**Preparing Churn Data for Analysis**

Kelseyann Wright

Student ID: 00565814

College of Information Technology, Western Governors University

Data Cleaning - D206

Dr. Eric Straw

March 5, 2023

**Preparing Churn Data for Analysis**

1. **Question or Decision**

What factors drive customer churn? This question is important because procuring customers is more expensive than retaining customers. Customer churn in the telecommunications industry is common. Understanding and targeting the factors that drive customer churn will allow the business to reduce churn rates and money spent on customer acquisition.

1. **Required Variables**

*Table 1: Variable Summary for Churn Dataset*

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Description** | **Type** | **Example** |
| Case Order | Original data order | Quantitative | 1 |
| Customer\_id | Unique identifier for each customer (primary key) | Qualitative | K409198 |
| Interaction, UID | Unique identifier for customer interactions including support, enrollment, and transactions | Qualitative | aa90260b-4141-4a24-8e36-b04ce1f4f77b |
| City | Billing city | Qualitative | Point Baker |
| State | Billing state | Qualitative | AK |
| County | Billing county | Qualitative | Prince of Wales-Hyder |
| Zip | Billing zip code | Qualitative | 99927 |
| Lat | Latitude of billing address | Quantitative | 56.25100 |
| Lng | Longitude of billing address | Quantitative | -133.37571 |
| Population | Census data of population within one mile of billing address | Quantitative | 10449 |
| Area | Classification of rural, urban, or suburban area from census data | Qualitative | Urban |
| TimeZone | Customer time zone | Qualitative | America/Detroit |
| Job | Customer job | Qualitative | Solicitor |
| Children | Number of children at the time of enrollment | Quantitative | 4.0 |
| Age | Age of customer at time of enrollment | Quantitative | 68.0 |
| Education | Customer’s highest degree at time of enrollment | Qualitative | Master’s Degree |
| Employment | Customer’s employment status at time of enrollment | Qualitative | Part Time |
| Income | Customer’s income at time of enrollment | Quantitative | 28561.99 |
| Marital | Customer’s marital status at time of enrollment | Qualitative | Widowed |
| Gender | Customer’s gender | Qualitative | Male |
| Churn | Whether service was ended in the last month | Qualitative | No |
| Outage\_sec\_perweek | Average outage seconds per week in customer’s neighborhood | Quantitative | 6.972566 |
| Email | Count of emails sent to the customer in the last year | Quantitative | 10 |
| Contacts | The number of times the customer contacted technical support | Quantitative | 0 |
| Yearly\_equip\_failure | Number times the customer’s equipment had to be reset or replaced due to failure in the last year | Quantitative | 1 |
| Techie | Does the customer identify as technically inclined? | Qualitative | No |
| Contract | Term of contract | Qualitative | Two Year |
| Port\_modem | Does the customer have a portable modem? | Qualitative | No |
| Tablet | Does the customer have a tablet? | Qualitative | No |
| InternetService | Type of internet service | Qualitative | DSL |
| Phone | Does the customer have phone service? | Qualitative | Yes |
| Multiple | Does the customer have multiple services? | Qualitative | No |
| OnlineSecurity | Does the customer have online security? | Qualitative | No |
| OnlineBackup | Does the customer have an online back-up add on? | Qualitative | No |
| DeviceProtection | Does the customer have device protection? | Qualitative | No |
| TechSupport | Does the customer have technical support add ons? | Qualitative | No |
| StreamingTV | Does the customer have streaming TV? | Qualitative | No |
| StreamingMovies | Does the customer have streaming movies? | Qualitative | No |
| PaperlessBilling | Is the costumer enrolled in paperless billing? | Qualitative | Yes |
| PaymentMethod | Customer payment method | Qualitative | Credit Card (automatic) |
| Tenure | Number of months with same provider | Quantitative | 6.795513 |
| MonthlyCharge | Average amount charged monthly | Quantitative | 171.449762 |
| Bandwidth\_GB\_Year | Average GB of data annually | Quantitative | 904.536110 |
| Item1 | Customer rating of importance of timely response (1 = high, 8 = low) | Quantitative | 2 |
| Item2 | Customer rating of importance of timely fixes (1 = high, 8 = low) | Quantitative | 2 |
| Item3 | Customer rating of importance of timely replacements (1 = high, 8 = low) | Quantitative | 2 |
| Item4 | Customer rating of importance of reliability (1 = high, 8 = low) | Quantitative | 2 |
| Item5 | Customer rating of importance of options (1 = high, 8 = low) | Quantitative | 2 |
| Item6 | Customer rating of importance of respectful response (1 = high, 8 = low) | Quantitative | 2 |
| Item7 | Customer rating of importance of courteous exchange (1 = high, 8 = low) | Quantitative | 2 |
| Item8 | Customer rating of importance of active listening (1 = high, 8 = low) | Quantitative | 2 |

