**Part A: Research Question**

1.  Can we estimate if a customer will churn based on their, services, demographic details, and survey responses?

2.  The goal of this analysis is to produce a KNN algorithm to estimate if a customer will churn based several variables in our data set. This would give a company valuable insight to why customers churn, how to anticipate churn, and how to prevent customers from churning.

**Part B: Method Justification**

1.  KNN is a great, simple method for estimating variables. It has no underlying assumptions about the distribution of the data. Also, it doesn’t need the dependent variables to have linear relationships so it can handle a more robust set of inputs.

2. KNN assumes that, when graphed, the variables exist in relative proximity to each other. This is important because the whole premise is estimated outcomes based on neighbor data points.

3.  List the packages or libraries you have chosen for Python or R, and justify how each item on the list supports the analysis:

* + - Pandas: Managing the dataset
    - Numpy: Performing mathematical operations on arrays.
    - MatPlotlib: Graphing.
    - Seaborn: Graphing.
    - Sklearn: Statistical modeling.

**Part C: Data Preparation**

1.  The goal of the data preparation phase is to select, transform and train the data in a way that will produce a strong KNN algorithm.

1. Numeric Features - These are numeric variables that include quantitative data.
   * Age: Numeric (represents the age of the customer).
   * Income: Numeric (represents the customer's income).
   * Children: Numeric (represents the number of children a customer has).
   * Outage\_sec\_perweek: Numeric (represents the number of seconds per week the customer experiences outages).
   * Email: Numeric (likely represents the number of emails sent to the customer).
   * Contacts: Numeric (represents the number of contacts between the customer and the company).
   * Yearly\_equip\_failure: Numeric (represents the number of equipment failures experienced yearly).
   * MonthlyCharge: Numeric (represents the monthly charge billed to the customer).
   * Bandwidth\_GB\_Year: Numeric (represents the bandwidth usage in gigabytes per year).

Categorical Features - These variables contain qualitative data.

* + Techie: Categorical (A binary feature indicating whether the customer considers themselves tech-savvy).
  + Contract: Categorical (represents the type of contract the customer has, e.g., month-to-month, one year, two years).
  + Port\_modem: Categorical (binary, indicating whether the customer has a portable modem).
  + Tablet: Categorical (binary, indicating whether the customer owns a tablet).
  + InternetService: Categorical (type of internet service, e.g., DSL, Fiber Optic, None).
  + Phone: Categorical (binary, indicating whether the customer has a phone service).
  + Multiple: Categorical (binary, indicating whether the customer uses multiple lines).
  + OnlineSecurity: Categorical (binary, indicating whether the customer has online security services).
  + OnlineBackup: Categorical (binary, indicating whether the customer uses online backup services).
  + DeviceProtection: Categorical (binary, indicating whether the customer has device protection).
  + TechSupport: Categorical (binary, indicating whether the customer has tech support services).
  + StreamingTV: Categorical (binary, indicating whether the customer uses streaming TV services).
  + StreamingMovies: Categorical (binary, indicating whether the customer uses streaming movie services).
  + PaperlessBilling: Categorical (binary, indicating whether the customer has opted for paperless billing).
  + PaymentMethod: Categorical (represents the method used by the customer for payments, e.g., electronic check, mailed check, bank transfer, credit card).

1. The first step was to convert ‘Churn’ to binary since it will be our target variable.

A computer screen shot of a computer code

Description automatically generated

Second, I filtered the data for the variables I wanted to include in the KNN algorithm. I chose a breadth of variables that cover customer demographic, services, usage, and customer engagement.

A screenshot of a computer program

Description automatically generated

Third, I used one hot encoding to convert the variables with dtype object to dummie variables. This is important so they can be properly fed into the algorithm.

A close-up of a computer code

Description automatically generated

Last, I separated the target variable (‘Churn’) and normalized the remaining numeric data. This is important because KNN measures distance between points.

A screenshot of a computer code

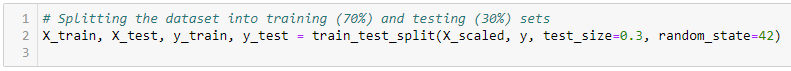
Description automatically generated

A screenshot of a computer program

Description automatically generated

**Part D: Analysis**

1.  I split the data into training and test data using a 70/30 split.



Provide a copy of the cleaned data set. – ***See Attached***

A screenshot of a computer program

Description automatically generated

2.  To figure out the appropriate K value I charted a range of K values against their produced accuracy score and found a value of 25 to be near optimal and then performed KNN classification.

3.  A screenshot of a computer program

Description automatically generated

A graph with a blue line

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Part E: Data Summary and Implications**

1. To summarize area under the curve I plotted a ROC curve. This roc curve analysis shows the true predicted values on the Y axis and false positives on the x axis. The Roc curve area shows a value of .89 which is a very strong score as the maximum theoretical value could be 1. This also means it has a strong probability of correctly classifying a randomly chosen positive input.

A screen shot of a graph

Description automatically generated

2.  As stated above the model has and accuracy score of .8156 and AOC score of .89 which means it is accurate at correctly classifying inputs to churn or no churn. When we look at eh Confusion matrix, we will see that the model is particularly good at classifying non churn instances. This makes sense given the data set and its distribution of churn and no churn. That would also be a limitation of this classification model because it is stronger at predicting actual negatives. There is a normal trade of between these two abilities.

1. My recommendation would be to employ this model specifically in the light of estimating if a customer will not churn. Given this information we could build a profile of a loyal customer and aim to onboard or build customer profiles that match that.

**Part F: Demonstration**

 Will upload once the rest of the submission has been reviewed and I know I don’t have to change the code.

**Part G.  Web Sources.**

<https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn>

<https://stackoverflow.com/questions/63410524/k-nearest-neighbour-classifier-random-state-for-train-test-split-leads-to-diff>