A Quick Overview

1. Collect relevant customer data.
2. Preprocess and clean the data.
3. Engineer meaningful features.
4. Analyze data for trends and predictors.
5. Select appropriate machine learning algorithm.
6. Split data into train and test sets.
7. Train the chosen model.
8. Evaluate model performance.
9. Make predictions and interpret results.
10. Deploy and monitor the churn prediction model

In [13]:
```
import pandas as pd
Telecom_churn=pd.read_csv("/Users/asifsiraz/Desktop/telecom_custome
```

In [14]:
```
Telecom_churn
```

Out[14]:

ity	Zip Code	Latitude	Longitude	Number of Referrals	...	Payment Method	Monthly Charge	Total Charges	Total Refunds	Ch
ier ark	93225	34.827662	-118.999073	2	...	Credit Card	65.60	593.30	0.00	
ale	91206	34.162515	-118.203869	0	...	Credit Card	-4.00	542.40	38.33	
sta esa	92627	33.645672	-117.922613	0	...	Bank Withdrawal	73.90	280.85	0.00	
iez	94553	38.014457	-122.115432	1	...	Bank Withdrawal	98.00	1237.85	0.00	
illo	93010	34.227846	-119.079903	3	...	Credit Card	83.90	267.40	0.00	
...	...	...	...	...	...	...	...	...	...	
esa	91941	32.759327	-116.997260	0	...	Credit Card	55.15	742.90	0.00	
ink	95367	37.734971	-120.954271	1	...	Bank Withdrawal	85.10	1873.70	0.00	
Elk	95432	39.108252	-123.645121	0	...	Credit Card	50.30	92.75	0.00	
ina ich	92075	33.001813	-117.263628	5	...	Credit Card	67.85	4627.65	0.00	
rra City	96125	39.600599	-120.636358	1	...	Bank Withdrawal	59.00	3707.60	0.00	

In [15]:
```
Telecom_churn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 38 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Customer ID                        7043 non-null   object 
 1   Gender                             7043 non-null   object 
 2   Age                                7043 non-null   int64  
 3   Married                            7043 non-null   object 
 4   Number of Dependents               7043 non-null   int64  
 5   City                               7043 non-null   object 
 6   Zip Code                           7043 non-null   int64  
 7   Latitude                           7043 non-null   float64
 8   Longitude                          7043 non-null   float64
 9   Number of Referrals                7043 non-null   int64  
 10  Tenure in Months                   7043 non-null   int64  
 11  Offer                              7043 non-null   object 
 12  Phone Service                      7043 non-null   object 
 13  Avg Monthly Long Distance Charges  6361 non-null   float64
 14  Multiple Lines                     6361 non-null   object 
 15  Internet Service                   7043 non-null   object 
 16  Internet Type                      5517 non-null   object 
 17  Avg Monthly GB Download            5517 non-null   float64
 18  Online Security                    5517 non-null   object 
 19  Online Backup                      5517 non-null   object 
 20  Device Protection Plan             5517 non-null   object 
 21  Premium Tech Support               5517 non-null   object 
 22  Streaming TV                       5517 non-null   object 
 23  Streaming Movies                   5517 non-null   object 
 24  Streaming Music                    5517 non-null   object 
 25  Unlimited Data                     5517 non-null   object 
 26  Contract                           7043 non-null   object 
 27  Paperless Billing                  7043 non-null   object 
 28  Payment Method                     7043 non-null   object 
 29  Monthly Charge                     7043 non-null   float64
 30  Total Charges                      7043 non-null   float64
 31  Total Refunds                      7043 non-null   float64
 32  Total Extra Data Charges           7043 non-null   int64  
 33  Total Long Distance Charges        7043 non-null   float64
 34  Total Revenue                      7043 non-null   float64
 35  Customer Status                    7043 non-null   object 
 36  Churn Category                     1869 non-null   object 
 37  Churn Reason                       1869 non-null   object 
dtypes: float64(9), int64(6), object(23)
memory usage: 2.0+ MB
```

