# **D207 Exploratory Data Analysis Performance Assessment**

Jason Pena

WGU University

**Data file being used:**

*Churn\_clean.csv*

1. **Describe a real-world organizational situation or issue in the Data Dictionary you chose, by doing the following:**

Which customers are at high risk of churn? Also, which features/variables are most significant to churn?

1. **Explain how stakeholders in the organization could benefit from the analysis of the data.**

“Churn analytics provides valuable capabilities to predict customer churn and also define the underlying reasons that drive it.”(Frohbose, 2020, November 24)

The analysis will provide a clear understanding of which customers are at highest risk of churn. This will provide supporting evidence for decisions that will improve the services to customers with these anomalies that past users have experienced.

1. **Identify all data in your data set that are relevant to answering your question in part 1A.**

After analyzing the cleaned dataset, I have identified that the dependent variable "Churn" is one the most relevant variables in the decision-making process for my analysis. 'Churn' is a binary categorical character string that indicates whether the client has stopped using a provider’s product or service during a certain time frame. ("Yes or No").

I also have discovered relevance in the continuous numerical data columns such as:

* Tenure - The number of months the customer has stayed with the organization.
* MonthlyCharge - The average monthly charge to the customer.
* Bandwidth\_GB\_Year - The average yearly amount of data used. in GB, per customer.

Lastly, in the surveys, customers provided ordinal numerical data by rating 8 customer service factors that are on a scale of 1 to 8 (1 = most important, 8 - Least important)

* Item1: Timely Response
* Item2: Timely Fixes
* Item3: Timely Replacements
* Item4: Reliability
* Item5: Options
* Item6: Respectful Response
* Item7: Courteous Exchange
* Item8: Evidence of Active Listening

1. **Describe the data analysis by doing the following:**

**1**.  Using one of the following techniques, write code (in either Python or R) to run the analysis of the data set:

* chi-square
* t-test
* ANOVA

My analysis will be using the following technique: chi-square testing

“The chi-square test of independence works by comparing the categorically coded data that you have collected (known as the observed frequencies) with the frequencies that you would expect to get in each cell of a table by chance alone (known as the expected frequencies).”(Page 162, Statistics in Plain English, Third Edition, 2010)

“The Chi-Squared test does this for a contingency table, first calculating the expected frequencies for the groups, then determining whether the division of the groups, called the observed frequencies, matches the expected frequencies.”(Brownlee, 2019)

**2. Provide the output and the results of any calculations from the analysis you performed.**

# Standard Data Science Imports

import numpy as np

import pandas as pd

from pandas import DataFrame

#Visualization libraries

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

#Statistics packages

import pylab

import statsmodels.api as sm

import statistics

from scipy import stats

# Import chisquare from scipy.stats

from scipy.stats import chisquare

from scipy.stats import chi2\_contingency

#Load data set into Pandas DataFrame

df = pd.read\_csv(r'C:\Users\Hydraconix\Desktop\DATA\churn\_clean.csv')

#Rename Last survey columns for better description of variables

df.rename(columns = {'Item1':'TimelyResponse',

'Item2':'Fixes',

'Item3':'Replacements',

'Item4':'Reliability',

'Item5':'Options',

'Item6':'Respectfulness',

'Item7':'Courteous',

'Item8':'Listening'},

inplace=True)

#Displaying the frequency distribution for churn

plt.figure(figsize=(5,5))

ax = sns.countplot(x=df['Churn'], palette="Blues", linewidth=1)

plt.show()

Chart, bar chart

Description automatically generated

#Creating Contingency table to compare 2 variables (Churn & Timely Response)

contingency = pd.crosstab(df['Churn'],df['Timely Response'])

contingency

Output:

|  |
| --- |
| *Timely Response 1 2 3 4 5 6 7* |
| *Churn* |
| *No 158 1002 2562 2473 994 146 15* |
| *Yes 66 391 886 885 365 53 4* |

#Creating Heatmap for data Visualization (Churn & Timely Response)

plt.figure(figsize=(18,8))

sns.heatmap(contingency, annot=True, cmap="YlGnBu")

Chart, treemap chart

Description automatically generated

# Chi-Square test of independence for Variable (Churn & Timely Response)

stat, p, dof, expected = chi2\_contingency(contingency)

print('dof=%d' % dof)

print(expected)

Output:

|  |
| --- |
| dof=6 |

#Interpret test-statistic (Churn & Timely Response)

prob=0.95

critical = chi2.ppf(prob, dof)

print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))

if abs(stat) >= critical:

print('Dependent (reject H0)')

else:

print('Independent (fail to reject H0)')

Output:

|  |
| --- |
| probability=0.950, critical=12.592, stat=4.332 |
| Independent (fail to reject H0) |

#Interpret p-value (Churn & Timely Response)

alpha = 1 - prob

print('significance=%.3f, p=%.3f' % (alpha, p))

if p <= alpha:

print('Dependent (reject H0)')

else:

print('Independent (fail to reject H0)')

