**Joseph Deal**

**Student ID: 000655053**

**Western Governors University**

**Multiple Regression**

**For Predictive Modeling**

**Part I:**

**RESEARCH QUESTION**

**A.**

**1.** Utilize the cable company data set to determine which services predict tenure of customers?

**2.** The goal of this analysis is to identify critical services that are used by customers that keep them coming back. Identifying these services would provide a basis for marketing emphasis to current customers as well as marketing for new customers.

**Part II:**

**METHOD JUSTIFICATION**

**B.**

**1.** Multiple regression analysis is dependent on a few assumptions regarding the dataset. The most obvious assumption is that the relationship between the target and predictor variables is linear.

A second assumption is that the residuals between the observed and predicted values should be normally distributed.

A third assumption is that there is no multicollinearity. The independent variables cannot be highly correlated, if so, the highly correlated variables should be removed from the analysis.

The last assumption is homoscedasticity. Homoscedasticity is focused on the variance of the independent variables. That variance needs to be finite or very similar over all the variables. (Assumptions of Multiple Linear Regression, n.d.)

**2.** Using Python as the primary programming tool for analysis provides significant opportunities for utilization, sharing, and frameworks. Many development environments can utilize Python from the most robust to an online text editor. Jupyter notebooks are an invaluable tool in which python programming can be used and shared readily. Google Colabs (colabs.research.google.com) is another environment that is accessible from anywhere. Almost any device can access Colabs if it has internet service.

According to a kite.com article, Python is adopted by data scientists for these reasons:

* Open source and an active Community
* Intuitive Syntax
* An extensive collection of standardized libraries (a few mentioned below)
* Integration with fast, compiled languages for numerical computation primitives (Numpy, Pandas)
* Ease of integrating modeling with database access, wrangling post-processing, and visualization and web-serving
* Availability and continued development of Pythonic interfaces to Big Data frameworks such as Apache Spark or MongoDB
* Support and development of python libraries by large and influential organizations such as Google or Facebook (e.g. TensorFlow and PyTorch)(Sarkar, 2019)

Python has numerous packages that can be added to its environment to improve performance and add algorithms to enhance data analysis. Some of those tools include Numpy, Pandas, Scikit-Learn, Seaborn, Matplotlib, Statsmodel, Pyspark, Spark, and many more.

**3.** Multiple Linear Regression analysis for the telecom churn data could be very beneficial in many ways. Analysis the tenure data will provide insight into potential features that are indicative of longer tenures and those that are present or absent for tenures that are short. The dataset has many features that is relative to the various services provided by the company. Utilizing multiple linear regression analysis will allow to mathematically evaluate each of those many features and identify which ones are the most significant for making decisions going forward.

**Part III:**

**DATA PREPARATION**

**C.**

**1.** Data Preparation will consist of several steps. Focusing on values that cause programming errors include; missing values like, Nulls, NaNs, and any strangely formatted data. There is a multitude of ways to deal with these types of issues. Replacing Nulls and NaNs for a numeric field can be as simple as replacing with zeros or as complex as calculating the values' average based on specific other characteristics of the dataset.

Checking for data types found in each feature is critical in analysis performance as well. Verifying data types will prevent another set of issues that programming and analysis will not interpret correctly. For instance, a feature may appear to be a currency field. The field includes the dollar sign, and the data type for the variable may be an object instead of a number.

Another data prep issue is determining if the numeric field is continuous or discrete or if the character fields are categorical or free text. The understanding of the structure of the variable helps in exploring extremes and rectifying any issues found.

Categorical fields will need enumeration for calculation and analytic purposes. Identifying all the categorical fields and the correctness of each category value will be critical.

Deletion of records with missing information may occur if there are plenty of other records in the dataset.

Duplicate records absolutely need deletion. There were no duplicate records found.

Checking for extremes in any of the column types is an excellent way to evaluate continuity. Many extreme entries are errors and need to be addressed. These extreme values could significantly impact the data analysis, particularly true if the dataset is small. Handling these extreme values is typically a logic check. If the data clearly makes no sense or is impossible, then a decision on how to handle the issues come to the forefront. Do you delete the record? Do you average out the value if it is a number? Do you make it zero or the mode value? There are lots of options.