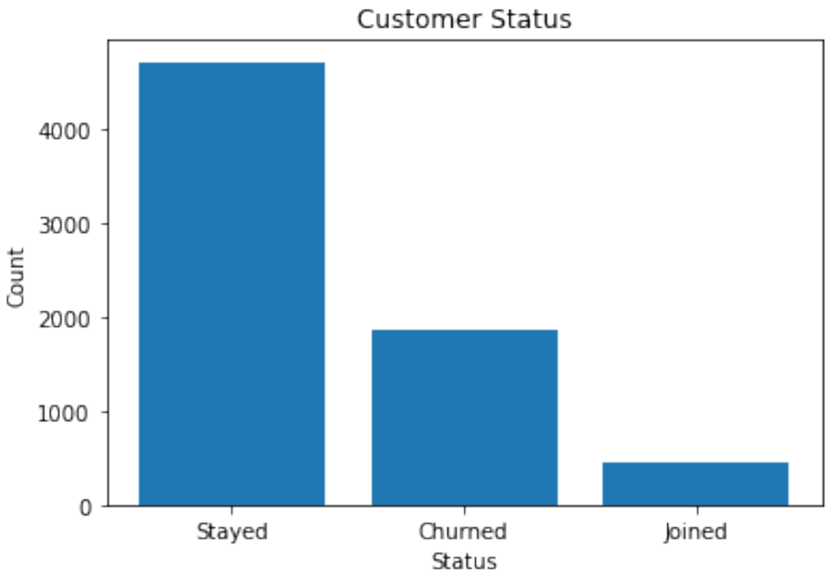
In [16]:
```
Telecom_churn.isna().sum()
```

Out[16]:
```
Customer ID                            0
Gender                                 0
Age                                    0
Married                                0
Number of Dependents                   0
City                                   0
Zip Code                               0
Latitude                               0
Longitude                              0
Number of Referrals                    0
Tenure in Months                       0
Offer                                  0
Phone Service                          0
Avg Monthly Long Distance Charges    682
Multiple Lines                       682
Internet Service                       0
Internet Type                       1526
Avg Monthly GB Download             1526
Online Security                     1526
Online Backup                       1526
Device Protection Plan              1526
Premium Tech Support                1526
Streaming TV                        1526
Streaming Movies                    1526
Streaming Music                     1526
Unlimited Data                      1526
Contract                               0
Paperless Billing                      0
Payment Method                         0
Monthly Charge                         0
Total Charges                          0
Total Refunds                          0
Total Extra Data Charges               0
Total Long Distance Charges            0
Total Revenue                          0
Customer Status                        0
Churn Category                      5174
Churn Reason                        5174
dtype: int64
```

In [17]:
```
Telecom_churn["Customer Status"].value_counts()
```

Out[17]:
```
Stayed     4720
Churned    1869
Joined      454
Name: Customer Status, dtype: int64
```

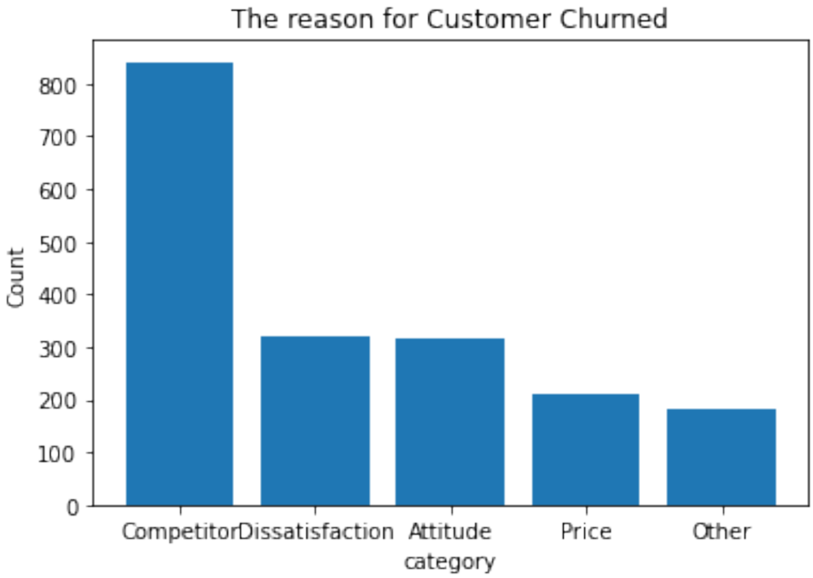
In [18]:
```
import matplotlib.pyplot as plt


status_counts = Telecom_churn["Customer Status"].value_counts()
plt.bar(status_counts.index, status_counts.values)


plt.title("Customer Status")
plt.xlabel("Status")
plt.ylabel("Count")
plt.xticks()
plt.show()
```



