Project Purpose

This project uses the data that was previously cleaned in the Jupyter Notebook "Data Cleaning". The majority of the data set is used to determine how tenure might be predicted. I plan to pare down the list of variables not by what might intuitively belong together, but by looking at numerical calculations that suggest which variables are significant to answering the research question. Armed with these variables, I hope to obtain a strong predictive model to assist the telecommunications company with decisions on how to increase tenure in its customer base.

Steps to Prepare the Data

The following steps will be taken to prepare the data for analysis:

1. Import data to pandas dataframe
2. Determine variable types and those which may require further investigation
3. Convert binary variables to yes = 1 and no = 0
4. Investigate potential categorical variables using bar charts, then convert categorical values to dummy variables
5. Drop columns that will not be used in regression analysis

Step 1

In [2]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle
import seaborn as sns
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.model_selection import cross_val_predict, train_test_split
from sklearn.linear_model import Lasso, LassoCV, Ridge, RidgeCV
import statsmodels.api as sm
import statsmodels.formula.api as smf
from yellowbrick.regressor import AlphaSelection, PredictionError, ResidualsPlot
import warnings
warnings.filterwarnings('ignore') # Ignore warning messages for readability
```

In [21]:
```
# Read in dataset and view head
df = pd.read_csv('churn_clean_data.csv', dtype={'Zip Code': 'str'})
pd.options.display.max_columns = None
df.head()
```

Out[21]:

	CustomerID	Count	Country	State	City	Zip Code	Lat Long	Latitude	Longitude	Gender	Senior Citizen	Partner	Dependents	M
0	3668-QPYBK	1	United States	California	Los Angeles	90003	33.964131, -118.272783	33.964131	-118.272783	Male	No	No	No	
1	9237-HQITU	1	United States	California	Los Angeles	90005	34.059281, -118.30742	34.059281	-118.307420	Female	No	No	Yes	
2	9305-CDSKC	1	United States	California	Los Angeles	90006	34.048013, -118.293953	34.048013	-118.293953	Female	No	No	Yes	
3	7892-POOKP	1	United States	California	Los Angeles	90010	34.062125, -118.315709	34.062125	-118.315709	Female	No	Yes	Yes	
4	0280-XJGEX	1	United States	California	Los Angeles	90015	34.039224, -118.266293	34.039224	-118.266293	Male	No	No	Yes	

Step 2

In [22]:
```
# View list of columns, data types, and missing values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   CustomerID         7043 non-null   object 
 1   Count              7043 non-null   int64  
 2   Country            7043 non-null   object 
 3   State              7043 non-null   object 
 4   City               7043 non-null   object 
 5   Zip Code           7043 non-null   object 
 6   Lat Long           7043 non-null   object 
 7   Latitude           7043 non-null   float64
 8   Longitude          7043 non-null   float64
 9   Gender             7043 non-null   object 
 10  Senior Citizen     7043 non-null   object 
 11  Partner            7043 non-null   object 
 12  Dependents         7043 non-null   object 
 13  Tenure Months      7043 non-null   float64
 14  Phone Service      7043 non-null   object 
 15  Multiple Lines     7043 non-null   object 
 16  Internet Service   7043 non-null   object 
 17  Online Security    7043 non-null   object 
 18  Online Backup      7043 non-null   object 
 19  Device Protection  7043 non-null   object 
 20  Tech Support       7043 non-null   object 
 21  Streaming TV       7043 non-null   object 
 22  Streaming Movies   7043 non-null   object 
 23  Contract           7043 non-null   object 
 24  Paperless Billing  7043 non-null   object 
 25  Payment Method     7043 non-null   object 
 26  Monthly Charges    7043 non-null   float64
 27  Total Charges      7043 non-null   float64
 28  Churn Label        7043 non-null   object 
 29  Churn Value        7043 non-null   float64
 30  Churn Score        7043 non-null   int64  
 31  CLTV               7043 non-null   int64  
 32  Churn Reason       1869 non-null   object 
dtypes: float64(6), int64(3), object(24)
memory usage: 1.8+ MB
```

In [23]:
```
df.columns
```

Out[23]:
```
Index(['CustomerID', 'Count', 'Country', 'State', 'City', 'Zip Code',
       'Lat Long', 'Latitude', 'Longitude', 'Gender', 'Senior Citizen',
       'Partner', 'Dependents', 'Tenure Months', 'Phone Service',
       'Multiple Lines', 'Internet Service', 'Online Security',
       'Online Backup', 'Device Protection', 'Tech Support', 'Streaming TV',
       'Streaming Movies', 'Contract', 'Paperless Billing', 'Payment Method',
       'Monthly Charges', 'Total Charges', 'Churn Label', 'Churn Value',
       'Churn Score', 'CLTV', 'Churn Reason'],
      dtype='object')
```

Results of Step 2

- There are no missing values and the dataset is clean and ready to prepare for analysis.

Breakdown of Variables

Identification variables:

- CustomerID

String variables:

- State, City, Zip Code, Lat Long

Numeric variables:

- Count, Latitude, Longitude, Tenure Months, Monthly Charges, Total Charges, Churn Value, Churn Score, CLTV

Binary variables:

- Senior Citizen, Partner, Dependents, Phone Service, Multiple Lines, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Paperless Billing, Churn Label

Possible Categorical variables (may be string, will investigate further):

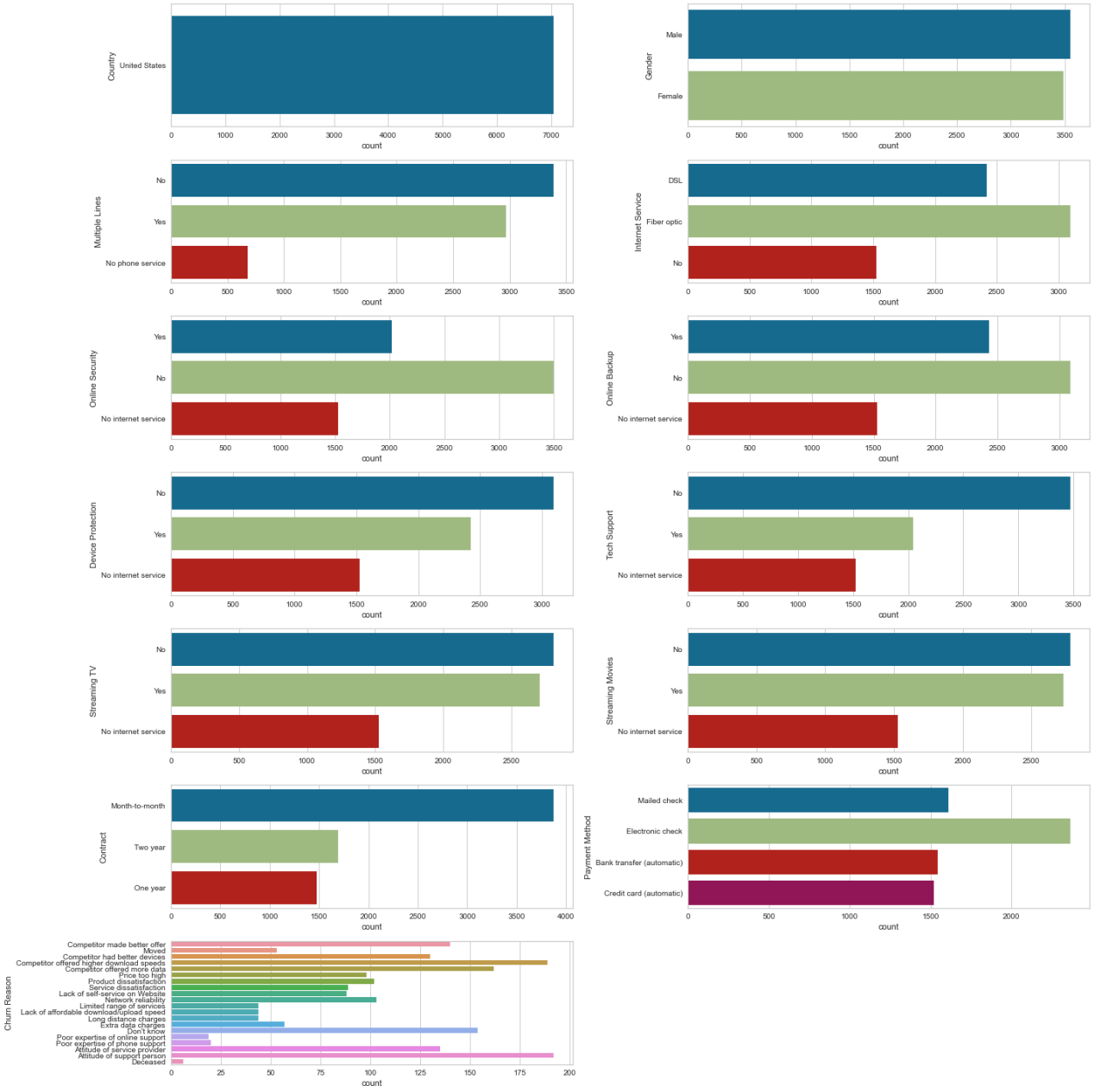
- Country, Gender, Multiple Lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Streaming Movies, Contract, Payment Method, Churn Reason

Step 3

In [24]:
```
# Convert binary variables into yes = 1, no = 0 (ref 1)
cols = ['Senior Citizen', 'Partner', 'Dependents', 'Phone Service', 'Paperless Billing', 'Churn Label']
df[cols] = df[cols].replace(to_replace = ['No', 'Yes'], value = [0, 1])
```

Step 4

In [28]:
```
# View bar charts for potential categorical variables to determine number of categories
figure, axes = plt.subplots(nrows=7, ncols=2, figsize=(20,20))
plt.subplot(7, 2, 1)
sns.countplot(data = df, y = 'Country')
plt.subplot(7, 2, 2)
sns.countplot(data = df, y = 'Gender')
plt.subplot(7, 2, 3)
sns.countplot(data = df, y = 'Multiple Lines')
plt.subplot(7, 2, 4)
sns.countplot(data = df, y = 'Internet Service')
plt.subplot(7, 2, 5)
sns.countplot(data = df, y = 'Online Security')
plt.subplot(7, 2, 6)
sns.countplot(data = df, y = 'Online Backup')
plt.subplot(7, 2, 7)
sns.countplot(data = df, y = 'Device Protection')
plt.subplot(7, 2, 8)
sns.countplot(data = df, y = 'Tech Support')
plt.subplot(7, 2, 9)
sns.countplot(data = df, y = 'Streaming TV')
plt.subplot(7, 2, 10)
sns.countplot(data = df, y = 'Streaming Movies')
plt.subplot(7, 2, 11)
sns.countplot(data = df, y = 'Contract')
plt.subplot(7, 2, 12)
sns.countplot(data = df, y = 'Payment Method')
plt.subplot(7, 2, 13)
sns.countplot(data = df, y = 'Churn Reason')
plt.subplot(7, 2, 14).set_visible(False)
figure.tight_layout()