1. **Cleaning Plan**

**C1. Plan to Assess Quality of Data**

The steps followed to assess the quality of the data are listed below.

1. Use the “head” and “info” functions to preview the dataset. The purpose of this step is to confirm that data imported correctly, check for an index field, and to ensure that all fields have appropriate names and data types. This step will identify categorical variables that need to be re-expressed as numeric.
2. Check for duplicates in the dataset using the “duplicated” function in the python “pandas” library. When checking for duplicates, check for complete duplicates and duplicates in specific fields (i.e., CustomerID).
3. Use minimum and maximum length functions to check lengths of string variables and assess inconsistencies.
4. Use histograms, boxplots, and z-scores of quantitative features to identify outliers and get familiar with the distributions of each feature.
5. Determine the amount of and location of missing data in the dataset by assessing “NaN” values and misleading data values.
6. Use matrix plots and correlation heat maps to assess the randomness of the missing data and determine how it should be handled.

**C2. Justification of Approach**

The key characteristics being assessed are structure, accuracy, and completeness. All of these characteristics are important for determining the types of statistical analysis and modeling that can be used to analyze the dataset (Larose & Larose, 2019). The goal of this dataset and analysis is to understand patterns in customer churn. A complete dataset with appropriate structure and accuracy will be crucial to execute predictive modeling techniques that will allow the company to predict and reduce customer churn.

The first characteristic, structure, was analyzed by previewing the dataset (see C1, step 1). In this step, data was checked for general structure, proper feature names, an index field, and appropriate data types. Index fields are important for keeping track of the order of the data (Larose & Larose, 2019). Field names and data types are important for determining appropriate statistical analysis and modeling methods (Larose & Larose, 2019) and help identify issues that arose during import of the data. In addition, this step helps identify categorical variables that would be better re-expressed as numeric values so they can be used in statistical models (Larose & Larose, 2019).

The second characteristic, accuracy, was assessed by checking for duplicates, outliers, and inconsistent string variables. This “dirty data” can drastically impact statistical results in later analyses if not dealt with properly (Larose & Larose, 2019). Duplicates were assessed using the “duplicated” function in the python “pandas” library. This function allows for identification of complete and partial duplicates (McKinney, 2010). Histograms of quantitative variables were created to identify outliers, assess data distributions, and identify misleading values in the quantitative data (Larose & Larose, 2019). In addition, z-scores were calculated and used to identify specific outliers in the dataset (Larose & Larose, 2019). To check for consistency of string variables, minimum and maximum variable lengths were determined. This method can identify data entry errors or inconsistencies in categorical data columns with restricted lengths, such as yes/no columns or state abbreviations (Nehme, n.d.).

The third characteristic, completeness, was assessed by identifying missing values in the dataset and using multiple visualizations and functions to determine the randomness of the missing values. Missing data can affect the conclusions drawn from statistical analysis (Larose & Larose, 2019), and therefore must be handled before modeling. The “missingno” package in python was used to analyze missing data and assess its randomness. It is important to understand the randomness of the missing data to identify how to treat it (Larose & Larose, 2019).