With categorical variables, it is requisite to check for different spellings. The various spellings and misspellings could dilute the potential value and impact a certain category may have in the analysis.

For analysis purposes, text-based classification variables will need to be enumerated. Encoding the categorical fields with pandas.get\_dummies (pandas.get\_dummies, n.d.) on the following fields, PaymentMethod, Contract, Marital, InternetService, Port\_modem, Phone, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, and StreamingMovies.

**2.** A quick look through the table finds that numerous fields will not be part of the general analysis. These include Lat, Lng, UID, Interaction, Customer\_id, CaseOrder, TimeZone, Job, Zip, County, State, City, and Area. These variables show no variation or pattern in distribution graphs and identify all customers to be in North America.

A quick thought on the locations and their relevance is visible in the zip field. The zip encompasses all the state, city, area, county, lat and lng, and TimeZone. The zip field distribution in the below chart shows even distribution of customers from the 00xxx's to the 999xx's. This distribution indicates that no one area is suffering from churn anymore than any other. The slight dip in the 00xxx's and the 88xxxs ranges reflect the lack of population in the northern New England area and the Rocky Mountain states inclusively.

|  |
| --- |
|  |
| Chart 1 – Zip Code Distribution |

There are 10,000 records for every field. There are no records that have blanks, NaNs, or Null values.

Looking for extreme values are easily seen in the following table showing field name, min, max, average, and quartiles.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Age** | **Income** | **Tenure** | **MonthlyCharge** | **GB/Year** | **Contacts** | **EquipFail** | **Items** |
| **Mean** | 53.07 | 39806.92 | 34.53 | 172.62 | 3392.34 | 0.99 | 0.39 | 3.5 |
| **std** | 20.69 | 28199.91 | 26.44 | 42.94 | 2185.29 | 0.98 | 0.63 | 1.0 |
| **min** | 18 | 348.67 | 1.00 | 79.97 | 155.5 | 0.00 | 0.00 | 1.00 |
| **25%** | 35 | 19224.71 | 7.92 | 139.97 | 1236.47 | 0.00 | 0.00 | 3.00 |
| **50%** | 53 | 33170.61 | 35.43 | 167.48 | 3279.53 | 1.00 | 0.00 | 4.00 |
| **75%** | 71 | 53246.17 | 61.48 | 200.73 | 5586.14 | 2.00 | 1.00 | 4.00 |
| **max** | 89 | 258900.70 | 71.99 | 290.16 | 7158.98 | 7.00 | 6.00 | 7.00 |

**Table 1: df.describe() results**

In the table, the Age range goes from 18 to 89, which seems very reasonable. 75% of the users are under the age of 71. Income ranges from $348 to $258,900. The low end seems odd, but it could be college-age customers who may not earn a great deal and are subsidized by parents.

The Tenure field seems reasonable as well, with extremes of 1 to 72 months. Monthly charges from $79.97 to $290.16 appear to be a reasonable spread. Contacts align nicely with equipment fails (EquipFail) from min to max with an additional contact than an equipment failure.

The target field is Tenure. This field the number of months a customer has paid for the telecom’s services. It reflects that there is a large turnover of customers in the last 40 months. The bulk of new customers have been with the company for less than 12 months. There is another bulk of the customer base that have been with the company for 4.5 years or more, as noted in Chart 2.

|  |
| --- |
|  |
| Chart 2 – Tenure Distribution |

It is observed that Marital, Gender, InternetService, PaymentMethod, Item1 through Item8, OnlineBackup, OnlineSecurity, Multiple, Phone, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, paperlessBilling, TimeZone, Area, County, Zip, State, City, Churn are all categorical fields. Most are binary such as Churn, OnlineBackup, OnlineSecurity, Multiple, Phone, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, paperlessBilling. Simultaneously, the balance of categorical fields has multiple nominal values except for the Item1 thru Item8 fields, which are ordinal values received from a survey given after each customer interaction.