plt.show();
```



- Country appears to currently only have one value, so will be treated as a string. The remaining variables will be converted into dummy variables.

In [29]:
```
# Create separate variables for each categorical value, with a 1 if the value is present in that row and 0 if
 not present
df = pd.get_dummies(data=df, columns=['Gender', 'Multiple Lines', 'Internet Service', 'Online Security', 'Onli
ne Backup',
                                      'Device Protection', 'Tech Support', 'Streaming TV', 'Streaming Movies',
                                      'Contract', 'Payment Method', 'Churn Reason'])
```

Step 5

In [30]:
```
# Drop columns not needed for analysis
drops = ['CustomerID', 'Country', 'State', 'City', 'Zip Code', 'Lat Long']
df = df.drop(drops, axis = 1)
```

VISUALIZATIONS

Univariate Visualizations

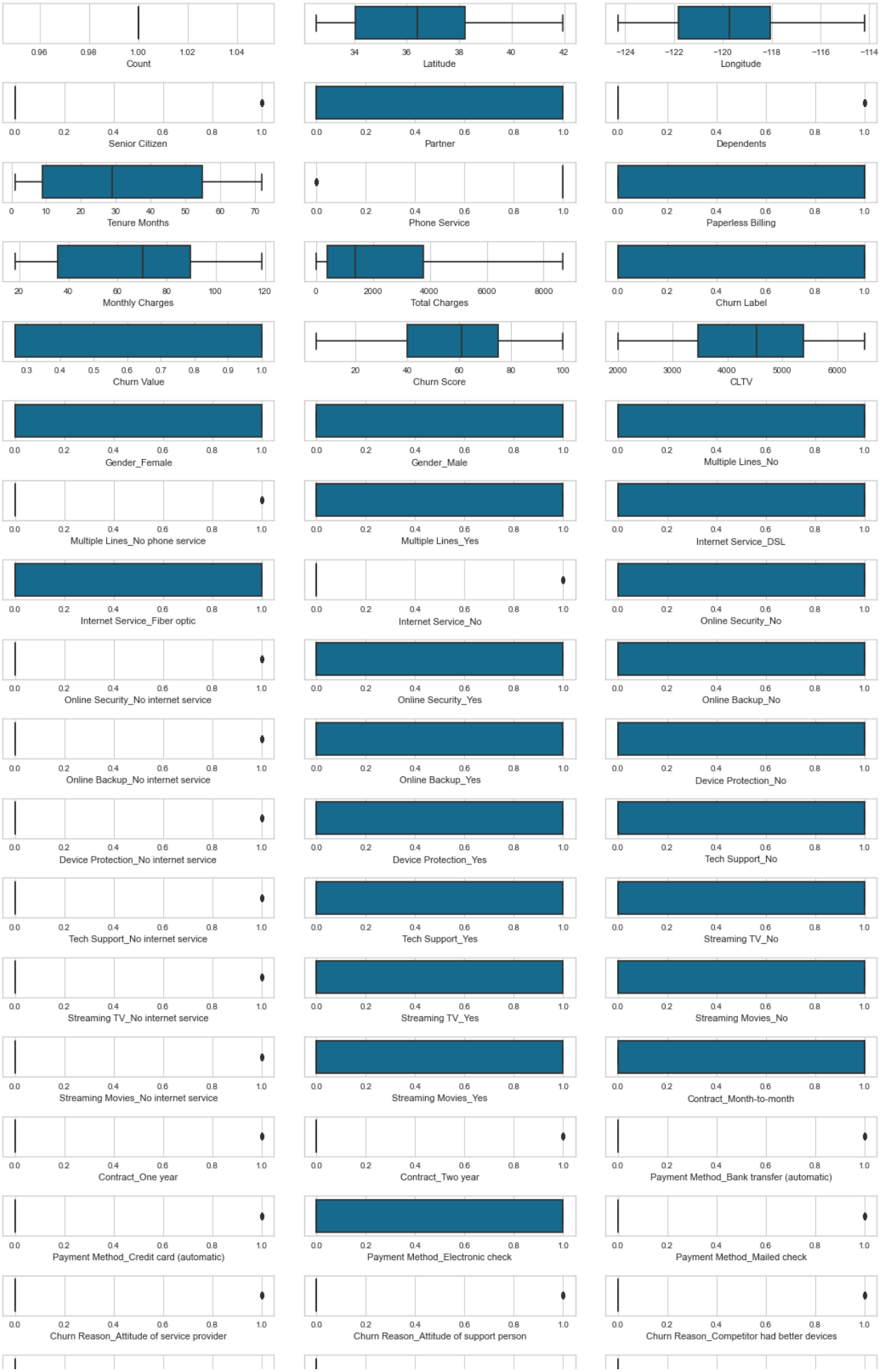
In [31]:
```
df.info()
```

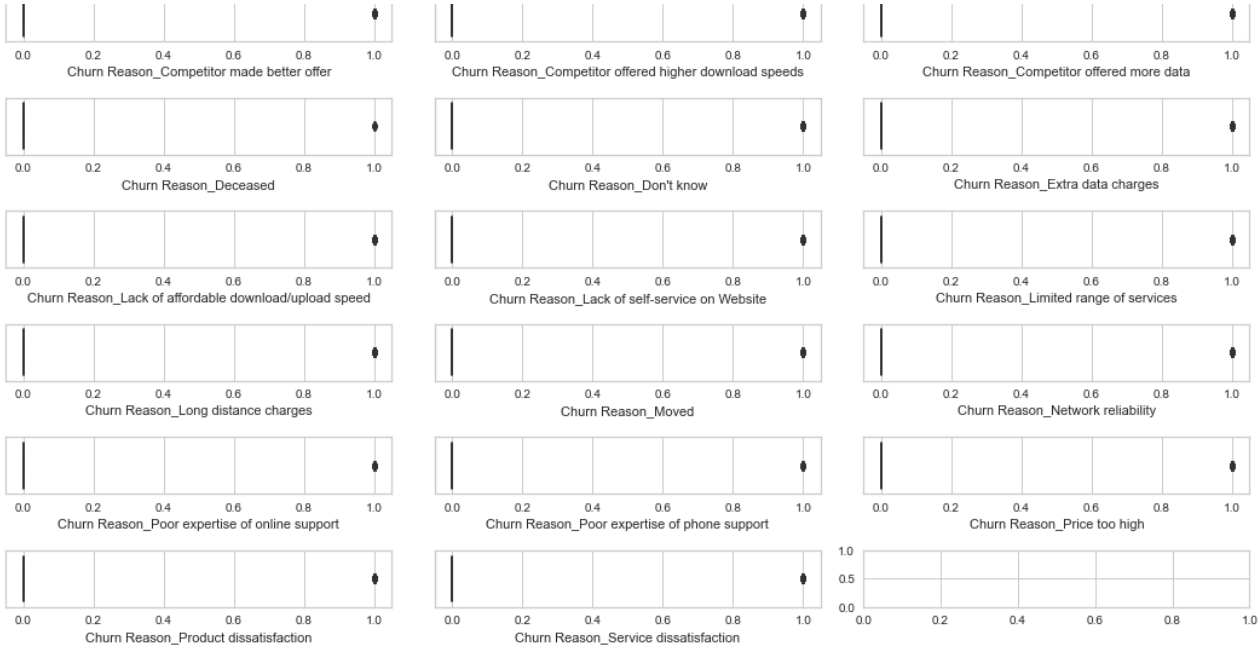
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 68 columns):
 #   Column                                                 Non-Null Count  Dtype  
---  ------                                                 --------------  -----  
 0   Count                                                  7043 non-null   int64  
 1   Latitude                                               7043 non-null   float64
 2   Longitude                                              7043 non-null   float64
 3   Senior Citizen                                         7043 non-null   int64  
 4   Partner                                                7043 non-null   int64  
 5   Dependents                                             7043 non-null   int64  
 6   Tenure Months                                          7043 non-null   float64
 7   Phone Service                                          7043 non-null   int64  
 8   Paperless Billing                                      7043 non-null   int64  
 9   Monthly Charges                                        7043 non-null   float64
 10  Total Charges                                          7043 non-null   float64
 11  Churn Label                                            7043 non-null   int64  
 12  Churn Value                                            7043 non-null   float64
 13  Churn Score                                            7043 non-null   int64  
 14  CLTV                                                   7043 non-null   int64  
 15  Gender_Female                                          7043 non-null   uint8  
 16  Gender_Male                                            7043 non-null   uint8  
 17  Multiple Lines_No                                      7043 non-null   uint8  
 18  Multiple Lines_No phone service                        7043 non-null   uint8  
 19  Multiple Lines_Yes                                     7043 non-null   uint8  
 20  Internet Service_DSL                                   7043 non-null   uint8  
 21  Internet Service_Fiber optic                           7043 non-null   uint8  
 22  Internet Service_No                                    7043 non-null   uint8  
 23  Online Security_No                                     7043 non-null   uint8  
 24  Online Security_No internet service                    7043 non-null   uint8  
 25  Online Security_Yes                                    7043 non-null   uint8  
 26  Online Backup_No                                       7043 non-null   uint8  
 27  Online Backup_No internet service                      7043 non-null   uint8  
 28  Online Backup_Yes                                      7043 non-null   uint8  
 29  Device Protection_No                                   7043 non-null   uint8  
 30  Device Protection_No internet service                  7043 non-null   uint8  
 31  Device Protection_Yes                                  7043 non-null   uint8  
 32  Tech Support_No                                        7043 non-null   uint8  
 33  Tech Support_No internet service                       7043 non-null   uint8  
 34  Tech Support_Yes                                       7043 non-null   uint8  
 35  Streaming TV_No                                        7043 non-null   uint8  
 36  Streaming TV_No internet service                       7043 non-null   uint8  
 37  Streaming TV_Yes                                       7043 non-null   uint8  
 38  Streaming Movies_No                                    7043 non-null   uint8  
 39  Streaming Movies_No internet service                   7043 non-null   uint8  
 40  Streaming Movies_Yes                                   7043 non-null   uint8  
 41  Contract_Month-to-month                                7043 non-null   uint8  
 42  Contract_One year                                      7043 non-null   uint8  
 43  Contract_Two year                                      7043 non-null   uint8  
 44  Payment Method_Bank transfer (automatic)               7043 non-null   uint8  
 45  Payment Method_Credit card (automatic)                 7043 non-null   uint8  
 46  Payment Method_Electronic check                        7043 non-null   uint8  
 47  Payment Method_Mailed check                            7043 non-null   uint8  
 48  Churn Reason_Attitude of service provider              7043 non-null   uint8  
 49  Churn Reason_Attitude of support person                7043 non-null   uint8  
 50  Churn Reason_Competitor had better devices             7043 non-null   uint8  
 51  Churn Reason_Competitor made better offer              7043 non-null   uint8  
 52  Churn Reason_Competitor offered higher download speeds 7043 non-null   uint8  
 53  Churn Reason_Competitor offered more data              7043 non-null   uint8  
 54  Churn Reason_Deceased                                  7043 non-null   uint8  
 55  Churn Reason_Don't know                                7043 non-null   uint8  
 56  Churn Reason_Extra data charges                        7043 non-null   uint8  
 57  Churn Reason_Lack of affordable download/upload speed  7043 non-null   uint8  
 58  Churn Reason_Lack of self-service on Website           7043 non-null   uint8  
 59  Churn Reason_Limited range of services                 7043 non-null   uint8  
 60  Churn Reason_Long distance charges                     7043 non-null   uint8  
 61  Churn Reason_Moved                                     7043 non-null   uint8  
 62  Churn Reason_Network reliability                       7043 non-null   uint8  
 63  Churn Reason_Poor expertise of online support          7043 non-null   uint8  
 64  Churn Reason_Poor expertise of phone support           7043 non-null   uint8  
 65  Churn Reason_Price too high                            7043 non-null   uint8  
 66  Churn Reason_Product dissatisfaction                   7043 non-null   uint8  
 67  Churn Reason_Service dissatisfaction                   7043 non-null   uint8  
dtypes: float64(6), int64(9), uint8(53)
memory usage: 1.2 MB
```

