Data Cleaning and Feature Engineering Report

1. Data Overview
    a. Missing Values: No missing values found across all columns.
    b. Data Types: Reviewed the data type of each column and converted them in the next step, where applicable.
2. Data Cleaning
    a. Boolean variables (YES/NO) converted to binary (0/1):
    Partner, Dependents, PhoneService, PaperlessBilling, and Churn were mapped from "Yes"/"No" to 1/0.
    b. Standardization for service features:
    Columns with responses "Yes", "No", and "No internet service" (e.g., OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, OnlineSecurity) were encoded as 1, 0, and 2.
    c. The following screenshot shows the resulting data types. The new dataset was saved as a pickle to retain these data types.

```
[157]: print(df.dtypes)

customerID          object
gender              object
SeniorCitizen        int64
Partner              int64
Dependents           int64
tenure               int64
PhoneService         int64
MultipleLines       object
InternetService     object
OnlineSecurity       int64
OnlineBackup         int64
DeviceProtection     int64
TechSupport          int64
StreamingTV          int64
StreamingMovies      int64
Contract            object
PaperlessBilling     int64
PaymentMethod       object
MonthlyCharges     float64
TotalCharges       float64
Churn                int64
```

3. Issue with TotalCharges column
    a. Initially, TotalCharges was of data type object. An error resulted when trying to convert to a float, as this is a monetary observation (a decimal). The error showed blanks or " ". Upon initial observation, there were no missing values, so I converted these blanks to zeros.

```
df['TotalCharges'] = df['TotalCharges'].str.strip().replace('', '0').astype(float)
```

To Check that these values should remain zeros, first see how many there are. Then, look at the actual observations.

```
[139]: # Check and make sure that TotalCharges didn't get messed up.
       print((df['TotalCharges'] == 0).sum())
```

```
11
```

```
[143]: print(df[df['TotalCharges'] == 0].head(20))
```

```
MonthlyCharges  TotalCharges  Churn  tenure  \
         52.55           0.0      0       0
         20.25           0.0      0       0
         80.85           0.0      0       0
         25.75           0.0      0       0
         56.05           0.0      0       0
         19.85           0.0      0       0
         25.35           0.0      0       0
         20.00           0.0      0       0
         19.70           0.0      0       0
         73.35           0.0      0       0
         61.90           0.0      0       0
```

After conversion, 11 records showed TotalCharges = 0, which didn't make sense because they had nonzero MonthlyCharges. Root cause: These customers had tenure = 0 (had not completed a full month). The system recorded TotalCharges as blank, which was converted to 0. Resolution: For these cases, TotalCharges was replaced with the value from MonthlyCharges:

```
[149]: # Change zeros to match MonthlyCharges
       df.loc[df['TotalCharges'] == 0, 'TotalCharges'] = df['MonthlyCharges']
       print(df[df['tenure'] == 0][['MonthlyCharges', 'TotalCharges']])
```

```
      MonthlyCharges  TotalCharges
488            52.55         52.55
753            20.25         20.25
936            80.85         80.85
1082           25.75         25.75
1340           56.05         56.05
3331           19.85         19.85
3826           25.35         25.35
4380           20.00         20.00
5218           19.70         19.70
6670           73.35         73.35
6754           61.90         61.90
```

4. Feature Engineering
   Several new variables were created to enhance analysis and prediction:

   a. Average Monthly Revenue (AvgMonthlyRevenue)
      - TotalCharges / tenure (tenure adjusted to 1 if zero).
      - Measures average spend per month, highlighting high-value vs low-value customers.
   b. Tenure Buckets (TenureGroup)
      - Binned into categories: 0–1yr, 1–2yr, 2–4yr, 4–6yr.
      - Helps analyze churn by customer lifecycle stage.
   c. Number of Services (NumServices)
      - Counts how many optional services (e.g., OnlineBackup, TechSupport) a customer subscribed to.
      - Indicates product adoption and engagement.
   d. Autopay Indicator (IsAutoPay)
      - Created from PaymentMethod to flag if the customer uses automatic payments.
      - Autopay customers may have lower churn risk.
   e. Churn Risk Score (RiskScore)
      - Composite indicator combining:
        - Short tenure (< 12 months)
        - Month-to-month contract
        - Above-median monthly charges
      - Higher score = higher churn risk.
   f. Senior Tech Usage (SeniorHighTech)
      - Flags seniors (SeniorCitizen = 1) with more than 3 services.
      - Identifies a niche segment of tech-savvy older customers.