**C3. Justification of Tools**

Python programming language was used to clean the data. Python was chosen for its versatility, readability, and diverse functions. In addition, the syntax is consistent and simple to interpret, which is ideal when sharing your analysis with others who may not have coding experience. R does not have consistent syntax across libraries, making it more difficult to understand learn and understand.

Several python libraries were used to clean the data, including pandas, numpy, matplotlib, missingno, scipy, and sci-kit learn. Pandas is a common library used for working with dataframes in python. It includes many tools for computations and analysis on dataframes (McKinney, 2010), including functions like head, info, columns, and len which are used frequently to assess data and check if data manipulations and transformations were completed as intended. Numpy is used for many summary statistics throughout the analysis process and for replacing misleading values with NaN (Harris et al., 2020; Larose & Larose, 2019). Scipy and sci-kit learn libraries work together to offer various statistical models used to impute missing data (Pedregosa et al., 2011; Virtanen et al., 2020). The “missingno” library includes many useful data visualization tools for assessing missing values and their randomness (Biloger, 2018).

**C4. Provide the Code**

See attached jupyter notebook file titled “D206.ipynb”. The annotated code is in the “Assess Data Quality” section of the notebook file.

1. **Cleaning Summary**

**D1. Cleaning Findings**

Several quality issues were identified when assessing the structure of the dataset using the “head” and “info” functions. First, the original index column was “unnamed” when loaded into Python (Figure 1). In addition, this original index starts at 1, while Python indexes start at 0. Second, several features (i.e., education, contract, and several yes/no logical columns) are classified as “object” string variables may be able to be re-expressed as numeric for modeling purposes.

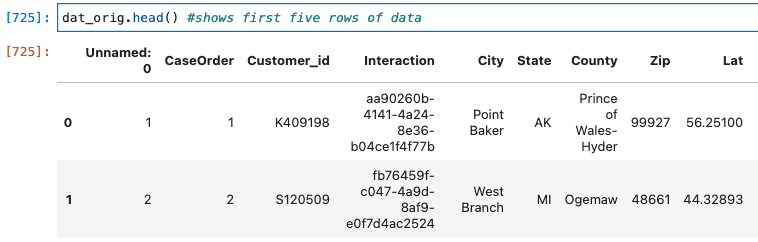


Figure 1: Preview of data after initial upload. Notice the first column is named "Unnamed: 0". This column represents the original index column of the dataset.

Diagram

Description automatically generated No duplicates were identified in the dataset, but several potential outliers were identified when viewing the histograms of the quantitative variables in the dataset (Figure 2). Upon closer examination, it was determined that in the “population” feature there are an unusually large number of zero values. A second feature, “area”, gave some insight into this issue. The “area” feature classifies the location as urban, rural, or suburban. If the zeros in the population feature were true zeros, I would have expected them to be in rural areas. However, there was an almost equal amount of zero values in urban, rural, and suburban areas. Therefore, it was concluded that these zeros in the population feature are likely missing data values and were replaced with “NaN” so they could be imputed with the rest of the missing data later on.

Figure 2:Histograms of quantitative variables.

Table

Description automatically generatedOutliers were identified in all quantitative variables statistically using z-scores (Larose & Larose, 2019). The total number of outliers, defined as having a z-value less than -3 or greater than 3 (Larose & Larose, 2019), was 1123 (Figure 3). The “outage\_sec\_perweek” variable contained the most outliers, with a total of 491. Outliers were visualized using boxplots (Figure 4).

Figure 3: Total outliers identified with z-scores for each column with outliers.

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Figure : Boxplots of all features with outliers.