In [32]:
```
df.columns
```

Out[32]:
```
Index(['Count', 'Latitude', 'Longitude', 'Senior Citizen', 'Partner',
       'Dependents', 'Tenure Months', 'Phone Service', 'Paperless Billing',
       'Monthly Charges', 'Total Charges', 'Churn Label', 'Churn Value',
       'Churn Score', 'CLTV', 'Gender_Female', 'Gender_Male',
       'Multiple Lines_No', 'Multiple Lines_No phone service',
       'Multiple Lines_Yes', 'Internet Service_DSL',
       'Internet Service_Fiber optic', 'Internet Service_No',
       'Online Security_No', 'Online Security_No internet service',
       'Online Security_Yes', 'Online Backup_No',
       'Online Backup_No internet service', 'Online Backup_Yes',
       'Device Protection_No', 'Device Protection_No internet service',
       'Device Protection_Yes', 'Tech Support_No',
       'Tech Support_No internet service', 'Tech Support_Yes',
       'Streaming TV_No', 'Streaming TV_No internet service',
       'Streaming TV_Yes', 'Streaming Movies_No',
       'Streaming Movies_No internet service', 'Streaming Movies_Yes',
       'Contract_Month-to-month', 'Contract_One year', 'Contract_Two year',
       'Payment Method_Bank transfer (automatic)',
       'Payment Method_Credit card (automatic)',
       'Payment Method_Electronic check', 'Payment Method_Mailed check',
       'Churn Reason_Attitude of service provider',
       'Churn Reason_Attitude of support person',
       'Churn Reason_Competitor had better devices',
       'Churn Reason_Competitor made better offer',
       'Churn Reason_Competitor offered higher download speeds',
       'Churn Reason_Competitor offered more data', 'Churn Reason_Deceased',
       'Churn Reason_Don't know', 'Churn Reason_Extra data charges',
       'Churn Reason_Lack of affordable download/upload speed',
       'Churn Reason_Lack of self-service on Website',
       'Churn Reason_Limited range of services',
       'Churn Reason_Long distance charges', 'Churn Reason_Moved',
       'Churn Reason_Network reliability',
       'Churn Reason_Poor expertise of online support',
       'Churn Reason_Poor expertise of phone support',
       'Churn Reason_Price too high', 'Churn Reason_Product dissatisfaction',
       'Churn Reason_Service dissatisfaction'],
      dtype='object')
```

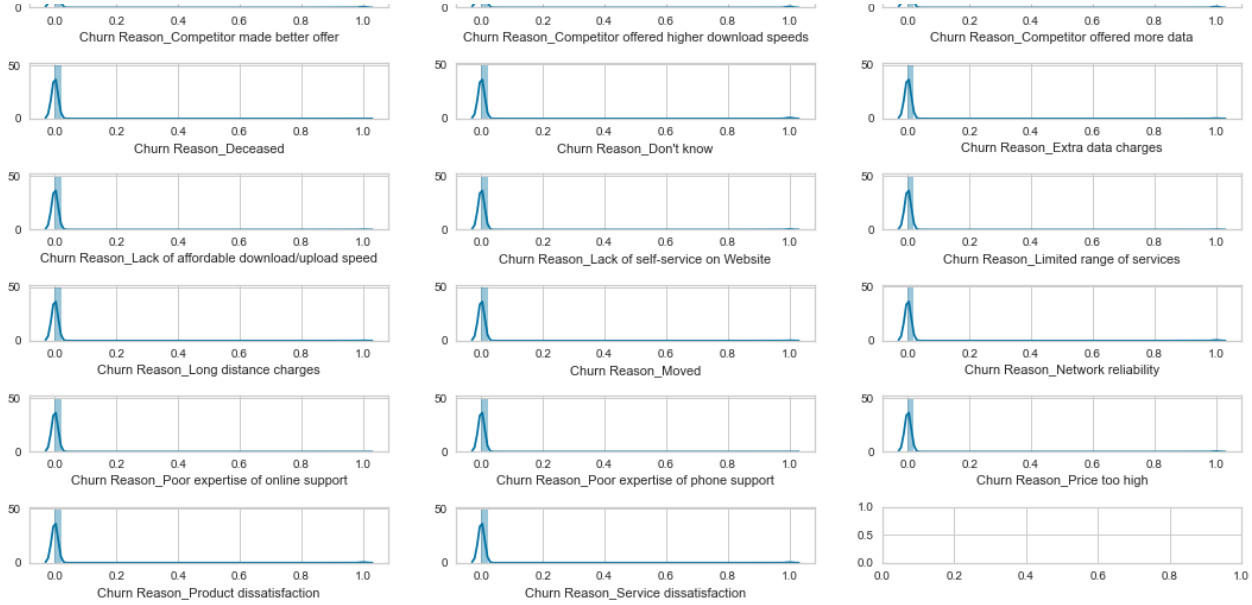
In [35]:
```
# Display boxplots of numeric columns (ref 2)
cols = df.columns
f, axes = plt.subplots(round(len(cols)/3), 3, figsize=(15,30))
y = 0;
for col in cols:
    i, j = divmod(y, 3)
    sns.boxplot(x=df[col], ax=axes[i, j])
    y = y + 1

plt.tight_layout()
plt.show();
```


Count
Latitude
Longitude
Senior Citizen
Partner
Dependents
Tenure Months
Phone Service
Paperless Billing
Monthly Charges
Total Charges
Churn Label
Churn Value
Churn Score
CLTV
Gender_Female
Gender_Male
Multiple Lines_No
Multiple Lines_No phone service
Multiple Lines_Yes
Internet Service_DSL
Internet Service_Fiber optic
Internet Service_No
Online Security_No
Online Security_No internet service
Online Security_Yes
Online Backup_No
Online Backup_No internet service
Online Backup_Yes
Device Protection_No
Device Protection_No internet service
Device Protection_Yes
Tech Support_No
Tech Support_No internet service
Tech Support_Yes
Streaming TV_No
Streaming TV_No internet service
Streaming TV_Yes
Streaming Movies_No
Streaming Movies_No internet service
Streaming Movies_Yes
Contract_Month-to-month
Contract_One year
Contract_Two year
Payment Method_Bank transfer (automatic)
Payment Method_Credit card (automatic)
Payment Method_Electronic check
Payment Method_Mailed check
Churn Reason_Attitude of service provider
Churn Reason_Attitude of support person
Churn Reason_Competitor had better devices


Churn Reason_Competitor made better offer
Churn Reason_Competitor offered higher download speeds
Churn Reason_Competitor offered more data
Churn Reason_Deceased
Churn Reason_Don't know
Churn Reason_Extra data charges
Churn Reason_Lack of affordable download/upload speed
Churn Reason_Lack of self-service on Website
Churn Reason_Limited range of services
Churn Reason_Long distance charges
Churn Reason_Moved
Churn Reason_Network reliability
Churn Reason_Poor expertise of online support
Churn Reason_Poor expertise of phone support
Churn Reason_Price too high
Churn Reason_Product dissatisfaction
Churn Reason_Service dissatisfaction
0.0 0.2 0.4 0.6 0.8 1.0
0.0 0.5 1.0

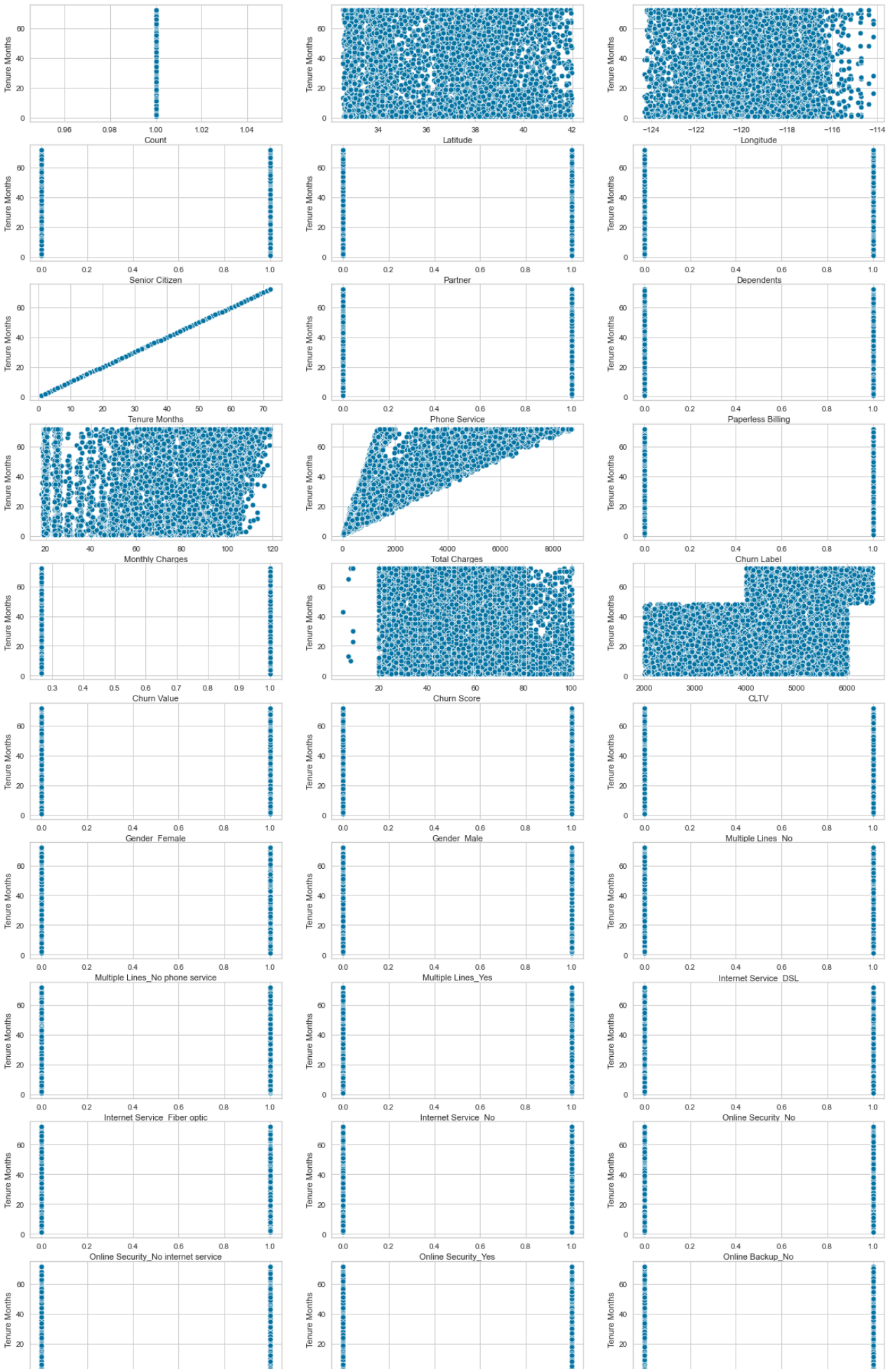
In [36]:
```
# Display histograms of numeric columns (ref 2)
f, axes = plt.subplots(round(len(cols)/3), 3, figsize=(15,30))
y = 0;
for col in cols:
    i, j = divmod(y, 3)
    sns.distplot(df[col], ax=axes[i, j], kde_kws={'bw':0.01})
    y = y + 1

plt.tight_layout()
plt.show();
```