In [19]:
```
Telecom_churn["Churn Category"].value_counts()
```

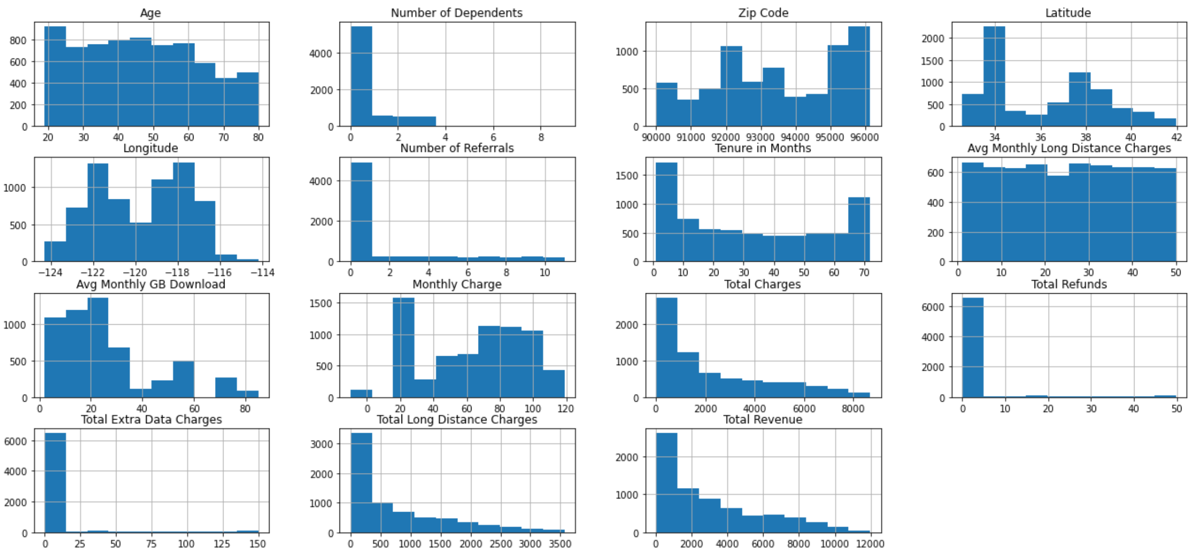
Out[19]:
```
Competitor         841
Dissatisfaction    321
Attitude           314
Price              211
Other              182
Name: Churn Category, dtype: int64
```

In [20]:
```
import matplotlib.pyplot as plt


status_counts = Telecom_churn["Churn Category"].value_counts()
plt.bar(status_counts.index, status_counts.values)


plt.title("The reason for Customer Churned")
plt.xlabel("category")
plt.ylabel("Count")
plt.xticks()
plt.show()
```



In [21]:
```
Telecom_churn.hist(figsize=(22,10))
```

Out[21]:
```
array([[<AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'Number of Dependents'}>,
        <AxesSubplot:title={'center':'Zip Code'}>,
        <AxesSubplot:title={'center':'Latitude'}>],
       [<AxesSubplot:title={'center':'Longitude'}>,
        <AxesSubplot:title={'center':'Number of Referrals'}>,
        <AxesSubplot:title={'center':'Tenure in Months'}>,
        <AxesSubplot:title={'center':'Avg Monthly Long Distance Ch
arges'}>],
       [<AxesSubplot:title={'center':'Avg Monthly GB Download'}>,
        <AxesSubplot:title={'center':'Monthly Charge'}>,
        <AxesSubplot:title={'center':'Total Charges'}>,
        <AxesSubplot:title={'center':'Total Refunds'}>],
       [<AxesSubplot:title={'center':'Total Extra Data Charges'}>,
        <AxesSubplot:title={'center':'Total Long Distance Charges'
}>,
        <AxesSubplot:title={'center':'Total Revenue'}>, <AxesSubpl
ot:>]],
      dtype=object)
```