Table

Description automatically generatedMissing values were found in 8 out of the 52 variables (Figure 5). After identifying the missing values and their location, the randomness of the missing values was analyzed using visualizations from the missingno python library. The matrix plot (Figure 5) shows the location of missing values in the rows of the dataset, while the correlation heat map (Figure 6) shows nullity correlations. Both of these plots confirmed that the missing values were randomly distributed.

Figure 5: percent of values missing for all fields with missing data.

Table

Description automatically generated

Figure : Matrix plot of missing data

**Chart, histogram

Description automatically generated**

Figure : Nullity correlations of variables with missing data.

To check for consistency in data entry and for issues in string variables, minimum and maximum length variables for each feature were identified. Through this analysis, it became apparent that there was an issue with the zip code variable. All zip codes should be five digits, but some zip codes (773 of them) had fewer than 5 digits. The issue was investigated further by previewing the data and using the internet to search for the actual zip codes of the cities with the short zip codes. This led to the discovery that the leading zeros in zip codes were removed when the data was imported as numeric and will need to be fixed during cleaning. This step in the analysis was also useful to confirm that there were likely not any data entry issues in the “yes/no” columns, as they all had a minimum length of 2 and a maximum length of 3, and in the state column, which had a minimum and maximum length of 2. These variable lengths are consistent with the expected lengths.

**D2. Justification of Mitigation Methods**

Table

Description automatically generatedThe first step of mitigation was re-expressing variables. First, the zip code variable was changed to a string variable, and the leading zeros were added back in. Typically, numeric variables would not be re-expressed as string variables. However, in this case it is necessary to keep the correct zip codes in case they were needed for mailing purposes. Numeric data for location is still available in the latitude and longitude columns, so there should not be loss of location information in future models.

Table 2: Re-expression of education and contract features as numeric.

The education and contract variables were string variables that were re-expressed as numeric in new features, “education\_num” and “contract\_num” (Table 2). Adding new columns for the numeric data allows for the original data to be conserved while providing a numeric measure of education level and contract length to be used in future modeling (Larose & Larose, 2019). In addition, features classified as “object” with ‘yes/no’ values were replaced with ‘0/1’ values, where “no” was changed to zero and “yes” was changed to one. This will allow these data to be used by models in future analysis and for imputation (Larose & Larose, 2019).

The outliers that were identified were not removed from the dataset. Based on the location of the outliers and the feature distributions (Figure 2, Figure 4), these statistical outliers don’t appear to be random data entry mistakes, but rather true outliers that are inherent to the dataset and may provide valuable information to the client. For instance, the large number of outliers in the “outage\_sec\_perweek” feature include values that are larger than the rest of the data distribution. This would indicate a group of customers that had longer or more frequent outages than others. This should be researched further during data analysis to determine if there is a pattern associated with these increased outages, such as their relationship with location or type of internet service. Outliers can provide valuable information for analysis and removing them for the sake of removing them may not be best as it could cause a loss of valuable information to the company (Larose & Larose, 2019).

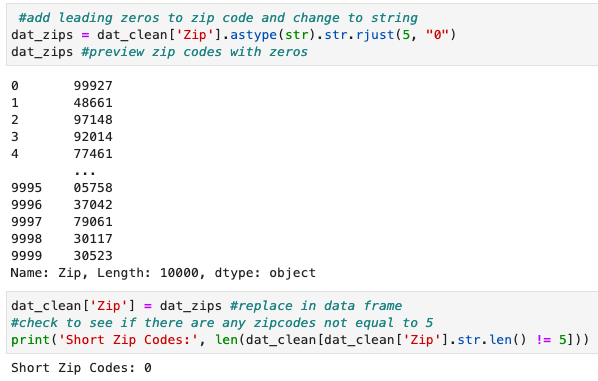
After variables were re-expressed, missing data were imputed. Several methods were compared for imputation, including all variations of the iterative imputer in the python sci-kit learn (Pedregosa et al., 2011; Virtanen et al., 2020). After comparing histograms for variables after each imputation method, Chart

Description automatically generatedthe k-Nearest Neighbors (KNN) approach (Larose & Larose, 2019; Malarvizhi & Thanamani, 2012) was selected because it resulted in the least kurtosis of distributions (Yamada et al. 2015), specifically in the ‘Income’ variable (Figure 8). This approach was executed using the “KNNimputer” function in the sci-kit learn python library.