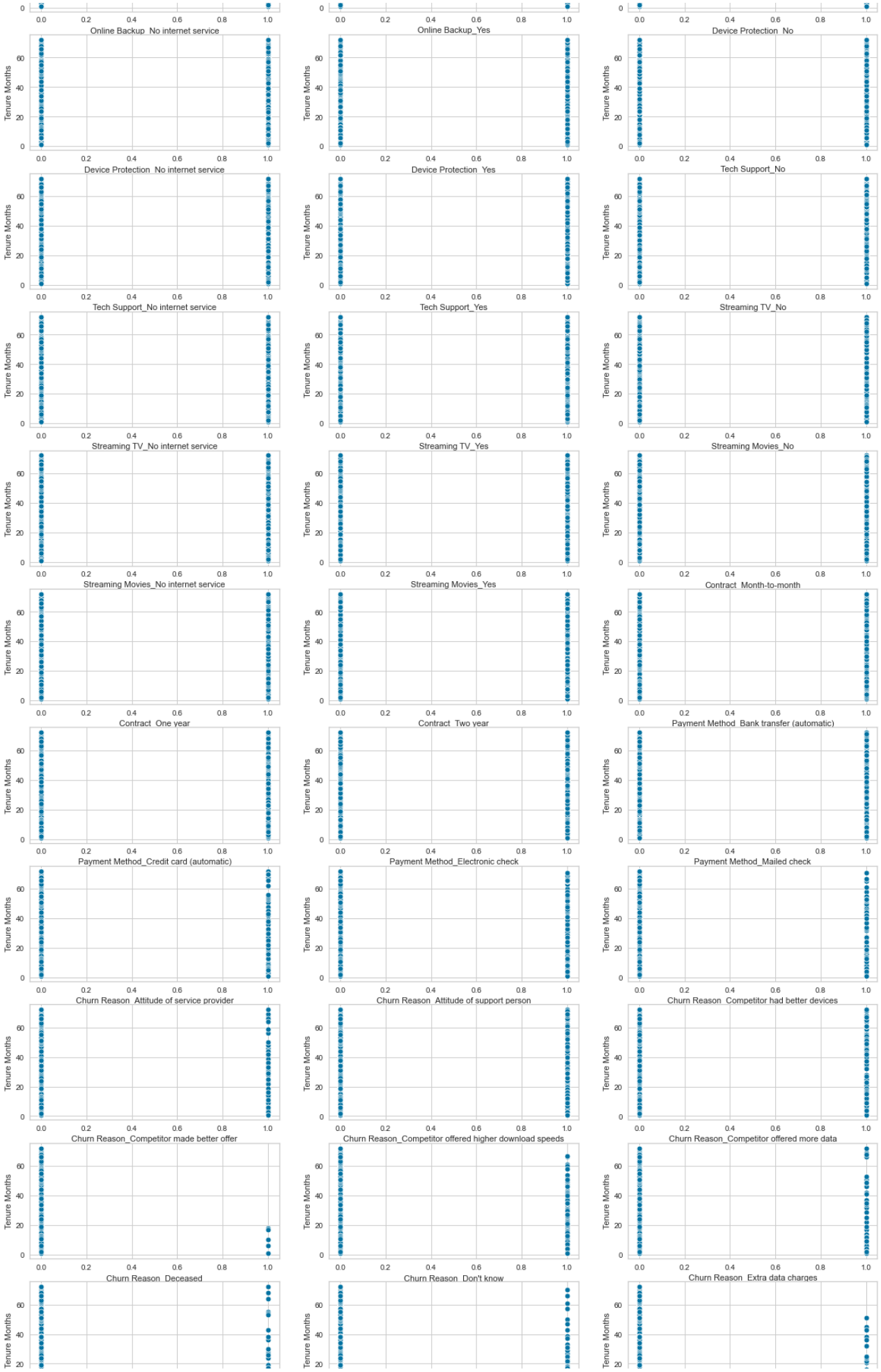

Count
Latitude
Longitude
Senior Citizen
Partner
Dependents
Tenure Months
Phone Service
Paperless Billing
Monthly Charges
Total Charges
Churn Label
Churn Value
Churn Score
CLTV
Gender_Female
Gender_Male
Multiple Lines_No
Multiple Lines_No phone service
Multiple Lines_Yes
Internet Service_DSL
Internet Service_Fiber optic
Internet Service_No
Online Security_No
Online Security_No internet service
Online Security_Yes
Online Backup_No
Online Backup_No internet service
Online Backup_Yes
Device Protection_No
Device Protection_No internet service
Device Protection_Yes
Tech Support_No
Tech Support_No internet service
Tech Support_Yes
Streaming TV_No
Streaming TV_No internet service
Streaming TV_Yes
Streaming Movies_No
Streaming Movies_No internet service
Streaming Movies_Yes
Contract_Month-to-month
Contract_One year
Contract_Two year
Payment Method_Bank transfer (automatic)
Payment Method_Credit card (automatic)
Payment Method_Electronic check
Payment Method_Mailed check
Churn Reason_Attitude of service provider
Churn Reason_Attitude of support person
Churn Reason_Competitor had better devices

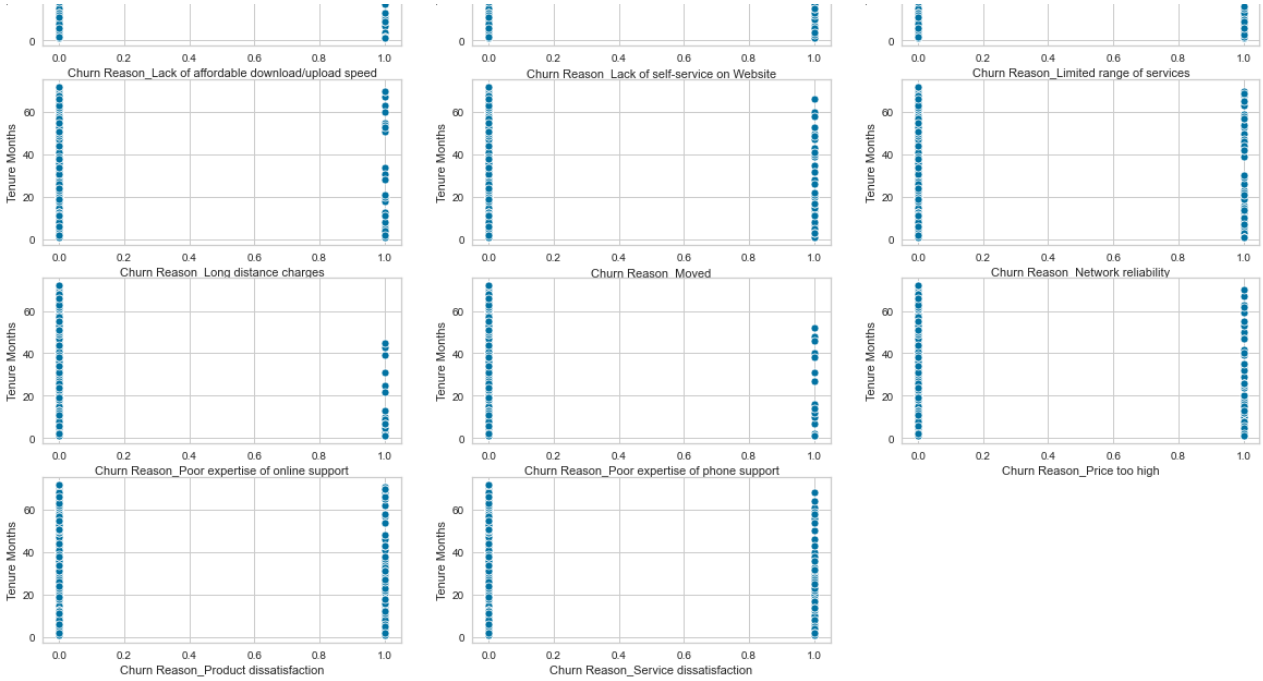


Bivariate Visualizations

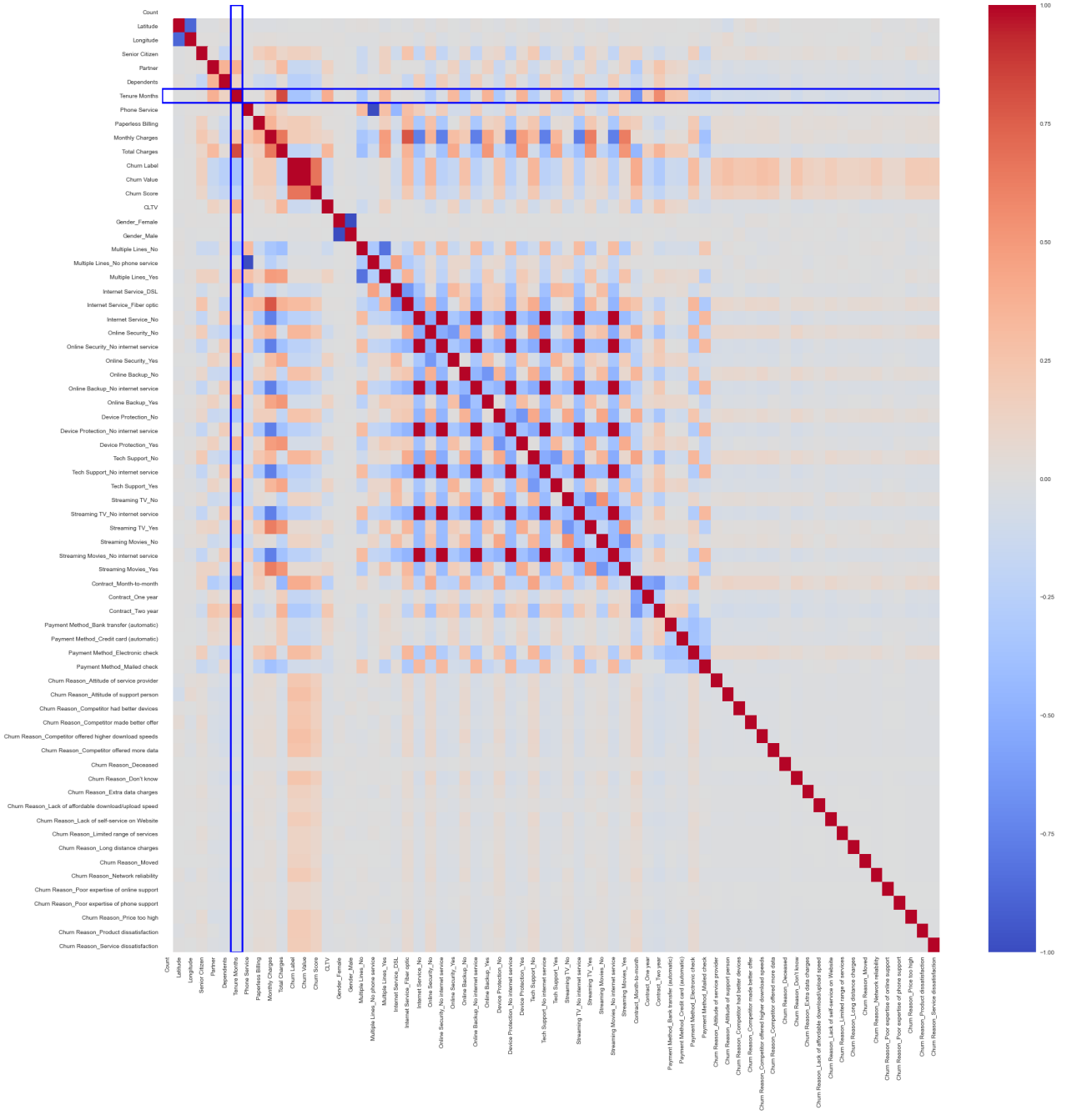
In [41]:
```
# Plot scatter plots for each predictive variable on the x axis and Tenure on y axis (Ref 3)
count=1
plt.subplots(figsize=(20, 80))
for i in df.columns:
    plt.subplot(24,3,count)
    sns.scatterplot(df[i], df["Tenure Months"])
    count+=1

plt.show()
```


Tenure Months
Count
Latitude
Longitude
Senior Citizen
Partner
Dependents
Tenure Months
Phone Service
Paperless Billing
Monthly Charges
Total Charges
Churn Label
Churn Value
Churn Score
CLTV
Gender_Female
Gender_Male
Multiple Lines_No
Multiple Lines_No phone service
Multiple Lines_Yes
Internet Service_DSL
Internet Service_Fiber optic
Internet Service_No
Online Security_No
Online Security_No internet service
Online Security_Yes
Online Backup_No