The customer interaction survey results appear to be very consistent. See the charts for items1 through item8 below. The consistency from one question to the next is not very informative.

|  |
| --- |
|  |
| Chart 3 – Survey Question Distributions |

All the categorical variables have a consistent and reasonable selection of values. Marital status has 'Widowed', 'Married', 'Separated', 'Never Married', 'Divorced'. PaymentMethods variable has 'Electronic Check', 'Mailed Check', 'Bank Transfer(automatic)', 'Credit Card(automatic)' ared standard values. Gender has 'Male', 'Female', 'Nonbinary'. InternetService has 'Fiber Optic', 'DSL', and 'None'. Contracts have three values of 'Month-to-month', 'Two Year', 'One Year'.

**3.** After loading the data into a pandas dataframe in Python, the steps used to prepare the data for analysis include the following steps:

* A quick review of the data to evaluate what the data looks like using the following code:
  + df.info – which displays the top 5 rows and the bottom five rows of the dataset loaded into the dataframe df.
* A quick statistical overview of the dataframe is produced using:
  + Df.describe()
    - This command provides a count, mean, standard deviation, minimum, 1st quartile, 2nd quartile, 3rd quartile, and max values for all the numerical variables in the dataframe, as noted in Table 1 above.
    - A review of these results should expose any extreme values should they exist.
* Review all the column names.
  + Df.columns will present the list of column names in the dataframe.
* Review all the variable data types.
  + Df.dtypes will return a list of all the variable and the data types associated with each, such as int64, object, float64.
  + A quick review here will identify the numeric fields from the text fields. The object fields will be considered categorical and addressed by encoding methods if appropriate. If you have 1,000 different categories, it would not be appropriate.
* Review general contents of all variables.
  + Using df.info() will display a count of each variable, the variable name, the number of Non-Null values, and the data type of the field.
  + Info() is a quick and easy way to find missing values and locate them in the dataframe.
* Review for missing information
  + Using df.isna().any() will review each variable and check for NaNs, Nulls, NaTs for each variable. It returns a list of variables with a True if there are missing values or False if there are no missing values.
* Review all the object variables and values
  + Df[‘Marital’].unique() will return all the unique values in the variable.
  + Reviewing those unique values will spot any misspellings or extreme values for the variables.
  + Used .unique() on all categorical variables such as Marital, PaymentMethod, InternetService, Contract, Gender, TechSupport, DeviceProtection, OnlineBackup, OnlineSecurity, Multiple, Port\_modem, Tablet, Phone, PaperlessBilling, StreamingTV, StreamingMovies, and Techie.
  + There were no extreme or irrational values in these variables.

**4.** Univariate visualizations:

|  |
| --- |
|  |
| Chart 4 – Churn Percentages |

|  |
| --- |
|  |
| Chart 5 – Distribution of Customers by Tenure |
|  |

|  |
| --- |
|  |
| Chart 6 – Internet Service Distribution |
|  |

Bivariate visualizations:

|  |
| --- |
|  |
| Chart 7 – Frequency of Churn by Tenure |
|  |
|  |
| Chart 8 – Churn Frequency by Monthly Charge |
|  |
| Chart 9 – Monthly Charge by Internet Service |

**5.** A copy of the cleaned dataset is in the file named prepared\_multipleregression\_churn\_data.csv.

**Part IV:**

**MODEL COMPARISON AND ANALYSIS**

**D.**

**MULTIPLE LINEAR REGRESSION WITH ALL PREDICTORS**

1. The following code provided the structure and output for the first model.

(Comments are bold. Code is plain text.)

**# Loaded the proper modules from sklearn**

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

**# Setup the X and Y datasets for splitting**

X\_features = df.loc[:,df.columns != ‘MonthlyCharge’]

y\_features = df[‘MonthlyCharge]

**# Split the dataset into a training and testing set. Using an 80/20 testing/training split.**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_features, y\_features, test\_size=0.25, random\_state=234)

reg = LinearRegression()

**# Run the predict method with the test data to evaluate the logistic model with all the predictors.**

regmodel = reg.fit(X\_train, y\_train)

**# Run the score method to calculate the precision of the logistic model.**

Regmodel = regmodel.score(X\_test,y\_test)

**# Results of the scoring was 0.999. An 99.99% prediction rate for monthly charge is overfit and bogus.**

y\_prediction = regmodel.predict(X\_test)

result = pd.DataFrame({‘Actual’: y-test, ‘Predicted’: y\_prediciton})

print(result)

**# The results are below:**

Actual Predicted

2464 11.496870 11.386171

8137 65.133710 65.235689

4388 20.883240 20.987621

7968 54.237610 54.120833

7526 62.999280 63.114627

... ... ...