In [22]:
```
Telecom_churn=Telecom_churn.dropna(axis=1)
Telecom_churn
```

Out[22]:

City	Zip Code	Latitude	Longitude	Number of Referrals	...	Contract	Paperless Billing	Payment Method	M C
Frazier Park	93225	34.827662	-118.999073	2	...	One Year	Yes	Credit Card	
Glendale	91206	34.162515	-118.203869	0	...	Month-to-Month	No	Credit Card	
Costa Mesa	92627	33.645672	-117.922613	0	...	Month-to-Month	Yes	Bank Withdrawal	
Martinez	94553	38.014457	-122.115432	1	...	Month-to-Month	Yes	Bank Withdrawal	
Camarillo	93010	34.227846	-119.079903	3	...	Month-to-Month	Yes	Credit Card	
...	...	...	...	...	...	...	...	...	
La Mesa	91941	32.759327	-116.997260	0	...	One Year	No	Credit Card	
Riverbank	95367	37.734971	-120.954271	1	...	Month-to-Month	Yes	Bank Withdrawal	
Elk	95432	39.108252	-123.645121	0	...	Month-to-Month	Yes	Credit Card	
Solana Beach	92075	33.001813	-117.263628	5	...	Two Year	No	Credit Card	
Sierra City	96125	39.600599	-120.636358	1	...	Two Year	No	Bank Withdrawal	

In [23]:
```
Telecom_churn.isna().sum()
```

Out[23]:
```
Customer ID                      0
Gender                           0
Age                              0
Married                          0
Number of Dependents             0
City                             0
Zip Code                         0
Latitude                         0
Longitude                        0
Number of Referrals              0
Tenure in Months                 0
Offer                            0
Phone Service                    0
Internet Service                 0
Contract                         0
Paperless Billing                0
Payment Method                   0
Monthly Charge                   0
Total Charges                    0
Total Refunds                    0
Total Extra Data Charges         0
Total Long Distance Charges      0
Total Revenue                    0
Customer Status                  0
dtype: int64
```

In [24]:
```
Telecom_churn.describe()
```

Out[24]:

	Age	Number of Dependents	Zip Code	Latitude	Longitude	Number of Referrals	
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	704
mean	46.509726	0.468692	93486.070567	36.197455	-119.756684	1.951867	3
std	16.750352	0.962802	1856.767505	2.468929	2.154425	3.001199	2
min	19.000000	0.000000	90001.000000	32.555828	-124.301372	0.000000	
25%	32.000000	0.000000	92101.000000	33.990646	-121.788090	0.000000	
50%	46.000000	0.000000	93518.000000	36.205465	-119.595293	0.000000	2
75%	60.000000	0.000000	95329.000000	38.161321	-117.969795	3.000000	5
max	80.000000	9.000000	96150.000000	41.962127	-114.192901	11.000000	7