Online Backup_No internet service
Online Backup_Yes
Device Protection_No
Device Protection_No internet service
Device Protection_Yes
Tech Support_No
Tech Support_No internet service
Tech Support_Yes
Streaming TV_No
Streaming TV_No internet service
Streaming TV_Yes
Streaming Movies_No
Streaming Movies_No internet service
Streaming Movies_Yes
Contract_Month-to-month
Contract_One year
Contract_Two year
Payment Method_Bank transfer (automatic)
Payment Method_Credit card (automatic)
Payment Method_Electronic check
Payment Method_Mailed check
Churn Reason_Attitude of service provider
Churn Reason_Attitude of support person
Churn Reason_Competitor had better devices
Churn Reason_Competitor made better offer
Churn Reason_Competitor offered higher download speeds
Churn Reason_Competitor offered more data
Churn Reason_Deceased
Churn Reason_Don't know
Churn Reason_Extra data charges
Tenure Months


Churn Reason_Lack of affordable download/upload speed
Churn Reason_Lack of self-service on Website
Churn Reason_Limited range of services
Churn Reason_Long distance charges
Churn Reason_Moved
Churn Reason_Network reliability
Churn Reason_Poor expertise of online support
Churn Reason_Poor expertise of phone support
Churn Reason_Price too high
Churn Reason_Product dissatisfaction
Churn Reason_Service dissatisfaction
Tenure Months

In [42]:
```
# Display heatmap to view correlation between variables highlighting the Tenure variable (ref 4)
t = df.columns.get_loc('Tenure Months')
l = len(df.columns)
fig, ax = plt.subplots(figsize=(30,30))
sns.heatmap(df.corr(), cmap='coolwarm')
ax.add_patch(Rectangle((t, 0), 1, l, fill=False, edgecolor='blue', lw=3))
ax.add_patch(Rectangle((0, t), l, 1, fill=False, edgecolor='blue', lw=3))
plt.show()
```



Prepared Data

In [43]:
```
# Save cleaned dataframe to CSV
df.to_csv('churn_clean_data_final.csv', index = False, encoding = 'utf-8')
```

Initial Model

In [44]:
```
# Set up input matrix and response variable
observations = len(df)
variables = df.columns[:-1]
Xinit = df.iloc[:,:-1]
y = df['Tenure Months'].values
```

In [45]:
```
# Veiw number of independent variables
print ("There are", Xinit.shape[1], "independent variables in the initial model.")
```

```
There are 67 independent variables in the initial model.
```

In [46]:
```
# Add y-intercept, create model, and view summary
Xc_init = sm.add_constant(Xinit)
linear_regression = sm.OLS(y,Xc_init)
fitted_model1 = linear_regression.fit()
fitted_model1.summary()
```

Out[46]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	1.000
Model:	OLS	**Adj. R-squared:**	1.000
Method:	Least Squares	**F-statistic:**	4.657e+29
Date:	Mon, 03 May 2021	**Prob (F-statistic):**	0.00
Time:	20:10:37	**Log-Likelihood:**	1.9042e+05
No. Observations:	7043	**AIC:**	-3.807e+05
Df Residuals:	6995	**BIC:**	-3.804e+05
Df Model:	47		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Count	1.773e-12	1.05e-13	16.939	0.000	1.57e-12	1.98e-12
Latitude	3.685e-15	4.46e-15	0.826	0.409	-5.06e-15	1.24e-14
Longitude	7.803e-14	5.07e-15	15.395	0.000	6.81e-14	8.8e-14
Senior Citizen	3.952e-14	1.52e-14	2.600	0.009	9.73e-15	6.93e-14
Partner	3.253e-14	1.22e-14	2.669	0.008	8.64e-15	5.64e-14
Dependents	1.24e-13	1.4e-14	8.830	0.000	9.65e-14	1.52e-13
Tenure Months	1.0000	5.93e-16	1.69e+15	0.000	1.000	1.000
Phone Service	5.627e-13	7.53e-14	7.472	0.000	4.15e-13	7.1e-13
Paperless Billing	-3.226e-13	1.18e-14	-27.439	0.000	-3.46e-13	-3e-13
Monthly Charges	1.013e-14	5.13e-15	1.973	0.049	6.48e-17	2.02e-14
Total Charges	-1.433e-16	7.64e-18	-18.760	0.000	-1.58e-16	-1.28e-16
Churn Label	-2.271e-13	3.46e-14	-6.569	0.000	-2.95e-13	-1.59e-13
Churn Value	3.042e-13	2.95e-14	10.323	0.000	2.46e-13	3.62e-13
Churn Score	-6.644e-16	3.27e-16	-2.033	0.042	-1.3e-15	-2.38e-17
CLTV	5.29e-17	4.85e-18	10.907	0.000	4.34e-17	6.24e-17
Gender_Female	8.891e-13	5.25e-14	16.923	0.000	7.86e-13	9.92e-13
Gender_Male	8.831e-13	5.26e-14	16.773	0.000	7.8e-13	9.86e-13
Multiple Lines_No	2.91e-13	2.8e-14	10.403	0.000	2.36e-13	3.46e-13
Multiple Lines_No phone service	1.211e-12	5.98e-14	20.236	0.000	1.09e-12	1.33e-12
Multiple Lines_Yes	2.758e-13	4.96e-14	5.555	0.000	1.78e-13	3.73e-13
Internet Service_DSL	6.442e-13	4.67e-14	13.783	0.000	5.53e-13	7.36e-13
Internet Service_Fiber optic	4.978e-13	9.91e-14	5.021	0.000	3.03e-13	6.92e-13
Internet Service_No	6.293e-13	3.36e-14	18.708	0.000	5.63e-13	6.95e-13
Online Security_No	5.952e-13	3.6e-14	16.552	0.000	5.25e-13	6.66e-13
Online Security_No internet service	6.225e-13	3.36e-14	18.507	0.000	5.57e-13	6.88e-13
Online Security_Yes	5.613e-13	5.33e-14	10.530	0.000	4.57e-13	6.66e-13
Online Backup_No	5.9e-13	3.61e-14	16.327	0.000	5.19e-13	6.61e-13
Online Backup_No internet service	6.2e-13	3.36e-14	18.432	0.000	5.54e-13	6.86e-13
Online Backup_Yes	5.5e-13	5.31e-14	10.366	0.000	4.46e-13	6.54e-13
Device Protection_No	5.839e-13	3.61e-14	16.176	0.000	5.13e-13	6.55e-13
Device Protection_No internet service	6.261e-13	3.36e-14	18.614	0.000	5.6e-13	6.92e-13
Device Protection_Yes	5.569e-13	5.32e-14	10.472	0.000	4.53e-13	6.61e-13
Tech Support_No	5.949e-13	3.6e-14	16.526	0.000	5.24e-13	6.65e-13
Tech Support_No internet service	6.238e-13	3.36e-14	18.546	0.000	5.58e-13	6.9e-13
Tech Support_Yes	5.609e-13	5.33e-14	10.517	0.000	4.56e-13	6.65e-13
Streaming TV_No	6.072e-13	3.23e-14	18.776	0.000	5.44e-13	6.71e-13
Streaming TV_No internet service	6.213e-13	3.36e-14	18.472	0.000	5.55e-13	6.87e-13
Streaming TV_Yes	5.36e-13	6.38e-14	8.408	0.000	4.11e-13	6.61e-13

Streaming Movies_No	5.946e-13	3.21e-14	18.543	0.000	5.32e-13	6.57e-13
Streaming Movies_No internet service	6.253e-13	3.36e-14	18.589	0.000	5.59e-13	6.91e-13
Streaming Movies_Yes	5.24e-13	6.39e-14	8.204	0.000	3.99e-13	6.49e-13
Contract_Month-to-month	5.906e-13	3.64e-14	16.217	0.000	5.19e-13	6.62e-13
Contract_One year	5.954e-13	3.59e-14	16.565	0.000	5.25e-13	6.66e-13
Contract_Two year	5.951e-13	3.68e-14	16.187	0.000	5.23e-13	6.67e-13
Payment Method_Bank transfer (automatic)	4.473e-13	2.81e-14	15.936	0.000	3.92e-13	5.02e-13
Payment Method_Credit card (automatic)	4.456e-13	2.81e-14	15.855	0.000	3.91e-13	5.01e-13
Payment Method_Electronic check	4.481e-13	2.77e-14	16.179	0.000	3.94e-13	5.02e-13
Payment Method_Mailed check	4.359e-13	2.82e-14	15.480	0.000	3.81e-13	4.91e-13
Churn Reason_Attitude of service provider	2.209e-14	6.01e-14	0.367	0.713	-9.58e-14	1.4e-13
Churn Reason_Attitude of support person	3.775e-15	5.66e-14	0.067	0.947	-1.07e-13	1.15e-13
Churn Reason_Competitor had better devices	3.375e-14	6.06e-14	0.557	0.578	-8.51e-14	1.53e-13
Churn Reason_Competitor made better offer	3.619e-14	5.98e-14	0.605	0.545	-8.1e-14	1.53e-13
Churn Reason_Competitor offered higher download speeds	3.131e-14	5.67e-14	0.552	0.581	-7.98e-14	1.42e-13
Churn Reason_Competitor offered more data	2.245e-14	5.81e-14	0.387	0.699	-9.14e-14	1.36e-13
Churn Reason_Deceased	-3.142e-14	1.86e-13	-0.169	0.866	-3.96e-13	3.33e-13
Churn Reason_Don't know	1.998e-14	5.87e-14	0.340	0.734	-9.51e-14	1.35e-13
Churn Reason_Extra data charges	1.266e-14	7.48e-14	0.169	0.866	-1.34e-13	1.59e-13
Churn Reason_Lack of affordable download/upload speed	2.232e-14	8.12e-14	0.275	0.783	-1.37e-13	1.81e-13
Churn Reason_Lack of self-service on Website	2.354e-14	6.62e-14	0.355	0.722	-1.06e-13	1.53e-13
Churn Reason_Limited range of services	1.16e-14	8.12e-14	0.143	0.886	-1.48e-13	1.71e-13
Churn Reason_Long distance charges	1.776e-14	8.12e-14	0.219	0.827	-1.41e-13	1.77e-13
Churn Reason_Moved	-3.153e-14	7.65e-14	-0.412	0.680	-1.82e-13	1.18e-13
Churn Reason_Network reliability	1.688e-14	6.38e-14	0.265	0.791	-1.08e-13	1.42e-13
Churn Reason_Poor expertise of online support	-1.077e-14	1.11e-13	-0.097	0.923	-2.29e-13	2.08e-13
Churn Reason_Poor expertise of phone support	1.643e-14	1.09e-13	0.151	0.880	-1.97e-13	2.3e-13
Churn Reason_Price too high	2.004e-14	6.45e-14	0.311	0.756	-1.06e-13	1.47e-13
Churn Reason_Product dissatisfaction	2.211e-14	6.39e-14	0.346	0.729	-1.03e-13	1.47e-13

Omnibus:	188.101	**Durbin-Watson:**	1.326
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	189.731
Skew:	0.376	**Prob(JB):**	6.32e-42
Kurtosis:	2.714	**Cond. No.**	3.82e+16

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.34e-22. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Model Reduction

The model has given us the highest R squared value possible which should be the mark of an excellent fit, however when so many variables are present, it tends to inflate the R squared value (Massaron & Boschetti, 2016). One sign of model instability is that the condition number is extremely high, which is a sign of multicollinearity between variables. Also, there are several variables with p-values above a 0.05 confidence interval, meaning that their coefficients are not significant to the model I will begin by removing those variables and also the Tenure Months variable which is our response variable and therefore should not be present in the model.