1756 7.712199 7.809636

2269 7.803130 7.906220

9085 64.713890 64.819239

3553 14.744150 14.857014

2435 20.221910 20.332673

[2500 rows x 2 columns]

**# The results appear to be rather close between the Actual and Predicted Values, which is not bad for a continuous variable like MonthlyCharge.**

Coefficients = regmodel.coef\_

Intercept = regmodel.intercept\_

**Intercept = -3.849010238786171**

**Coefficients =**

array([-3.75869018e-01, 3.99287852e-02, 3.22921782e-08, 3.25366180e-03,

2.70848624e-04, -2.05492876e-04, -2.93290725e-04, -1.83367372e-03,

-2.18635273e-03, 3.89541788e-03, -4.98454394e-04, 4.85681104e-03,

2.64198185e-01, -8.31750648e-01, -3.55951577e-01, -5.99460019e-01,

3.81297086e-01, -1.30502688e+00, -7.30056412e-01, -4.43298489e-03,

-3.50823492e-02, 1.22048358e-02, 1.17538630e-03, -1.06390161e-03,

1.82145874e-03, -3.95304313e-05, -1.02281248e-03, 7.26025017e-04,

-1.87397418e-03, -1.82104604e-04, -2.24371619e-03, -1.13807054e-03,

1.48981697e-04, 2.38060651e-03, 5.75225221e+00, 4.60276859e+00,

-7.94074694e-01, 2.60810079e-01, -2.73954392e-03, 2.44763434e-03,

6.42745892e-03, 9.60570952e-04, 1.13784044e-03])

**# Further tests for accuracy and believability**

print("Mean Squared Error (MSE): %.2f"% np.mean((regmodel.predict(X\_test)-y\_test) \*\*2))

Mean Squared Error (MSE) = 0.01

print("Variance score: %.2f" % regmodel.score(X\_test,y\_test))

Variance score: 1.00

1. There were 43 features evaluated during the multiple linear regression. Not all those predictors were helpful. Determining which features to focus on and which to ignore will require more information. Utilizing a correlation heatmap is useful, but it is difficult to read due to the number of features. Please refer to the chart below.

|  |
| --- |
|  |
| Chart 11 – Correlation Heat Map |

The target of the regression analysis is the ‘MonthlyCharge’ variable. As expected, there are noted correlations to ‘MonthlyCharge’ with ‘Multiple’, ‘OnlineSecurity’, ‘OnlineBackup’, ‘DeviceProtection’, ‘TechSupport’, ‘StreamingTV’, ‘StreamingMovies’, ‘Internet\_Fiber Optic’, and ‘Churn’. All of these features are actually services that require an additional fee for the subscriber, except for Churn. The relationship between ‘MonthlyCharge’ and Churn is of a different nature.