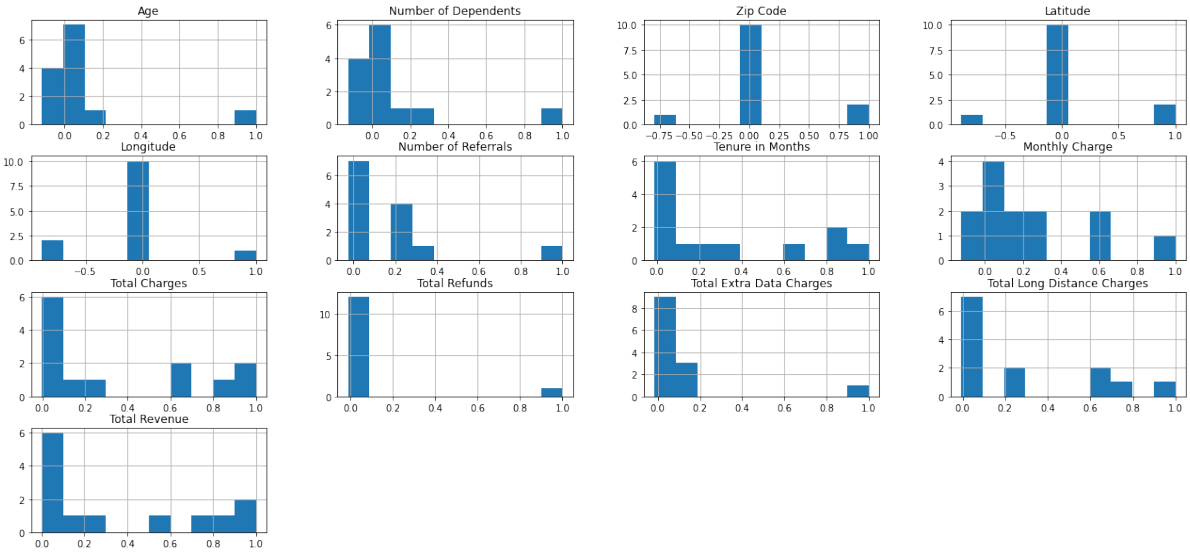
In [25]:
```
Telecom_churn.corr()
```

Out[25]:

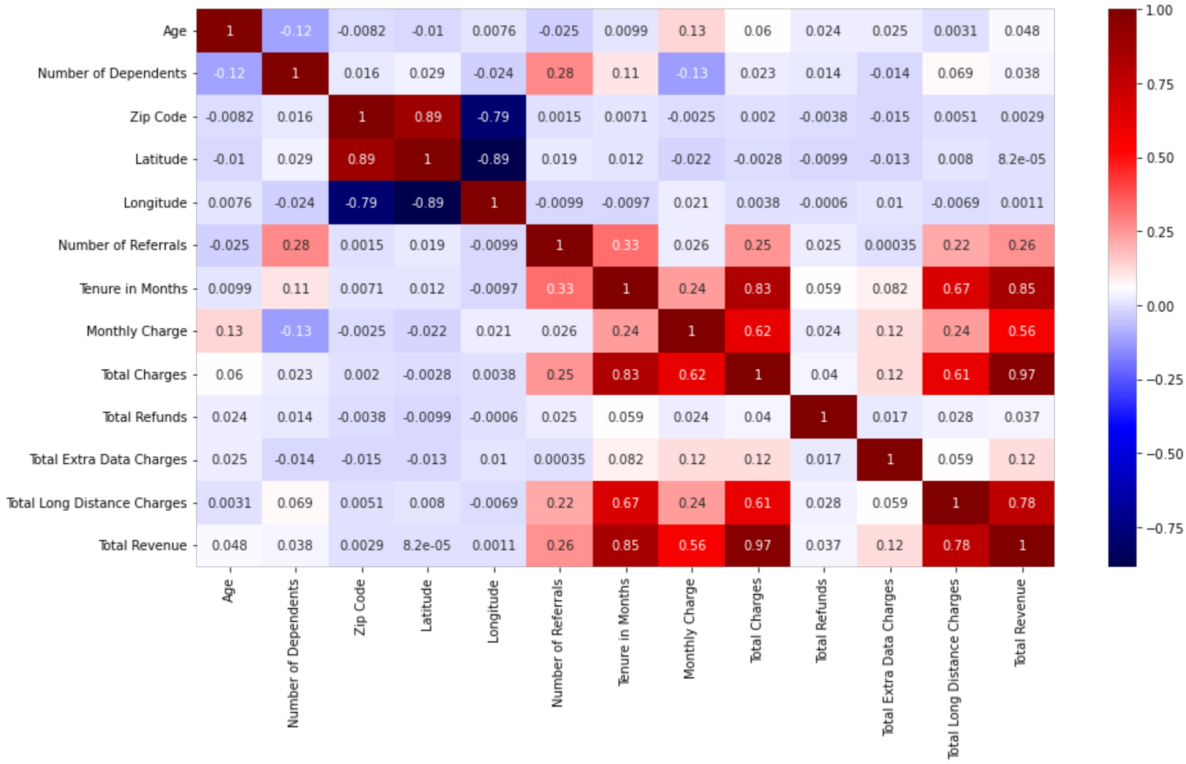
	Age	Number of Dependents	Zip Code	Latitude	Longitude	Number of Referrals	Tenure in Months
Age	1.000000	-0.119000	-0.008183	-0.010305	0.007612	-0.025141	0.009927
Number of Dependents	-0.119000	1.000000	0.016493	0.029081	-0.024271	0.278003	0.108237
Zip Code	-0.008183	0.016493	1.000000	0.894769	-0.790564	0.001463	0.007146
Latitude	-0.010305	0.029081	0.894769	1.000000	-0.885979	0.018715	0.011963
Longitude	0.007612	-0.024271	-0.790564	-0.885979	1.000000	-0.009893	-0.009672
Number of Referrals	-0.025141	0.278003	0.001463	0.018715	-0.009893	1.000000	0.326975
Tenure in Months	0.009927	0.108237	0.007146	0.011963	-0.009672	0.326975	1.000000
Monthly Charge	0.134511	-0.125649	-0.002517	-0.021613	0.021052	0.026301	0.239065
Total Charges	0.059684	0.022535	0.001978	-0.002784	0.003811	0.250378	0.826074
Total Refunds	0.024168	0.014023	-0.003797	-0.009901	-0.000597	0.024756	0.059021
Total Extra Data Charges	0.025036	-0.014436	-0.014550	-0.013233	0.010461	0.000350	0.082266
Total Long Distance Charges	0.003065	0.068966	0.005063	0.008029	-0.006923	0.216190	0.674149
Total Revenue	0.048265	0.038038	0.002944	0.000082	0.001062	0.261853	0.853146