In [47]:
```
# Remove response and insignificant variables
drops = ['Latitude', 'Churn Reason_Attitude of service provider', 'Churn Reason_Attitude of support person',
         'Churn Reason_Competitor had better devices', 'Churn Reason_Competitor made better offer',
         'Churn Reason_Competitor offered higher download speeds', 'Churn Reason_Competitor offered more data'
,
         'Churn Reason_Deceased', 'Churn Reason_Don\'t know' , 'Churn Reason_Extra data charges',
         'Churn Reason_Lack of affordable download/upload speed','Churn Reason_Lack of self-service on Websit
e',
         'Churn Reason_Limited range of services', 'Churn Reason_Long distance charges', 'Churn Reason_Moved',
         'Churn Reason_Network reliability', 'Churn Reason_Poor expertise of online support',
         'Churn Reason_Poor expertise of phone support', 'Churn Reason_Price too high',
         'Churn Reason_Product dissatisfaction', 'Tenure Months']
Xfin = Xinit.drop(drops, axis = 1)
```

In [48]:
```
# Re-run model
Xcfin = sm.add_constant(Xfin)
linear_regression = sm.OLS(y,Xcfin)
fitted_model2 = linear_regression.fit()
fitted_model2.summary()
```

Out[48]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.870
Model:	OLS	**Adj. R-squared:**	0.869
Method:	Least Squares	**F-statistic:**	1802.
Date:	Mon, 03 May 2021	**Prob (F-statistic):**	0.00
Time:	20:22:07	**Log-Likelihood:**	-25350.
No. Observations:	7043	**AIC:**	5.075e+04
Df Residuals:	7016	**BIC:**	5.094e+04
Df Model:	26		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Count	5.0526	1.509	3.348	0.001	2.095	8.011
Longitude	-0.0152	0.049	-0.309	0.757	-0.111	0.081
Senior Citizen	1.0004	0.305	3.277	0.001	0.402	1.599
Partner	2.3911	0.244	9.812	0.000	1.913	2.869
Dependents	-0.3983	0.283	-1.408	0.159	-0.953	0.156
Phone Service	2.0137	1.400	1.438	0.150	-0.731	4.758
Paperless Billing	0.5171	0.237	2.184	0.029	0.053	0.981
Monthly Charges	-0.3330	0.103	-3.225	0.001	-0.535	-0.131
Total Charges	0.0099	9.8e-05	101.264	0.000	0.010	0.010
Churn Label	-1.5421	0.299	-5.161	0.000	-2.128	-0.956
Churn Value	0.2079	0.312	0.666	0.506	-0.405	0.820
Churn Score	0.0056	0.007	0.859	0.391	-0.007	0.019
CLTV	0.0010	9.69e-05	10.039	0.000	0.001	0.001
Gender_Female	2.5002	0.760	3.288	0.001	1.010	3.991
Gender_Male	2.5524	0.763	3.344	0.001	1.056	4.049
Multiple Lines_No	-0.0603	0.482	-0.125	0.901	-1.006	0.885
Multiple Lines_No phone service	3.0390	0.812	3.744	0.000	1.448	4.630
Multiple Lines_Yes	2.0739	0.956	2.169	0.030	0.200	3.948
Internet Service_DSL	1.2965	0.801	1.618	0.106	-0.274	2.867
Internet Service_Fiber optic	2.1555	1.936	1.113	0.266	-1.640	5.951
Internet Service_No	1.6007	0.463	3.458	0.001	0.693	2.508
Online Security_No	1.9789	0.535	3.699	0.000	0.930	3.028
Online Security_No internet service	1.6007	0.463	3.458	0.001	0.693	2.508
Online Security_Yes	1.4730	0.955	1.542	0.123	-0.400	3.346
Online Backup_No	1.7683	0.536	3.301	0.001	0.718	2.818
Online Backup_No internet service	1.6007	0.463	3.458	0.001	0.693	2.508
Online Backup_Yes	1.6837	0.952	1.769	0.077	-0.183	3.550
Device Protection_No	2.0171	0.536	3.765	0.000	0.967	3.067
Device Protection_No internet service	1.6007	0.463	3.458	0.001	0.693	2.508
Device Protection_Yes	1.4349	0.954	1.504	0.133	-0.435	3.305
Tech Support_No	2.4854	0.535	4.645	0.000	1.437	3.534
Tech Support_No internet service	1.6007	0.463	3.458	0.001	0.693	2.508
Tech Support_Yes	0.9666	0.956	1.011	0.312	-0.908	2.841
Streaming TV_No	1.8740	0.423	4.434	0.000	1.045	2.703
Streaming TV_No internet service	1.6007	0.463	3.458	0.001	0.693	2.508
Streaming TV_Yes	1.5780	1.190	1.326	0.185	-0.754	3.910
Streaming Movies_No	1.9289	0.421	4.586	0.000	1.104	2.753
Streaming Movies_No internet service	1.6007	0.463	3.458	0.001	0.693	2.508

Streaming Movies_Yes	1.5231	1.190	1.280	0.201	-0.810	3.856
Contract_Month-to-month	-4.8342	0.543	-8.907	0.000	-5.898	-3.770
Contract_One year	1.8288	0.534	3.422	0.001	0.781	2.876
Contract_Two year	8.0580	0.544	14.801	0.000	6.991	9.125
Payment Method_Bank transfer (automatic)	3.1707	0.427	7.419	0.000	2.333	4.008
Payment Method_Credit card (automatic)	2.5205	0.427	5.901	0.000	1.683	3.358
Payment Method_Electronic check	1.9197	0.423	4.540	0.000	1.091	2.749
Payment Method_Mailed check	-2.5583	0.430	-5.956	0.000	-3.400	-1.716

Omnibus:	64.392	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	79.049
Skew:	0.159	**Prob(JB):**	6.84e-18
Kurtosis:	3.410	**Cond. No.**	2.15e+16

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.24e-22. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

- This has given a new list of high p-values, and those variables will be dropped and the model re-run

In [49]:
```
# Remove insignificant variables
drops = ['Longitude', 'Dependents', 'Phone Service', 'Churn Value', 'Churn Score', 'Multiple Lines_No',
         'Internet Service_DSL', 'Internet Service_Fiber optic', 'Online Security_Yes', 'Online Backup_Yes',
         'Device Protection_Yes', 'Tech Support_Yes', 'Streaming TV_Yes', 'Streaming Movies_Yes']
Xfin = Xfin.drop(drops, axis = 1)
```

In [50]:
```
# Re-run model
Xcfin = sm.add_constant(Xfin)
linear_regression = sm.OLS(y,Xcfin)
fitted_model2 = linear_regression.fit()
fitted_model2.summary()
```

Out[50]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.870
Model:	OLS	**Adj. R-squared:**	0.869
Method:	Least Squares	**F-statistic:**	2130.
Date:	Mon, 03 May 2021	**Prob (F-statistic):**	0.00
Time:	20:27:45	**Log-Likelihood:**	-25352.
No. Observations:	7043	**AIC:**	5.075e+04
Df Residuals:	7020	**BIC:**	5.091e+04
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Count	10.3901	0.612	16.979	0.000	9.191	11.590
Senior Citizen	1.0584	0.303	3.498	0.000	0.465	1.652
Partner	2.2669	0.228	9.945	0.000	1.820	2.714
Paperless Billing	0.5247	0.237	2.218	0.027	0.061	0.988
Monthly Charges	-0.2992	0.013	-23.372	0.000	-0.324	-0.274
Total Charges	0.0099	9.79e-05	101.463	0.000	0.010	0.010
Churn Label	-1.1419	0.282	-4.043	0.000	-1.696	-0.588
CLTV	0.0010	9.69e-05	10.032	0.000	0.001	0.001
Gender_Female	5.1690	0.324	15.945	0.000	4.533	5.804
Gender_Male	5.2211	0.323	16.149	0.000	4.587	5.855
Multiple Lines_No phone service	1.7531	0.553	3.173	0.002	0.670	2.836
Multiple Lines_Yes	1.9671	0.270	7.295	0.000	1.438	2.496
Internet Service_No	0.4897	0.143	3.435	0.001	0.210	0.769
Online Security_No	0.6847	0.279	2.454	0.014	0.138	1.232
Online Security_No internet service	0.4897	0.143	3.435	0.001	0.210	0.769
Online Backup_No	0.2586	0.271	0.953	0.341	-0.273	0.790
Online Backup_No internet service	0.4897	0.143	3.435	0.001	0.210	0.769
Device Protection_No	0.7499	0.280	2.680	0.007	0.201	1.298
Device Protection_No internet service	0.4897	0.143	3.435	0.001	0.210	0.769
Tech Support_No	1.6874	0.283	5.957	0.000	1.132	2.243
Tech Support_No internet service	0.4897	0.143	3.435	0.001	0.210	0.769
Streaming TV_No	0.6362	0.313	2.032	0.042	0.023	1.250
Streaming TV_No internet service	0.4897	0.143	3.435	0.001	0.210	0.769
Streaming Movies_No	0.7315	0.312	2.342	0.019	0.119	1.344
Streaming Movies_No internet service	0.4897	0.143	3.435	0.001	0.210	0.769
Contract_Month-to-month	-3.0520	0.300	-10.186	0.000	-3.639	-2.465
Contract_One year	3.6201	0.268	13.519	0.000	3.095	4.145
Contract_Two year	9.8221	0.284	34.589	0.000	9.265	10.379
Payment Method_Bank transfer (automatic)	4.5077	0.251	17.928	0.000	4.015	5.001
Payment Method_Credit card (automatic)	3.8569	0.249	15.478	0.000	3.368	4.345
Payment Method_Electronic check	3.2574	0.260	12.525	0.000	2.748	3.767
Payment Method_Mailed check	-1.2318	0.244	-5.042	0.000	-1.711	-0.753