Another python statistic module called statsmodel has a linear regression method called OLS that has a robust statistical analysis included. To verify the results using the scikitLearn LinearRegression model, I installed statsmodels.api and ran statsmodels OLS function on the same dataset. The summary results are noted below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | OLS Regression Results | | | | | **Dep. Variable:** | MonthlyCharge | **R-squared (uncentered):** | 0.999 | | **Model:** | OLS | **Adj. R-squared (uncentered):** | 0.999 | | **Method:** | Least Squares | **F-statistic:** | 2.524e+05 | | **Date:** | Thu, 28 Jan 2021 | **Prob (F-statistic):** | 0.00 | | **Time:** | 13:35:14 | **Log-Likelihood:** | -22184. | | **No. Observations:** | 7500 | **AIC:** | 4.445e+04 | | **Df Residuals:** | 7457 | **BIC:** | 4.475e+04 | | **Df Model:** | 43 |  |  | | **Covariance Type:** | nonrobust |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | **coef** | **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** | | **Children** | -5.9163 | 0.042 | -142.112 | 0.000 | -5.998 | -5.835 | | **Age** | 0.6031 | 0.004 | 157.931 | 0.000 | 0.596 | 0.611 | | **Income** | -7.112e-06 | 1.89e-06 | -3.764 | 0.000 | -1.08e-05 | -3.41e-06 | | **Churn** | 1.0930 | 0.172 | 6.372 | 0.000 | 0.757 | 1.429 | | **Outage\_sec\_perweek** | -0.1969 | 0.018 | -10.908 | 0.000 | -0.232 | -0.162 | | **Email** | -0.1956 | 0.018 | -10.990 | 0.000 | -0.231 | -0.161 | | **Contacts** | -0.1809 | 0.055 | -3.315 | 0.001 | -0.288 | -0.074 | | **Yearly\_equip\_failure** | -0.2062 | 0.085 | -2.420 | 0.016 | -0.373 | -0.039 | | **Techie** | -0.0820 | 0.147 | -0.559 | 0.576 | -0.370 | 0.206 | | **Port\_modem** | -0.3401 | 0.108 | -3.145 | 0.002 | -0.552 | -0.128 | | **Tablet** | -0.2984 | 0.118 | -2.529 | 0.011 | -0.530 | -0.067 | | **Phone** | -1.8061 | 0.184 | -9.822 | 0.000 | -2.167 | -1.446 | | **Multiple** | 18.4431 | 0.135 | 136.359 | 0.000 | 18.178 | 18.708 | | **OnlineSecurity** | -12.0524 | 0.141 | -85.599 | 0.000 | -12.328 | -11.776 | | **OnlineBackup** | 4.3525 | 0.148 | 29.379 | 0.000 | 4.062 | 4.643 | | **DeviceProtection** | -3.9128 | 0.143 | -27.339 | 0.000 | -4.193 | -3.632 | | **TechSupport** | 11.4062 | 0.112 | 101.604 | 0.000 | 11.186 | 11.626 | | **StreamingTV** | -1.6223 | 0.260 | -6.245 | 0.000 | -2.131 | -1.113 | | **StreamingMovies** | 11.8156 | 0.243 | 48.609 | 0.000 | 11.339 | 12.292 | | **PaperlessBilling** | -0.2980 | 0.110 | -2.711 | 0.007 | -0.514 | -0.083 | | **Tenure** | -15.5542 | 0.083 | -186.477 | 0.000 | -15.718 | -15.391 | | **Bandwidth\_GB\_Year** | 0.1898 | 0.001 | 187.422 | 0.000 | 0.188 | 0.192 | | **Item1** | -0.0407 | 0.078 | -0.521 | 0.602 | -0.194 | 0.112 | | **Item2** | 0.0118 | 0.072 | 0.162 | 0.871 | -0.130 | 0.154 | | **Item3** | -0.1752 | 0.066 | -2.637 | 0.008 | -0.305 | -0.045 | | **Item4** | -0.7546 | 0.058 | -12.976 | 0.000 | -0.869 | -0.641 | | **Item5** | -1.1077 | 0.059 | -18.701 | 0.000 | -1.224 | -0.992 | | **Item6** | -0.3944 | 0.063 | -6.229 | 0.000 | -0.519 | -0.270 | | **Item7** | -0.3803 | 0.060 | -6.368 | 0.000 | -0.497 | -0.263 | | **Item8** | -0.3246 | 0.057 | -5.672 | 0.000 | -0.437 | -0.212 | | **Marital\_Married** | -0.6620 | 0.171 | -3.875 | 0.000 | -0.997 | -0.327 | | **Marital\_Never Married** | -0.7377 | 0.170 | -4.341 | 0.000 | -1.071 | -0.405 | | **Marital\_Separated** | -0.6295 | 0.170 | -3.712 | 0.000 | -0.962 | -0.297 | | **Marital\_Widowed** | -0.8309 | 0.168 | -4.937 | 0.000 | -1.161 | -0.501 | | **Internet\_Fiber Optic** | 98.1420 | 0.424 | 231.212 | 0.000 | 97.310 | 98.974 | | **Internet\_None** | 65.4008 | 0.434 | 150.687 | 0.000 | 64.550 | 66.252 | | **Gender\_Male** | -12.8738 | 0.132 | -97.437 | 0.000 | -13.133 | -12.615 | | **Gender\_Nonbinary** | 3.3518 | 0.361 | 9.273 | 0.000 | 2.643 | 4.060 | | **PaymentMethod\_Credit Card (automatic)** | -0.7827 | 0.165 | -4.753 | 0.000 | -1.106 | -0.460 | | **PaymentMethod\_Electronic Check** | -0.6273 | 0.147 | -4.272 | 0.000 | -0.915 | -0.339 | | **PaymentMethod\_Mailed Check** | -0.5899 | 0.161 | -3.673 | 0.000 | -0.905 | -0.275 | | **Contract\_One year** | 0.1237 | 0.145 | 0.851 | 0.395 | -0.161 | 0.409 | | **Contract\_Two Year** | 0.1569 | 0.138 | 1.138 | 0.255 | -0.113 | 0.427 |  |  |  |  |  | | --- | --- | --- | --- | | **Omnibus:** | 1429.946 | **Durbin-Watson:** | 1.968 | | **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 293.356 | | **Skew:** | 0.037 | **Prob(JB):** | 1.99e-64 | | **Kurtosis:** | 2.034 | **Cond. No.** | 6.48e+05 |   Warnings: [1] Standard Errors assume that the covariance matrix of the errors is correctly specified. [2] The condition number is large, 6.48e+05. This might indicate that there are strong multicollinearity or other numerical problems. |
| Chart 12 – Statsmodel OLS Summary Data |