Figure 8: Comparison of iterative imputation with a Bayesian Ridge function (A) and k-Nearest Neighbors (KNN) imputation (B) for the "Income" variable.

After imputing the missing data, several “float” variables were generated that should be integers (children, techie, phone, age, and techsupport). Those features were converted back to integers using the “astype(int)” function in python. Computations are faster on integer variables than float variables. This change should make later modeling code run more efficiently.

Data was not normalized or standardized during the cleaning process. Although some models and machine learning algorithms require standardization for optimum performance (Larose & Larose, 2019), this varies with different techniques and can easily be done for each necessary analysis during the analysis phase. In addition, some R and python models have built-in functions or options for standardization and normalization of data.

**D3. Summary of the Outcomes**

The re-expression of zip code as a string variable led to all zip codes in the dataset being 5 characters long (Figure 9), as expected for U.S. zip codes. In addition, re-expressing the education and customer fields as numeric, as well as re-expressing the yes/no categorical variables as 0/1 led to several more numeric variables that will be available for future modeling (Figure 10, Table 3).

Figure 9: Preview of new zip code data as well as calculation of the number of zip codes with a length not equal to 5 after the data was re-expressed.

Imputation of missing data resulted in all features having 10000 values (Figure 10, Table 3). In addition, all appropriate features were re-assigned as integers (Table 3). The final clean dataset with the addition of the numeric “education” and “customer” fields contains 10,000 rows and 54 columns (Table 3). There are some statistical outliers remaining in the dataset according to the z-scores, but it was concluded that these outliers are true outliers that may provide valuable information for the company and should not be removed.

Diagram, engineering drawing

Description automatically generated

Figure : Histograms of numeric features in cleaned dataset.

*Table 3: Final features, data types, and number of non-nulls for cleaned dataset.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature | data\_type | count\_non\_null | Feature | data\_type | count\_non\_null |
| *OrigIndex* | int64 | 10000 | *Contract* | object | 10000 |
| *CaseOrder* | int64 | 10000 | *Port\_modem* | int64 | 10000 |
| *Customer\_id* | object | 10000 | *Tablet* | int64 | 10000 |
| *Interaction* | object | 10000 | *InternetService* | object | 10000 |
| *City* | object | 10000 | *Phone* | int64 | 10000 |
| *State* | object | 10000 | *Multiple* | int64 | 10000 |
| *County* | object | 10000 | *OnlineSecurity* | int64 | 10000 |
| *Zip* | object | 10000 | *OnlineBackup* | int64 | 10000 |
| *Lat* | float64 | 10000 | *DeviceProtection* | int64 | 10000 |
| *Lng* | float64 | 10000 | *TechSupport* | int64 | 10000 |
| *Population* | int64 | 10000 | *StreamingTV* | int64 | 10000 |
| *Area* | object | 10000 | *StreamingMovies* | int64 | 10000 |
| *Timezone* | object | 10000 | *PaperlessBilling* | int64 | 10000 |
| *Job* | object | 10000 | *PaymentMethod* | object | 10000 |
| *Children* | int64 | 10000 | *Tenure* | float64 | 10000 |
| *Age* | int64 | 10000 | *MonthlyCharge* | float64 | 10000 |
| *Education* | object | 10000 | *Bandwidth\_GB\_Year* | float64 | 10000 |
| *Employment* | object | 10000 | *item1* | int64 | 10000 |
| *Income* | float64 | 10000 | *item2* | int64 | 10000 |
| *Marital* | object | 10000 | *item3* | int64 | 10000 |
| *Gender* | object | 10000 | *item4* | int64 | 10000 |
| *Churn* | int64 | 10000 | *item5* | int64 | 10000 |
| *Outage\_sec\_perweek* | float64 | 10000 | *item6* | int64 | 10000 |
| *Email* | int64 | 10000 | *item7* | int64 | 10000 |
| *Contacts* | int64 | 10000 | *item8* | int64 | 10000 |
| *Yearly\_equip\_failure* | int64 | 10000 | *Education\_num* | int64 | 10000 |
| *Techie* | int64 | 10000 | *Contract\_num* | int64 | 10000 |