Omnibus:	63.682	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	78.522
Skew:	0.156	**Prob(JB):**	8.89e-18
Kurtosis:	3.412	**Cond. No.**	9.68e+18

```
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.08e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

- The Online Backup_No variable now has a high p-value and will be dropped.

In [51]:
```
# Remove insignificant variables
drops = ['Online Backup_No']
Xfin = Xfin.drop(drops, axis = 1)
```

In [52]:
```
# Re-run model
Xcfin = sm.add_constant(Xfin)
linear_regression = sm.OLS(y,Xcfin)
fitted_model2 = linear_regression.fit()
fitted_model2.summary()
```

Out[52]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.870
Model:	OLS	**Adj. R-squared:**	0.869
Method:	Least Squares	**F-statistic:**	2232.
Date:	Mon, 03 May 2021	**Prob (F-statistic):**	0.00
Time:	20:29:26	**Log-Likelihood:**	-25352.
No. Observations:	7043	**AIC:**	5.075e+04
Df Residuals:	7021	**BIC:**	5.090e+04
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Count	10.5626	0.585	18.068	0.000	9.417	11.709
Senior Citizen	1.0628	0.303	3.513	0.000	0.470	1.656
Partner	2.2652	0.228	9.938	0.000	1.818	2.712
Paperless Billing	0.5253	0.237	2.220	0.026	0.062	0.989
Monthly Charges	-0.3010	0.013	-23.785	0.000	-0.326	-0.276
Total Charges	0.0099	9.56e-05	103.763	0.000	0.010	0.010
Churn Label	-1.1270	0.282	-3.996	0.000	-1.680	-0.574
CLTV	0.0010	9.69e-05	10.043	0.000	0.001	0.001
Gender_Female	5.2538	0.312	16.853	0.000	4.643	5.865
Gender_Male	5.3088	0.310	17.129	0.000	4.701	5.916
Multiple Lines_No phone service	1.6748	0.546	3.065	0.002	0.604	2.746
Multiple Lines_Yes	1.9853	0.269	7.381	0.000	1.458	2.513
Internet Service_No	0.4449	0.135	3.305	0.001	0.181	0.709
Online Security_No	0.6867	0.279	2.461	0.014	0.140	1.234
Online Security_No internet service	0.4449	0.135	3.305	0.001	0.181	0.709
Online Backup_No internet service	0.4449	0.135	3.305	0.001	0.181	0.709
Device Protection_No	0.7399	0.280	2.646	0.008	0.192	1.288
Device Protection_No internet service	0.4449	0.135	3.305	0.001	0.181	0.709
Tech Support_No	1.6922	0.283	5.975	0.000	1.137	2.247
Tech Support_No internet service	0.4449	0.135	3.305	0.001	0.181	0.709
Streaming TV_No	0.6106	0.312	1.958	0.050	-0.001	1.222
Streaming TV_No internet service	0.4449	0.135	3.305	0.001	0.181	0.709
Streaming Movies_No	0.7006	0.311	2.255	0.024	0.092	1.310
Streaming Movies_No internet service	0.4449	0.135	3.305	0.001	0.181	0.709
Contract_Month-to-month	-2.9900	0.292	-10.223	0.000	-3.563	-2.417
Contract_One year	3.6744	0.262	14.044	0.000	3.162	4.187
Contract_Two year	9.8782	0.278	35.559	0.000	9.334	10.423
Payment Method_Bank transfer (automatic)	4.5504	0.247	18.393	0.000	4.065	5.035
Payment Method_Credit card (automatic)	3.8984	0.245	15.889	0.000	3.417	4.379
Payment Method_Electronic check	3.3074	0.255	12.985	0.000	2.808	3.807
Payment Method_Mailed check	-1.1935	0.241	-4.953	0.000	-1.666	-0.721

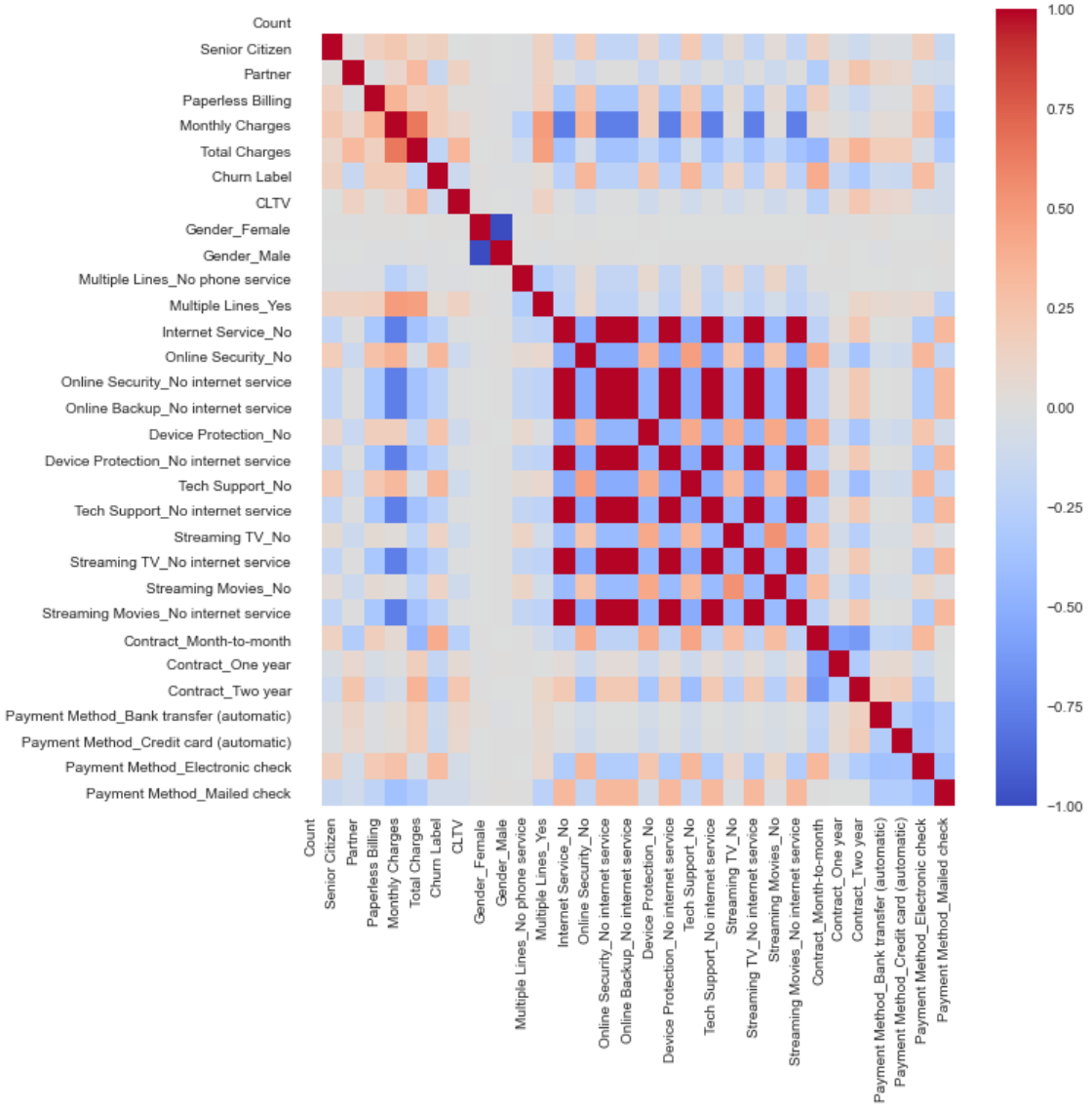
Omnibus:	64.522	**Durbin-Watson:**	2.013
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	79.699
Skew:	0.157	**Prob(JB):**	4.94e-18
Kurtosis:	3.415	**Cond. No.**	1.02e+19

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.88e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

- We now have a much more realistic R squared value. We have an extremely high condition number meaning multicollinearity may still be an issue. I will now look at the correlation between variables to determine if any can be removed and improve the model.

In [53]:
```
# Display heatmap to view correlation between predictor variables
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(Xfin.corr(), cmap='coolwarm')
plt.show()
```



- Several variables seem to be highly correlated. I will look for evidence of multicollinearity using Variable Inflation Factor analysis.

In [54]:
```
# Look for evidence of Variance Inflation Factors (ref 5)

# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = Xfin.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(Xfin.values, i)
                          for i in range(len(Xfin.columns))]

print(vif_data)
```

```
                                    feature        VIF
0                                     Count   0.000000
1                             Senior Citizen   1.114063
2                                    Partner   1.162271
3                           Paperless Billing   1.210871
4                             Monthly Charges  12.991177
5                               Total Charges   4.196093
6                                 Churn Label   1.389010
7                                        CLTV   1.176609
8                               Gender_Female        inf
9                                 Gender_Male        inf
10          Multiple Lines_No phone service   2.339706
11                        Multiple Lines_Yes   1.580949
12                       Internet Service_No        inf
13                        Online Security_No   1.743672
14      Online Security_No internet service        inf
15        Online Backup_No internet service        inf
16                    Device Protection_No   1.725607
17    Device Protection_No internet service        inf
18                           Tech Support_No   1.796200
19         Tech Support_No internet service        inf
20                           Streaming TV_No   2.089787
21         Streaming TV_No internet service        inf
22                       Streaming Movies_No   2.067509
23     Streaming Movies_No internet service        inf
24                  Contract_Month-to-month        inf
25                        Contract_One year        inf
26                        Contract_Two year        inf
27  Payment Method_Bank transfer (automatic)        inf
28    Payment Method_Credit card (automatic)        inf
29           Payment Method_Electronic check        inf
30               Payment Method_Mailed check        inf
```

- Several variables have a VIF of an infinite value, so those variables will be dropped.

In [55]:
```
# Drop columns with infinite VIF
drops = ['Gender_Female', 'Gender_Male', 'Internet Service_No', 'Online Security_No internet service',
         'Online Backup_No internet service', 'Device Protection_No internet service', 'Tech Support_No intern
et service',
         'Streaming TV_No internet service', 'Streaming Movies_No internet service', 'Contract_Month-to-month'
,
         'Contract_One year', 'Contract_One year', 'Contract_Two year', 'Payment Method_Bank transfer (automat
ic)',
         'Payment Method_Credit card (automatic)', 'Payment Method_Electronic check', 'Payment Method_Mailed c
heck']
Xfin = Xfin.drop(drops, axis = 1)
```

In [56]:
```
# Re-run model
Xcfin = sm.add_constant(Xfin)
linear_regression = sm.OLS(y,Xcfin)
fitted_model2 = linear_regression.fit()
fitted_model2.summary()
```

Out[56]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.839
Model:	OLS	**Adj. R-squared:**	0.839
Method:	Least Squares	**F-statistic:**	2616.
Date:	Tue, 04 May 2021	**Prob (F-statistic):**	0.00
Time:	18:36:56	**Log-Likelihood:**	-26097.
No. Observations:	7043	**AIC:**	5.222e+04
Df Residuals:	7028	**BIC:**	5.233e+04
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Count	23.4624	0.581	40.362	0.000	22.323	24.602
Senior Citizen	0.5928	0.333	1.780	0.075	-0.060	1.246
Partner	3.3252	0.251	13.223	0.000	2.832	3.818
Paperless Billing	0.1552	0.261	0.594	0.553	-0.357	0.667
Monthly Charges	-0.3791	0.007	-50.756	0.000	-0.394	-0.364
Total Charges	0.0114	9.61e-05	118.411	0.000	0.011	0.012
Churn Label	-1.9962	0.309	-6.466	0.000	-2.601	-1.391
CLTV	0.0012	0.000	11.563	0.000	0.001	0.001
Multiple Lines_No phone service	1.2200	0.438	2.785	0.005	0.361	2.079
Multiple Lines_Yes	2.6502	0.288	9.212	0.000	2.086	3.214
Online Security_No	-0.2669	0.299	-0.892	0.373	-0.854	0.320
Device Protection_No	-0.2928	0.295	-0.993	0.321	-0.871	0.285
Tech Support_No	0.0013	0.303	0.004	0.997	-0.592	0.595
Streaming TV_No	-0.2654	0.298	-0.889	0.374	-0.851	0.320
Streaming Movies_No	-0.1838	0.300	-0.612	0.540	-0.772	0.404

Omnibus:	183.842	**Durbin-Watson:**	2.017
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	266.618
Skew:	0.281	**Prob(JB):**	1.27e-58
Kurtosis:	3.771	**Cond. No.**	2.68e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.68e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

- Several variables now have high p-values and will be dropped. I will then re-run the model and VIF analysis.

In [57]:
```
drops = ['Senior Citizen', 'Paperless Billing', 'Online Security_No', 'Device Protection_No', 'Tech Support_No',
         'Streaming TV_No', 'Streaming Movies_No']
Xfin = Xfin.drop(drops, axis = 1)
```

In [58]:
```
# Re-run model
Xcfin = sm.add_constant(Xfin)
linear_regression = sm.OLS(y,Xcfin)
fitted_model2 = linear_regression.fit()
fitted_model2.summary()
```

Out[58]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.839
Model:	OLS	**Adj. R-squared:**	0.839
Method:	Least Squares	**F-statistic:**	5229.
Date:	Tue, 04 May 2021	**Prob (F-statistic):**	0.00
Time:	18:40:30	**Log-Likelihood:**	-26102.
No. Observations:	7043	**AIC:**	5.222e+04
Df Residuals:	7035	**BIC:**	5.227e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Count	23.3509	0.574	40.714	0.000	22.227	24.475
Partner	3.3611	0.251	13.384	0.000	2.869	3.853
Monthly Charges	-0.3843	0.006	-62.950	0.000	-0.396	-0.372
Total Charges	0.0115	8.62e-05	133.077	0.000	0.011	0.012
Churn Label	-1.9788	0.303	-6.538	0.000	-2.572	-1.385
CLTV	0.0012	0.000	11.528	0.000	0.001	0.001
Multiple Lines_No phone service	1.0207	0.421	2.422	0.015	0.195	1.847
Multiple Lines_Yes	2.6857	0.287	9.360	0.000	2.123	3.248

Omnibus:	204.815	**Durbin-Watson:**	2.017
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	290.827
Skew:	0.312	**Prob(JB):**	7.04e-64
Kurtosis:	3.776	**Cond. No.**	2.64e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.64e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

In [59]:
```
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = Xfin.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(Xfin.values, i)
                          for i in range(len(Xfin.columns))]