Interpretation of this chart focuses on the fourth column of data or the p-value. Any feature with a p-value smaller than 0.05 is a viable candidate for selection as a feature for further investigation.

Upon review, Techie, Item1, Item2, Contract\_One year, and Contract\_Two year do not fall below the 0.05 threshold.

Nothing in this analysis lines up with the correlation heatmap above. This issue could be caused by several things one being that there is no y-intercept. The y-intercept for MonthlyCharge would be zero and there are not bills at zero. The lack of a y-intercept at times confuses the linear regression models. The high 99.999% prediction rate reflects that confusion as well. Using all the predictors is generally not a good idea for a linear regression model.

1. The Reduced Predictor Multiple Linear Regression analysis produced more realistic numbers. However, finding the right predictor variables to focus on was two fold. One is use the ones that correlate with the ‘MonthlyCharge’ target or use another technique to identify the most impactful variables.

There is another regression technique calle the Lasso Regression. The Lasso Regression adjusts out variables that variances can be minimized and those variables with large or inconsequential variances are turned to zero and have no impact. (Zach, 2020)

Running the Lasso regression with the following code;

from sklearn.linear\_model import Lasso

lassoreg = Lasso(alpha=0.01, normalize = True)

lassoreg.fit(X\_train, y\_train)

**# PRINT OUT THE RESULTING COEFFICIENTS**

print(lassoreg.coef\_)

[ 0. 0. 0. 1.66109624 -0.

0. -0. -0. 0. -0.

-0. -0. 30.64819113 0.83895828 20.69201036

10.8748363 10.64463426 40.13439908 50.00559052 0.

0. 0. 0. 0. -0.

0. 0. -0. -0. 0.

0. -0. 0. -0. 18.72733752

-11.44378493 -0. -0. -0. 0.

0. 0. 0. ]

**# TEST THE LASSO REGRESSION**

y\_pred = lassoreg.predict(X\_test)

**# Print out the test scores using;**

from sklearn import metrics

print("R2 (R-Square Value)",r2\_score(y\_test,y\_pred))

print("\n")

print ("MAE (mean\_absolute\_error) :",metrics.mean\_absolute\_error(y\_test, y\_pred))

print("\n")

print ("MSE (mean\_squared\_error) : ",metrics.mean\_squared\_error(y\_test, y\_pred))

print("\n")

print ("RMSE (root\_mean\_squared\_error) : ",np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

**# RESULTS:**

R2 (R-Square Value) 0.9554539407908741

MAE (mean\_absolute\_error) : 8.73259057926937

MSE (mean\_squared\_error) : 83.69730606806327

RMSE (root\_mean\_squared\_error) : 9.148623178821131

These numbers are not great, but seem to be more reasonable than with the ‘all preditor’ linear regression model.

