the tree) and `min_samples_leaf` (minimum samples required in a leaf node) were fine-tuned to prevent overfitting and enhance the model's accuracy. This process evaluates different combinations of hyperparameters to find the best-performing model.

Code:

```
params_dt = {
    'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'min_samples_leaf': [10, 25, 50, 75, 100]
}
grid_dt = GridSearchCV(estimator=dt, param_grid=params_dt,
scoring='r2', cv=10, n_jobs=-1)
grid_dt.fit(X_train, y_train)
```

```
Test set Mean Squared Error: 7.700
Test set R-squared: 0.989
{'max_depth': 9, 'min_samples_leaf': 25}
```

Model Evaluation: After training the model and performing hyperparameter tuning, the best model was evaluated on the test set. The Mean Squared Error (MSE) and R-squared ($R^2$) metrics were used to assess the model's performance. MSE quantifies the average squared difference between the predicted and actual values, while $R^2$ measures how well the independent variables explain the variability in the dependent variable.

Code:

```
y_pred = best_model.predict(X_test)
test_mse = mse(y_test, y_pred)
test_r2 = r2_score(y_test, y_pred)
```

```
Test set Mean Squared Error: 7.700
Test set R-squared: 0.989
```

The Decision Tree Regression technique allows us to determine the most important factors that influence customer tenure. It creates a model that can predict how long a customer will stay with the company based on key features. By visualizing the decision tree, we gain insights into the decision paths that lead to different tenure outcomes, which can help identify potential customer churn points and opportunities for targeted interventions.

D3:CODE EXECUTION

```
#split data into train and test data
X = df.drop(columns=['tenure'])
y = df['tenure']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


#save train and test datasets
X_train.to_csv('X_train_T2.csv')
X_test.to_csv('X_test_T2.csv')
y_train.to_csv('y_train_T2.csv')
y_test.to_csv('y_test_T2.csv')


# Define parameters
params_dt = {
    'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'min_samples_leaf': [10, 25, 50, 75, 100]
}

# Create regressor
dt = DecisionTreeRegressor(random_state=42)

# Perform GridSearchCV
grid_dt = GridSearchCV(estimator=dt, param_grid=params_dt, scoring='r2', cv=10, n_jobs=-1)

# Fit GridSearch to training data
```

```
grid_dt.fit(X_train, y_train)

# Get the best estimator from the grid search
best_model = grid_dt.best_estimator_

# Predict on the test set
y_pred = best_model.predict(X_test)

# Calculate and print test set metrics
test_mse = mse(y_test, y_pred)
print('Test set Mean Squared Error: {:.3f}'.format(test_mse))

test_r2 = r2_score(y_test, y_pred)
print('Test set R-squared: {:.3f}'.format(test_r2))

# Display best parameters
print(grid_dt.best_params_)


#build the tuned model
dt = DecisionTreeRegressor(max_depth=9, min_samples_leaf=25, random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)
test_mse = mse(y_test, y_pred)
print('Test set Mean Squared Error: {:.3f}'.format(test_mse))
test_r2 = r2_score(y_test, y_pred)
print('Test set R-squared: {:.3f}'.format(test_r2))
```

E1:ACCURACY AND MSE

The results of the prediction model indicate a strong performance in predicting customer tenure using the Decision Tree Regression model. Here's an explanation of the two key metrics:

## 1. Mean Squared Error (MSE):

- **Test set Mean Squared Error**: 7.700
- MSE measures the average squared difference between the predicted values and the actual values. A lower MSE indicates that the model's predictions are close to the actual data points. In this case, the MSE of 7.700 is relatively low, meaning the model's predictions for customer tenure are quite accurate, with minimal error.

## 2. R-squared ($R^2$):

- **Test set R-squared**: 0.989

- R-squared represents the proportion of the variance in the target variable (customer tenure) that is explained by the independent variables in the model. An $R^2$ value of 0.989 indicates that 98.9% of the variability in customer tenure can be explained by the model. This high $R^2$ value suggests that the model fits the data exceptionally well and is capable of making highly accurate predictions.

The model, with an optimal depth of 9 and a minimum of 25 samples per leaf, demonstrates excellent predictive power. The low MSE and the near-perfect $R^2$ value highlight the model's ability to accurately capture the relationship between the independent variables and customer tenure, making it a reliable tool for predicting how long customers will stay with the company.