In [26]:
```
Telecom_churn.corr().hist(figsize=(22,10))
```

Out[26]:
```
array([[<AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'Number of Dependents'}>,
        <AxesSubplot:title={'center':'Zip Code'}>,
        <AxesSubplot:title={'center':'Latitude'}>],
       [<AxesSubplot:title={'center':'Longitude'}>,
        <AxesSubplot:title={'center':'Number of Referrals'}>,
        <AxesSubplot:title={'center':'Tenure in Months'}>,
        <AxesSubplot:title={'center':'Monthly Charge'}>],
       [<AxesSubplot:title={'center':'Total Charges'}>,
        <AxesSubplot:title={'center':'Total Refunds'}>,
        <AxesSubplot:title={'center':'Total Extra Data Charges'}>,
        <AxesSubplot:title={'center':'Total Long Distance Charges'
}>],
       [<AxesSubplot:title={'center':'Total Revenue'}>, <AxesSubpl
ot:>,
        <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```

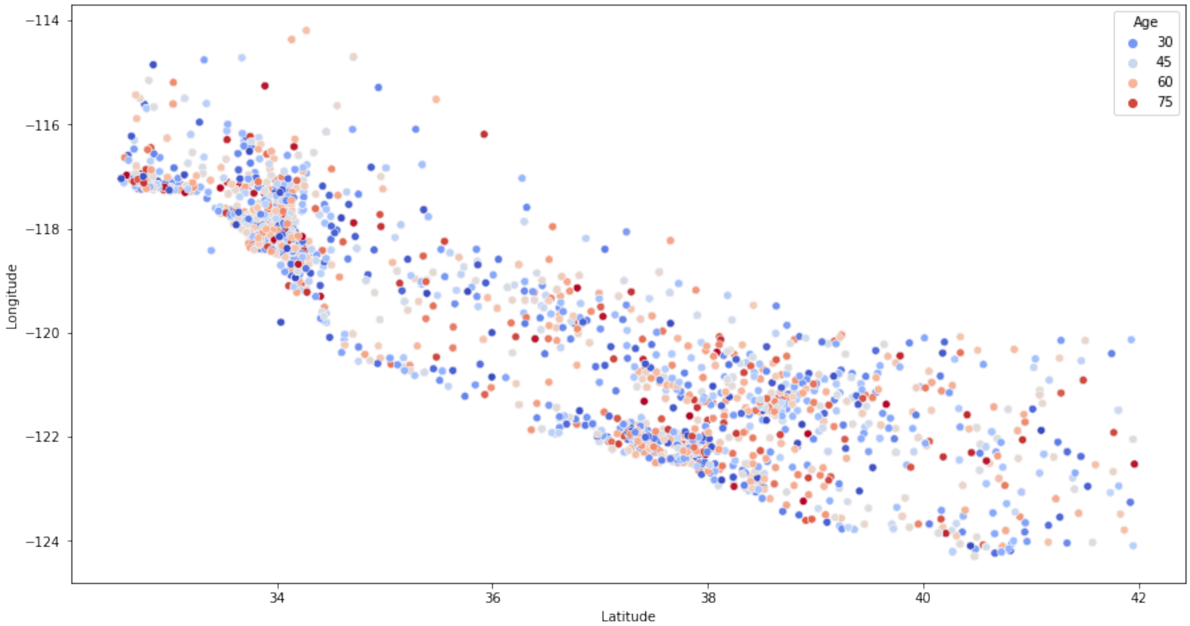


In [27]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(15,8))
sns.heatmap(Telecom_churn.corr(),annot=True,cmap="seismic")
plt.show()
```