Output:

|  |
| --- |
| significance=0.050, p=0.632 |
| Independent (fail to reject H0) |

#Creating Contingency table to compare 2 variables (Churn & Reliability)

contingency2 = pd.crosstab(df['Churn'],df['Reliability'])

contingency2

Output:

|  |
| --- |
| Reliability 1 2 3 4 5 6 7 |
| Churn |
| No 162 990 2524 2523 998 145 8 |
| Yes 59 360 906 929 337 58 1 |

#Creating Heatmap (Churn & Reliability)

plt.figure(figsize=(12,8))

sns.heatmap(contingency2, annot=True, cmap="YlGnBu")

Chart, treemap chart

Description automatically generated

# Chi-Square Test: degrees of freedom (Churn & Reliability)

stat, p, dof, expected = chi2\_contingency(contingency2)

print('dof=%d' % dof)

Output:

|  |
| --- |
| dof=6 |

#Interpret test-statistic (Churn & Reliability)

prob=0.95

critical = chi2.ppf(prob, dof)

print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))

if abs(stat) >= critical:

print('Dependent (reject H0)')

else:

print('Independent (fail to reject H0)')

Output:

|  |
| --- |
| probability=0.950, critical=12.592, stat=2.961 |
| Independent (fail to reject H0) |

#Interpret p-value (Churn & Reliability)

alpha = 1 - prob

print('significance=%.3f, p=%.3f' % (alpha, p))

if p <= alpha:

print('Dependent (reject H0)')

else:

print('Independent (fail to reject H0)')

Output:

|  |
| --- |
| significance=0.050, p=0.814 |
| Independent (fail to reject H0) |

#Creating Contingency table to compare 2 variables (Churn & Courteous)

contingency3 = pd.crosstab(df['Churn'],df['Courteous'])

contingency3

Output:

|  |
| --- |
| Courteous 1 2 3 4 5 6 7 |
| Churn |
| No 161 946 2516 2585 967 166 9 |
| Yes 58 363 930 871 368 58 2 |

#Creating Heatmap (Churn & Courteous)

plt.figure(figsize=(12,8))

sns.heatmap(contingency3, annot=True, cmap="YlGnBu")

Chart, treemap chart

Description automatically generated

# Chi-Square Test: degrees of freedom (Churn & Courteous)

stat, p, dof, expected = chi2\_contingency(contingency3)

print('dof=%d' % dof)

Output:

|  |
| --- |
| dof=6 |

#Interpret test-statistic (Churn & Courteous)

prob=0.95

critical = chi2.ppf(prob, dof)

print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))

if abs(stat) >= critical:

print('Dependent (reject H0)')

else:

print('Independent (fail to reject H0)')

Output:

|  |
| --- |
| probability=0.950, critical=12.592, stat=5.638 |
| Independent (fail to reject H0) |

#Interpret p-value (Churn & Courteous)

alpha = 1 - prob

print('significance=%.3f, p=%.3f' % (alpha, p))

if p <= alpha:

print('Dependent (reject H0)')

else:

print('Independent (fail to reject H0)')

Output:

|  |
| --- |
| significance=0.050, p=0.465 |
| Independent (fail to reject H0) |

**3. Justify why you chose this analysis technique.**

The chi-square test uses “statistical tests that determine whether the output variable is dependent or independent on the input variable(s)”(Brownlee, 2019). Since churn is a binary categorical dependent variable, the chi-square testing is a great technique as it is a non-parametric test for this “yes/no” target variable. The other categorical variables (Timely Response, Reliability, and Courteous) are the input variable being tested.

This analysis technique indicates whether the input features are relevant to the outcome that will be predicted.

1. **Identify the distribution of two continuous variables and two categorical variables using univariate statistics from your cleaned and prepared data.**

Two continuous variables from the dataset

|  |
| --- |
| * MonthlyCharge |
| * Bandwidth\_GB\_Year |

Two categorical (ordinal) variables from the dataset

|  |
| --- |
| * Item1 – Timely Response |
| * Item7 - Courteous |

# Create histograms of continous & categorical variables

df[['MonthlyCharge', 'Bandwidth\_GB\_Year', 'Timely Response', 'Courteous']].hist()

plt.savefig('churn\_pyplot.jpg')

plt.tight\_layout()

Output:

Chart, histogram

Description automatically generated

#Create Seaborn boxplots for continous & categorical variables

sns.boxplot('MonthlyCharge', data = df)

plt.show()

Output:

Chart

Description automatically generated

sns.bloxplot('Bandwidth\_GB\_Year', data = df)

plt.show()

Output:

Chart, histogram

Description automatically generated

sns.boxplot('TimelyResponse', data = df)

plt.show()

Output:

Chart, box and whisker chart

Description automatically generated

sns.boxplot('Courteous', data = df)

plt.show()

Output:

Chart, box and whisker chart

Description automatically generated

1. **Identify the distribution of two continuous variables and two categorical variables using bivariate statistics from your cleaned and prepared data.**