print(vif_data)
```

```
                          feature        VIF
0                           Count  23.867979
1                         Partner   1.142685
2                 Monthly Charges   2.448792
3                   Total Charges   2.765469
4                     Churn Label   1.295813
5                            CLTV   1.167854
6  Multiple Lines_No phone service   1.126745
7                Multiple Lines_Yes   1.457012
```

- I will drop the variable with a high VIF, then re-run the model.

In [60]:
```
drops = ['Count']
Xfin = Xfin.drop(drops, axis = 1)
```

In [61]:
```
# Re-run model
Xcfin = sm.add_constant(Xfin)
linear_regression = sm.OLS(y,Xcfin)
fitted_model2 = linear_regression.fit()
fitted_model2.summary()
```

Out[61]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.839
Model:	OLS	**Adj. R-squared:**	0.839
Method:	Least Squares	**F-statistic:**	5229.
Date:	Tue, 04 May 2021	**Prob (F-statistic):**	0.00
Time:	18:42:33	**Log-Likelihood:**	-26102.
No. Observations:	7043	**AIC:**	5.222e+04
Df Residuals:	7035	**BIC:**	5.227e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	23.3509	0.574	40.714	0.000	22.227	24.475
Partner	3.3611	0.251	13.384	0.000	2.869	3.853
Monthly Charges	-0.3843	0.006	-62.950	0.000	-0.396	-0.372
Total Charges	0.0115	8.62e-05	133.077	0.000	0.011	0.012
Churn Label	-1.9788	0.303	-6.538	0.000	-2.572	-1.385
CLTV	0.0012	0.000	11.528	0.000	0.001	0.001
Multiple Lines_No phone service	1.0207	0.421	2.422	0.015	0.195	1.847
Multiple Lines_Yes	2.6857	0.287	9.360	0.000	2.123	3.248

Omnibus:	204.815	**Durbin-Watson:**	2.017
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	290.827
Skew:	0.312	**Prob(JB):**	7.04e-64
Kurtosis:	3.776	**Cond. No.**	2.64e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.64e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

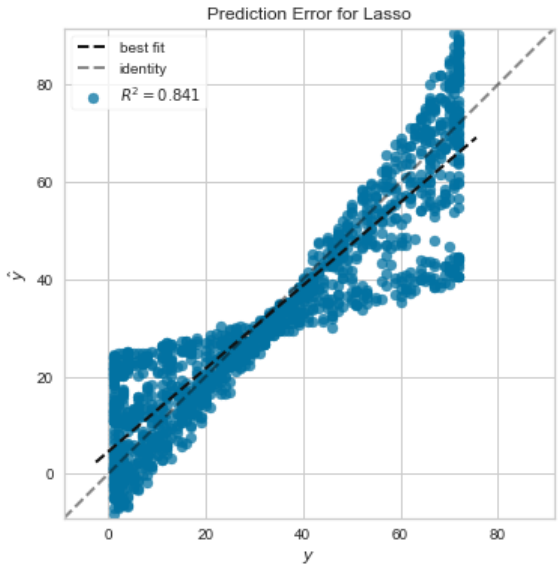
Model Comparison

The initial model of all numeric variables in the provided data set contained 67 independent variables. The inclusion of so many variables inflated the R squared value, which can describe how accurate the model is, resulting in the perfect score of 1.0. Variables were removed first by considering their significance, by utilizing the calculation of their p-value from the model, and by recalculating the model after each deletion to again assess the p-value. Once the insignificant variables were eliminated, the focus was then turned to instances of multicollinearity, as variables that behave similarly can be problematic for predictions as they mean that certain variables are not truly independent of each other. To identify those variables, I used a tool that tested variance inflation factors (VIF) which can measure how much each variable is being affected by other variables (Bhandari, 2020). As with p-values, lower scores are desirable for VIF and should ideally be under 5. In the tutorials that I read, variables with similar scores were related and models improved when removing one of each related pair of variables. I did not see cases in my dataset where VIF scores were similar to be able to decide between two variables. Several variables came back with a score of "inf" (infinite), therefore I chose to remove all of those variables for my final model since I couldn't determine which of those variables might be pairs that I could keep one of. In my final model, there were seven independent variables, meaning 60 were removed.

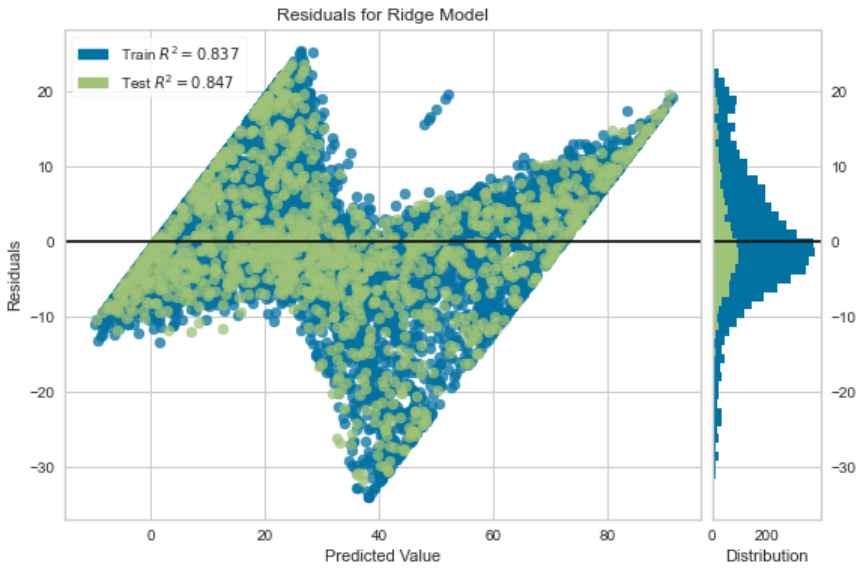
My final model had an R squared value of 0.839. This means that the regression model explains approximately 83.9% of the variation in the sample values of the number of months of a customer's tenure. This value was slightly lower in the final model than the initial model, however, the final equation would be a better determination of how well the model would perform when answering the research question.

The residuals are plotted below. The first plot, or PredictionError plot, compares the actual values from the dataset against the predicted values from by the model to see how much variance is in the model. The plotted line (labeled "best fit") can then be compared against the 45-degree line (labeled "identity"), where the prediction exactly matches the model (McIntyre, 2018). The difference between the two is fairly small, meaning that there is evidence that the final model is reliable. The second residuals plot shows the error of prediction. An ideal residuals plot would have points randomly distributed on both sides of the axis, and this plot is mostly random. Also, this plot includes a histogram of the error values on the right side which ideally is normally distributed around zero. The plotted histogram is nearly a perfect normal distribution curve, which is more evidence of the accuracy of the model.

In [63]:
```
# Display prediction error plot to determine the amount of variance in model (ref 6)
mpl.rcParams['figure.figsize'] = (9,6)

# Create the train and test data
X_train, X_test, y_train, y_test = train_test_split(Xfin, y, test_size=0.2, random_state=42)

model = Lasso()
visualizer = PredictionError(model)

visualizer.fit(X_train, y_train)  # Fit the training data to the visualizer
visualizer.score(X_test, y_test)  # Evaluate the model on the test data
g = visualizer.poof()             # Draw/show/poof the data
```



In [64]:
```
# Display residuals plot (ref 6)
model = Ridge()
visualizer = ResidualsPlot(model)

visualizer.fit(X_train, y_train)  # Fit the training data to the visualizer
visualizer.score(X_test, y_test)  # Evaluate the model on the test data
g = visualizer.poof()             # Draw/show/poof the data
```



In [65]:
```
# Create dataframe of coefficients for regression equation (ref 7)
regressor = LinearRegression()
regressor.fit(Xfin, y)
coeff_df = pd.DataFrame(regressor.coef_.round(decimals = 2), Xfin.columns, columns = ['Coefficient'])
coeff_df['Variable'] = coeff_df.index

# Print regression equation
equation = 'y = '
y1 = regressor.intercept_.round(decimals = 2)
equation += str(y1)
for index, row in coeff_df.iterrows():
    if row['Coefficient'] < 0:
        c = str(row.Coefficient)
        equation += c + '(' + row['Variable'] + ')'
    else:
        c = str(row.Coefficient)
        equation += '+' + c + '(' + row['Variable'] + ')'
print(equation)
```

```
y = 23.35+3.36(Partner)-0.38(Monthly Charges)+0.01(Total Charges)-1.98(Churn Label)+0.0(CLTV)+1.02(Multiple L
ines_No phone service)+2.69(Multiple Lines_Yes)
```

Results

The equation of the final regression model is: y = 23.35 + 3.36(Partner) - 0.38(Monthly Charges) + 0.01(Total Charges) - 1.98(Churn Label) + 0.0(CLTV) + 1.02(Multiple Lines_No phone service) + 2.69(Multiple Lines_Yes). In this equation, 23.35 is the value of Tenure when all of the predictor variables are zero. Each coefficient of the predictor variables describes how much the target variable is estimated to change if all other variables remain constant. For example, if Multiple Lines_Yes increased by one unit, the mean value of Tenure would increase by 2.69. Beyond that Partner seems to have the largest effect on tenure positively by 3.36.

Sources

Bhandari, A. (2020). Multicollinearity | Detecting Multicollinearity with VIF. Analytics Vidhya. Retrieved 8 January 2021, from https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/ (https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/). Massaron, L., & Boschetti, A. (2016). Regression analysis with Python. Packt Publishing. ISBN: 9781785286315 McIntyre, K. (2018, August 18). Yellowbrick - Regression Visualizer Examples. Retrieved January 08, 2021, from https://www.kaggle.com/kautumn06/yellowbrick-regression-visualizer-examples (https://www.kaggle.com/kautumn06/yellowbrick-regression-visualizer-examples)

Helpful Sites Used in Coding Project

1. https://stackoverflow.com/questions/51672709/converting-no-and-yes-into-0-and-1-in-pandas-dataframe/51672855 (https://stackoverflow.com/questions/51672709/converting-no-and-yes-into-0-and-1-in-pandas-dataframe/51672855)
2. https://stackoverflow.com/questions/61526812/boxplot-for-all-data-in-dataframe-error-numpy-ndarray-object-has-no-attribut (https://stackoverflow.com/questions/61526812/boxplot-for-all-data-in-dataframe-error-numpy-ndarray-object-has-no-attribut)
3. https://datascience.stackexchange.com/questions/84840/how-to-create-multiple-subplots-scatterplot-in-for-loop (https://datascience.stackexchange.com/questions/84840/how-to-create-multiple-subplots-scatterplot-in-for-loop)
4. https://stackoverflow.com/questions/31290778/add-custom-border-to-certain-cells-in-a-matplotlib-seaborn-plot (https://stackoverflow.com/questions/31290778/add-custom-border-to-certain-cells-in-a-matplotlib-seaborn-plot)
5. https://analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/ (https://analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/)
6. https://www.kaggle.com/kautumn06/yellowbrick-regression-visualizer-examples (https://www.kaggle.com/kautumn06/yellowbrick-regression-visualizer-examples)
7. https://www.kdnuggets.com/2019/03/beginners-guide-linear-regression-python-scikit-learn.html/2 (https://www.kdnuggets.com/2019/03/beginners-guide-linear-regression-python-scikit-learn.html/2)
8. https://stackoverflow.com/questions/59740434/how-to-print-intercept-and-slope-of-a-simple-linear-regression-in-python-with-sc (https://stackoverflow.com/questions/59740434/how-to-print-intercept-and-slope-of-a-simple-linear-regression-in-python-with-sc)