**D4. Mitigation Code**

See attached notebook file titled “D206.ipynb”.

**D5. Clean Data**

See attached .csv file titled “churn\_clean\_data.csv”

**D6. Limitations**

There are several limitations in the data cleaning process. First, using the “duplicated” function in python identifies complete duplicates, or duplicates in a specified column. It is possible to “miss” duplicate values if the analyst does not look in the correct column(s). In addition, if the analyst only checks for complete duplicates, they may miss duplicates in specific columns.

Limitations also exist when using z-scores for outlier detection. Z scores will detect statistical outliers. Sometimes outliers come from errors in measurement or entry, but sometimes they are true measurements that provide valuable information for the analysis. Removing all values with z-scores outside the accepted range could result in loss of important information. Analysts must consider this when using statistical analyses to identify and remove outliers and should consult with the project manager before removing all outliers.

Data imputation has limitations as well. Imputed data is a statistical estimate, and there are many techniques for imputing missing data. Each technique can result in slightly different estimates for missing data, and those estimates can impact future models and analysis (Baneshi & Talei, 2012). Depending on the amount of data imputed, this could influence conclusions or predictions made and change data-driven decisions.

A final limitation of the cleaned data is that it is not standardized or normalized. It is important for some modeling and machine learning techniques to have standardized numeric variables when numeric fields have vastly different ranges (Larose & Larose, 2019). For instance, in this dataset, the “income” feature has values in the tens of thousands and hundreds of thousands, while the “age” feature is less than 100 and many other features have values less than or equal to 1. In some models, the “income” field would have more influence due to the large numeric values.

**D7. Impact of Limitations**

The goal of this dataset is to understand the factors that influence customer churn. To address this question, several types of models may be used in later analysis. Some features of the dataset had more missing data than others, and the imputed data for those features may influence model predictions (Baneshi & Talei, 2012). If features with a large amount of imputed data are determined to be important, it will be crucial to convey to the client the amount of data that was imputed and how that may influence results. Finally, the numeric variables in the dataset will need to be normalized or standardized for certain modeling and machine learning approaches (Larose & Larose, 2019). However, this is a minimally cumbersome task in R and Python and can be done at the time of modeling so that the standardization or normalization approach can best suit the model or algorithm being used.

1. **Principal Component Analysis (PCA)**

**E1. Principal Components**

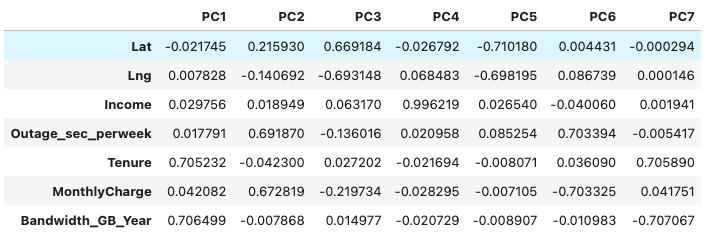
 The principal component analysis (PCA) included 7 principal components: 'Lat', 'Lng', 'Income', 'Outage\_sec\_perweek', ‘Tenure’, ‘MonthlyCharge’, and “Bandwidth\_GB\_Year’. All of these are continuous numeric variables, which is the preferred data type for PCA. The loading data (Table 4) was exported to a .csv file and is attached (“churn\_pca\_loadings.csv”).