Two continuous variables from the dataset

|  |
| --- |
| * MonthlyCharge |
| * Bandwidth\_GB\_Year |

Two categorical (binomial & ordinal) variables from the dataset

|  |
| --- |
| * Churn |
| * Item7 - Courteous |

#Create dataframe for heatmap bivariate analysis of correlation

churn\_bivariate = df[['MonthlyCharge', 'Bandwidth\_GB\_Year', 'TimelyResponse', 'Courteous']]

#Correlation Matrix

churn\_bivariate.corr()

Output:

MonthlyCharge Bandwidth\_GB\_Year Timely Response Courteous

MonthlyCharge 1.000000 0.060406 0.009756 -0.006399

Bandwidth\_GB\_Year 0.060406 1.000000 -0.007314 -0.001077

Timely Response 0.009756 -0.007314 1.000000 0.336782

Courteous -0.006399 -0.001077 0.336782 1.000000

#Heatmaps for bivariate analysis of correlation

sns.heatmap(churn\_bivariate.corr(), annot=True, cmap=’coolwarm’)

plt.show()

Chart, treemap chart

Description automatically generated

# Create a scatter plot of continous variables MontthlyCharge & Bandwidth\_GB\_Year

churn\_bivariate[churn\_bivariate['MonthlyCharge'] < 300] .sample (100).plot.scatter(X='MonthlyCharge',

y='Bandwidth\_GB\_Year')

Chart, scatter chart

Description automatically generated

# Create a scatter plot of categorical variables TimelyResponse & Courteous

churn\_bivariate[churn\_bivariate['TimelyResponse'] < 7].sample(100).plot.scatter(x='TimelyResponse',

y='Courteous')

Chart, scatter chart

Description automatically generated

# Create a hex plot of continuous variables MonthlyCharge & Bandwidth\_GB\_Year

churn\_bivariate[churn\_bivariate['MonthlyCharge'] < 300].plot.hexbin(x='MonthlyCharge', y='Bandwidth\_GB\_Year', gridsize=15)

Background pattern, bubble chart

Description automatically generated

1. **Summarize the implications of your data analysis by doing the following:**
2. **Discuss the results of the hypothesis test.**

With both the statistical outputs being low as well as the p-values being so high for the variables that were input in the Chi-Square Test:

* Churn and Timely Response
* Churn and Reliability
* Churn and Courteous

The analysis has failed to reject the null hypothesis for these significant variables that were analyzed to be relevant. It is unclear with the given dataset whether there is a statistically significant relationship between the survey responses.

The univariate statistics did not show any significant correlation(s) when analyzing the variables:

Continous variables:

* MonthlyCharge
* Bandwidth\_GB\_Year

Two categorical (ordinal) variables:

* Item1 - Timely Response
* Item7 - Courteous

The bivariate statistics also did not show any significant correlation(s) when analyzing variables:

Continous variables:

* MonthlyCharge
* Bandwidth\_GB\_Year

Categorical(binomial & ordinal) variables:

* Churn
* Item7 – Courteous

1. **Limitations of the Analysis**

The analysis has shown that we need to investigate further as well as gathering more insightful data. The analysis has proven with the limited dataset, it limits the ability to gather meaningful insights and actionable information.

1. **Recommended Course of Action**

While the tests show very little correlation between the variables involved in timely action regarding customer satisfaction (Timely Response, Reliability, and Courteous). The recommended course of action should be to focus on these features, and this could reduce the churn rate which is at %26.5. The emphasis on these features can increase the retention of customers by targeting more resources in Customer Service.

1. **Video**

[**https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=bfb314ae-ecfd-4657-ae77-adbd0068f352**](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=bfb314ae-ecfd-4657-ae77-adbd0068f352)

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

Kaggle. (2018, May 01). Bivariate plotting with pandas. Kaggle. <https://www.kaggle.com/residentmario/bivariate-plotting-with-pandas>

*Matplotlib.axes.axes.hexbin¶*. matplotlib.axes.Axes.hexbin - Matplotlib 3.4.3 documentation. (n.d.). Retrieved October 10, 2021, from <https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.hexbin.html#matplotlib.axes.Axes.hexbin>

Seaborn.jointplot¶. seaborn.jointplot - seaborn 0.11.2 documentation. (2012). Retrieved October 10, 2021, from <https://seaborn.pydata.org/generated/seaborn.jointplot.html>

Sree. (2020, October 27). *Predict customer churn in Python*. Medium. Retrieved October 10, 2021, from <https://towardsdatascience.com/predict-customer-churn-in-python-e8cd6d3aaa7>

1. **References**

Brownlee, J. (2019, October 30). *A gentle introduction to the chi-squared test for machine learning*. Machine Learning Mastery. Retrieved October 3, 2021, from <https://machinelearningmastery.com/chi-squared-test-for-machine-learning/>

Frohbose, F.   (2020, November 24).   Machine Learning Case Study: Telco Customer Churn Prediction. Towards Data Science. <https://towardsdatascience.com/machine-learning-case-study-telco-customer-churn-prediction-bc4be03c9e1d>

Urdan, T. C. (2010). Statistics in Plain English. Routledge.