E2:RESULTS AND IMPLICATIONS

The results of the prediction analysis show that the Decision Tree Regression model is highly effective at predicting customer tenure, as demonstrated by the low Mean Squared Error (MSE) of 7.700 and a very high R-squared ($R^2$) value of 0.989. These results indicate that the model can explain almost all the variability in the target variable (tenure) based on the features used in the analysis.

## Key Results:

1. **Accuracy of the Model**: The high $R^2$ value (0.989) indicates that the model fits the data exceptionally well, meaning it can accurately predict customer tenure based on factors such as age, number of children, monthly charges, bandwidth usage, and online security. This high accuracy suggests that the model has identified the most relevant features that influence how long customers stay with the company.
2. **Mean Squared Error**: The relatively low MSE of 7.700 implies that the model's predictions are close to the actual values, meaning the errors between the predicted and real tenure values are small. This makes the model reliable for identifying customers who are likely to stay for a shorter or longer period.

## Implications of the Prediction Analysis:

1. **Improved Customer Retention Strategies**: The results provide actionable insights into the factors that contribute most to customer tenure. With this information, the company can develop targeted retention strategies. For example, customers with lower tenure predictions can be offered customized incentives or enhanced services to increase their loyalty and reduce churn.
2. **Resource Allocation**: By knowing the key factors influencing tenure, the company can optimize the allocation of resources. For instance, it can focus customer service and marketing efforts on customers at risk of churning, thus potentially reducing costs associated with acquiring new customers and improving overall customer satisfaction.
3. **Product and Service Enhancement**: The insights from the model can guide product development by highlighting services (e.g., online security) that may have a significant

impact on customer retention. Offering improved or additional services could increase customer engagement and encourage long-term commitments.

4. **Proactive Interventions**: With a predictive model in place, the company can implement proactive interventions by identifying customers who may be at risk of leaving. Early intervention strategies, such as personalized offers or improved customer support, can help extend customer tenure and prevent churn.

The results of the analysis suggest that the Decision Tree Regression model is a powerful tool for predicting customer tenure, providing valuable insights that can help the company strengthen its customer retention efforts, optimize resources, and potentially increase overall profitability.

E3:LIMITATION

One limitation of this data analysis is the potential for overfitting inherent in decision tree models. Although hyperparameter tuning (e.g., adjusting the maximum depth and minimum samples per leaf) was performed to mitigate this issue, decision trees are still prone to capturing noise or overly specific patterns in the training data. This can lead to a model that performs exceptionally well on the training set but may not generalize as effectively to new, unseen data. While the model's performance metrics, such as the high R-squared value and low MSE, suggest strong predictive power on the test set, there's a possibility that the model may struggle with different datasets or future customer data that exhibits new patterns or slight variations. This overfitting risk is especially pronounced when the dataset is relatively small or does not capture the full range of customer behaviors.

E4:COURSE OF ACTION

Based on the results and implications discussed in part E2, I recommend that the organization implement targeted retention strategies using the model's insights into key factors influencing customer tenure, such as monthly charges, bandwidth usage, age, and online security features. Customers with shorter predicted tenure should be offered personalized incentives, such as discounts or additional services, to extend their relationship with the company. Additionally, the company should focus on improving high-impact services like Online Security, which could strengthen customer loyalty. Resource allocation should be optimized by concentrating efforts on customers more likely to churn, thus reducing costs and improving the effectiveness of retention strategies. Proactive customer engagement, such as setting up alerts to flag at-risk customers, would allow the company to address issues early and potentially prevent churn. It is also important to regularly monitor and retrain the model as new data becomes available to ensure its continued relevance. Moreover, considering the use of ensemble techniques like Random Forests could further enhance prediction accuracy and reliability, enabling more effective retention efforts. These actions, driven by the model's predictions, can help the organization reduce churn, improve customer retention, and enhance overall profitability.

G:SOURCES FOR THIRD-PARY CODE

"Reading Specific Columns of a CSV File Using Pandas." *GeeksforGeeks*, GeeksforGeeks, 3 Dec. 2023, www.geeksforgeeks.org/reading-specific-columns-of-a-csv-file-using-pandas/.