Also, diving into the variables that the Lasso Regression identified as impactful include the following; Churn, Multiple, Online Security, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Internet\_FiberOptic, and Internet\_None.

All these fields tightly coincide with the correlation heatmap.

**E. Data Analysis**

**1.** The multiple linear regression analysis is targeting the MonthlyCharge value. The analysis will identify the required variables or predictor features that are necessary to predict monthly billing rates for a customer. It should be easy to predict the monthly charge for a customer dependent on the different services that the customer has decided to utilize. The obvious services revolve around Bandwidth\_GB\_Year if there is a cost per GB or package deal on the GB used, Streaming TV, Streaming Movie services, Online Security, Online Backup, DeviceProtection, and TechSupport are other prime services that might carry a charge for a customer.

To test out that theory, using all the predictors is a general place to start. Evaluating the results and trimming down the features or predictors that do not make an impact. To assess that impact, we utilized the summary section of the statsmodel's OLS function. Each predictor is analyzed in the summary section and given a series of statistics. The most important for this evaluation is the p-value noted as P>|z|.

Focus is on the p-values that are 0.05 and below. This process was of no use. It only reduced the number of predictors from 43 to 38, which was not helpful. The resulting performance parameters of Mean Squared Error(MSE) and Variance score were 0.01 and 1.00 inclusively. These results are nonsense. The use of the Lasso Regression technique provided a much better response. The Lasso Regression identified ten predictors that were vital in predicting the monthly charge. The lasso regression listed the following variable with their coefficients;

|  |  |
| --- | --- |
| Predictor | Coefficient |
| Churn | 1.66109 |
| Multiple | 30.648 |
| OnlineSecurity | 0.8389 |
| OnlineBackup | 20.692 |
| DeviceProtection | 10.874 |
| TechSupport | 10.644 |
| StreamingTV | 40.134 |
| StreamingMovies | 50.005 |
| Internet\_Fiber Optic | 18.727 |
| Internet\_None | -11.444 |

There are a couple of predictors that were unexpected that being Churn and Multiple. Churn has a coefficient of 1.66, which is lower than most other predictors, but there appears to be some relationship with the monthly charge. The ‘Multiple’ predictor must designate multiple users or feeds that a customer is charged. Predictably, the Internet\_None has a negative coefficient while all the remaining predictors are positive. The coefficient selection makes sense in the commercial sense that these added features would cost additional, thus the positive coefficients for the predictors, while the Internet\_None is lacking a feature and would represent no fee associated for the customer.

The reduced predictor set provides a more normal scoring

R2 (R-Square Value) 0.9554539407908741

MAE (mean\_absolute\_error) : 8.73259057926937

MSE (mean\_squared\_error) : 83.69730606806327

RMSE (root\_mean\_squared\_error) : 9.148623178821131

The R-Square Value of 0.955 seems to identify approximately 95% of the variability in the values in these predictor values.

A chart of the predicted Y values to actual X values for Multiple.

|  |
| --- |
|  |
| Chart 13 – Y and Fitted vs Multiple |

The fitted Y values to actual values captures most values except the most expensive.

Evaluating the residuals shows a relatively narrow range with no discernible pattern deviating from the median. This reflects that there is no trending in the data from one value of Multiple to the next.

|  |
| --- |
|  |
| Chart 14 – Residuals versus Multiple |

|  |
| --- |
|  |
| Chart 15 – Residuals |

The plotting of the residuals provided an exceptional result. It appears that three distinct patterns present themselves. Further investigation would indicate different predictor or groups of predictors driving each of these trends.