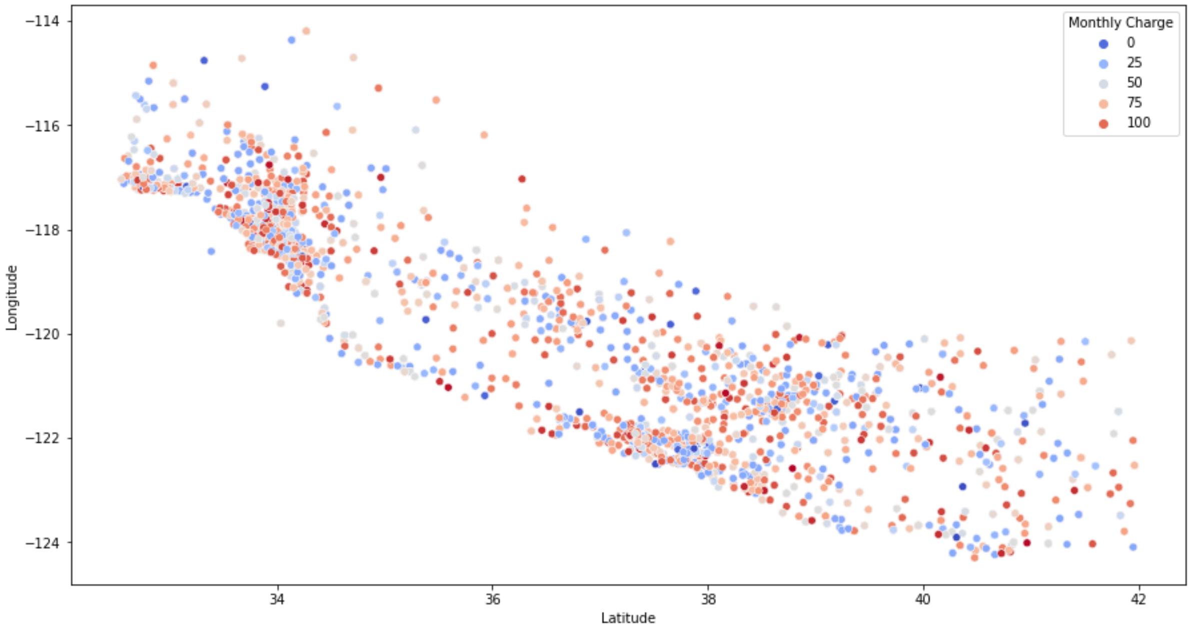
In [28]:
```
plt.figure(figsize=(15,8))
sns.scatterplot(x="Latitude",y="Longitude",data=Telecom_churn,hue="
```