Table 4: PCA Loadings

**E2. Criteria Used**

The PCA analysis revealed that 3 principal components would be best for this dataset (Figure 11). After 3 components, the eigenvalues fall below 1.0 (Figure 11).

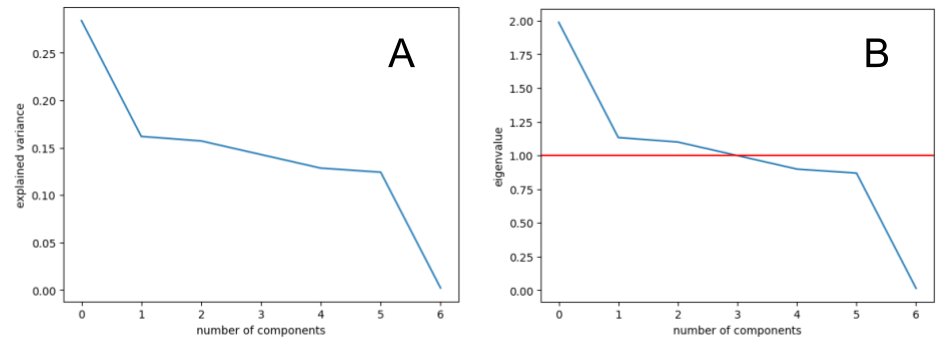


Figure : Scree plot (A) and Eigenvalues (B) for PCA analysis.

**E3. Benefits**

PCA is useful because it helps reduce the number of features in the dataset, which can improve efficiency of models and reduce feature correlation when modeling. Large amounts of feature correlation can negatively affect model predictions, so reducing feature correlation is incredibly important. From the PCA analysis on the churn dataset, it was concluded that no more than 3 principal components are required. Table 5 lists the selected principal components and the highest correlated features from the “loadings” table for each. Based on this information, we can select which features would be best for future modeling of the dataset.

Table 5: Selected PCA components and correlated features

|  |  |
| --- | --- |
| **Component** | **Top Correlated Features (loading > 0.5).** |
| PC1 | Bandwidth\_GB\_Year, Tenure |
| PC2 | Outage\_sec\_perweek, MonthlyCharge |
| PC3 | Lng, Lat |

1. **Video**

A Panopto video was attached and submitted with the performance task. The link can also be accessed here: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=ad382f2f-c9c6-44df-a891-afc8008d41f2

1. **Sources of Third-Party Code**

No additional web sources of data or third-party code were used.

1. **Sources**

Baneshi, M. R., & Talei, A. R. (2012). Does the missing data imputation method affect the composition and performance of prognostic models?. *Iranian Red Crescent medical journal*, *14*(1), 31–36.

Bilogur, A. (2018). Missingno: a missing data visualization suite. *Journal of Open Source Software*, 3(22), 547. <https://doi.org/10.21105/joss.00547>

Harris, C.R., Millman, J.K., van der Walt, S.J, Gommers, R., Virtanen, P., Cournapeau, V., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Fernandez del Rio, F., Wiebe, M., Peterson, P., Gerard-Marchant, P., Sheppard, K., … Oliphant, T.E. (2020). Array programming with NumPy*. Nature*, *585*(7825), 357-362.

Larose, C. D., & Larose, D. T. (2019). *Data science using Python and R.* Wiley.

Malarvizhi, R., & Thanamani, A. S. (2012). K-nearest neighbor in missing data imputation. *Int. J. Eng. Res. Dev*, *5*(1), 5-7.

McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

Nehme, A. (n.d.) *Cleaning Data in Python* [Online course]. DataCamp. <https://app.datacamp.com/learn/courses/cleaning-data-in-python>

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., … van Mulbregt,P. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272.

Yamada, T., Romer, M. M., & Richards, D. S. P. (2015). Kurtosis tests for multivariate normality with monotone incomplete data. *Test*, *24*(3), 532-557.