Since most of the predictors for the target variable monthlycharge are categorical, the reduction in the number of predictors should reduce the R2 value to less predictive. Reducing the number of predictors from 43 to 10 only reduced the R squared value by 1 percent.

**2.** The reduced predictor analysis produced decent results compared to the all predictor analysis. The OLS Regression stats are presented below in Chart 16 reflect p values for each predictor to be zero and an R-Squared value representing 91.7% of the variance. Referring to Chart 16, Note 2 states there is potentially an issue with multicollinearity within this model. However, the indicators of multicollinearity problems are much less than the full set of predictors. Please refer to note 2 in Chart 12.

|  |
| --- |
|  |
| Chart 16 – Reduced Predictors OLS Summary Statistics |

Calculating the performance parameters for the multiple regressions of the reduced parameters changed from the fantastic 99.9% for all predictors to just 95.5% for the reduced predictors.

|  |
| --- |
|  |
| Chart 16 – Reduced Predictors Performance / Error Calculations |

**3.** Code See attached file: Churn\_Analysis\_Multiple\_Regression Code.pdf

**Part V:**

**DATA SUMMARY AND IMPLICATIONS**

**F. Summarize Findings**

**1.Data Analysis**

**Regression Equation:**

y = 0(y\_Intercept) + 1.66(Churn) + 30.65(Multiple) + 0.83(OnlineSecurity) + 20.69(OnlineBackup + 10.87(DeviceProtection) + 10.64(TechSupport)+40.13(StreamingTV) + 50.01(StreamingMovies) +18.73(Internet\_Fiber Optic) – 11.44(Internet\_None)

**Interpretation of Coefficients**

The coefficients are a multiplier of the predictor positively or negatively. The coefficient will reflect a positive value for a predictor that increases the monthly charge regarding the current question. In contrast, a negative value is a predictor that favors reducing the monthly amount.

The positive predictors for monthly charges are Churn, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, and Internet\_Fiber Optic, while Internet\_None are reducing forces for monthly charge.

The largest impact predictors are Multiple, StreamingTV, StreamingMovies, and Internet\_Fiber Optic. These predictors account for nearly 90% of the estimated impact.

**Limitations of Data Analysis**

As with any data analysis, it depends on the dataset is representative of the whole population. The data is clean and free from duplicates and derived values. However, it is just 100,000 records of a larger dataset, reflective of a limited time-frame.

Also, there are portions of the monthly charges that are not present in the dataset. Possibly taxes and other fees are part of the monthly charge and are not reflected as predictors of the total.

Doing a Multiple Linear Regression analysis to predict monthly charges seems more complicated than necessary. The regression metrics of R-squared appear to be excellent. However, at least one assumption regarding multiple linear regressions does not hold. The assumption is that the residuals are normally distributed. Chart 15 above shows non-normal distributions of those residuals. Another issue is determining whether the relationship between MonthlyCharges and the predictor values are linear. Most are logistic, since the components of the MonthlyCharges are either on or off.

**2. Recommendations**

Using the multiple linear regression for predicting future monthly charges is not recommended. The analysis provides an estimate of the cost of each feature used by the customer. Each of those features has a set known cost and should be readily accessible in a database via SQL queries.

Other possible regression analysis may focus on which predictors are most likely indicators of churn, or which predictors are most predictive in upgrading services. Other questions could be investigated productively with multiple regressions in this dataset.

**REFERENCES**

*Assumptions of Multiple Linear Regression*. (n.d.). Https://Www.Statisticssolutions.Com. Retrieved January 27, 2021, from https://www.statisticssolutions.com/assumptions-of-multiple-linear-regression/

*pandas.get\_dummies*. (n.d.). Pandas.Pydata.Org. Retrieved January 7, 2021, from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get\_dummies.html

*sklearn.model\_selection.train\_test\_split*. (n.d.). Scikit-Learn.Org. Retrieved January 8, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.train\_test\_split.html

Spiegelhalter, D. S. (2019). *The Art of Statistics: How to Learn from Data* (1st ed.). Basic Books.

Zach. (2020, November 12). *Why Use Lasso Regression*. Https://Www.Statology.Org. https://www.statology.org/lasso-regression/