Out[28]: <AxesSubplot:xlabel='Latitude', ylabel='Longitude'>



In [29]:
```
plt.figure(figsize=(15,8))
sns.scatterplot(x="Latitude",y="Longitude",data=Telecom_churn,hue="
```

Out[29]: <AxesSubplot:xlabel='Latitude', ylabel='Longitude'>



In [30]:
```
from sklearn.model_selection import train_test_split
```

In [31]:
```
Telecom_churn=pd.get_dummies(Telecom_churn)
Telecom_churn
```

Out[31]:

	Age	Number of Dependents	Zip Code	Latitude	Longitude	Number of Referrals	Tenure in Months	Monthly Charge	To Charg
0	37	0	93225	34.827662	-118.999073	2	9	65.60	593.
1	46	0	91206	34.162515	-118.203869	0	9	-4.00	542.
2	50	0	92627	33.645672	-117.922613	0	4	73.90	280.
3	78	0	94553	38.014457	-122.115432	1	13	98.00	1237.
4	75	0	93010	34.227846	-119.079903	3	3	83.90	267.
...	...	...	...	...	...	...	...	...	
7038	20	0	91941	32.759327	-116.997260	0	13	55.15	742.
7039	40	0	95367	37.734971	-120.954271	1	22	85.10	1873.
7040	22	0	95432	39.108252	-123.645121	0	2	50.30	92.
7041	21	0	92075	33.001813	-117.263628	5	67	67.85	4627.
7042	36	0	96125	39.600599	-120.636358	1	63	59.00	3707.

7043 rows × 8187 columns

In [32]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# Split the features and target variables
x=Telecom_churn.drop(["Customer Status_Stayed"],axis=1)
y=Telecom_churn["Customer Status_Stayed"]

# Split the data into training and test sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size

# Create a Random Forest Classifier
rf = RandomForestClassifier(n_estimators=150, max_depth=None, max_f

# Train the classifier on the training data
rf.fit(x_train, y_train)

# Evaluate the accuracy on the entire dataset
accuracy = rf.score(x, y)
print("Accuracy on the entire dataset:", accuracy)

# Evaluate the accuracy on the test set
test_accuracy = rf.score(x_test, y_test)
print("Accuracy on the test set:", test_accuracy)

# Evaluate the accuracy on the training set
train_accuracy = rf.score(x_train, y_train)
print("Accuracy on the training set:", train_accuracy)
```

```
Accuracy on the entire dataset: 0.9929007525202329
Accuracy on the test set: 0.964513839602555
Accuracy on the training set: 1.0
```

In [33]:
```
y_pred=rf.predict(x_train)
```

In [34]:
```
from sklearn import metrics
import numpy as np
print('R^2:',metrics.r2_score(y_train,y_pred))
print('MAE:',metrics.mean_absolute_error(y_train, y_pred))
print('MSE:',metrics.mean_squared_error(y_train, y_pred))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2: 1.0
MAE: 0.0
MSE: 0.0
RMSE: 0.0
```

In [35]:
```
# Split the features and target variables
x = Telecom_churn.drop(["Customer Status_Stayed"], axis=1)
y = Telecom_churn["Customer Status_Stayed"]

# Split the data into training and test sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size

# Create a Random Forest Classifier
rf = RandomForestClassifier(n_estimators=150, max_depth=None, max_f

# Train the classifier on the training data
rf.fit(x_train, y_train)

# Calculate feature importances
importances = rf.feature_importances_

# Get the feature names
feature_names = x.columns

# Create a DataFrame to store feature importance scores
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Im

# Sort the features by importance in descending order
feature_importance_df = feature_importance_df.sort_values(by='Impor

# Display the feature importance scores
feature_importance_df.head(10)
```

Out[35]:

	Feature	Importance
8184	Customer Status_Churned	0.154526
6	Tenure in Months	0.071887
12	Total Revenue	0.058611
8	Total Charges	0.052277
11	Total Long Distance Charges	0.048051
8176	Contract_Month-to-Month	0.040976
8185	Customer Status_Joined	0.024122
5	Number of Referrals	0.023147
7	Monthly Charge	0.022592
8178	Contract_Two Year	0.022006