In [39]:

```
#About this file ******https://www.kaggle.com/radmirzosimov/telecom-users-dataset*******
#customerID - customer id
#gender - client gender (male / female)
#SeniorCitizen - is the client retired (1, 0)
#Partner - is the client married (Yes, No)
#tenure - how many months a person has been a client of the company
#PhoneService - is the telephone service connected (Yes, No)
#MultipleLines - are multiple phone lines connected (Yes, No, No phone service)
#InternetService - client's Internet service provider (DSL, Fiber optic, No)
#OnlineSecurity - is the online security service connected (Yes, No, No internet service)
#OnlineBackup - is the online backup service activated (Yes, No, No internet service)
#DeviceProtection - does the client have equipment insurance (Yes, No, No internet service)
#TechSupport - is the technical support service connected (Yes, No, No internet service)
#StreamingTV - is the streaming TV service connected (Yes, No, No internet service)
#StreamingMovies - is the streaming cinema service activated (Yes, No, No internet service)
#Contract - type of customer contract (Month-to-month, One year, Two year)
#PaperlessBilling - whether the client uses paperless billing (Yes, No)
#PaymentMethod - payment method (Electronic check, Mailed check, Bank transfer (automatic),
#MonthlyCharges - current monthly payment
#TotalCharges - the total amount that the client paid for the services for the entire time
#Churn - whether there was a churn (Yes or No)
```

In [ ]:

```
#Logistic Regression
```

In [ ]:

```
#Read Data
```

In [ ]:

```
import pandas as pd
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
```

In [40]:

```
data=pd.read_csv('telecom_users.csv')
```

In [41]:

```
data.isnull().sum()
```

Out[41]:

```
Unnamed: 0          0
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

In [42]:

```
data.head()
```

Out[42]:

	Unnamed: 0	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	I
0	1869	7010-BRBUU	Male	0	Yes	Yes	72	Yes	
1	4528	9688-YGXVR	Female	0	No	No	44	Yes	
2	6344	9286-DOJGF	Female	1	Yes	No	38	Yes	
3	6739	6994-KERXL	Male	0	No	No	4	Yes	
4	432	2181-UAESM	Male	0	No	No	2	Yes	

5 rows × 22 columns

In [43]:

```
data['Churn'].value_counts()
```

Out[43]:

```
No     4399
Yes    1587
Name: Churn, dtype: int64
```

In [44]:

```
data.shape
```

Out[44]:

```
(5986, 22)
```

In [45]:

```
data.columns
```

Out[45]:

```
Index(['Unnamed: 0', 'customerID', 'gender', 'SeniorCitizen', 'Partner',
       'Dependents', 'tenure', 'PhoneService', 'MultipleLines',
       'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtectio
n',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharge
s',
       'Churn'],
      dtype='object')
```

In [46]:

```
data.value_counts
```

Out[46]:

```
<bound method DataFrame.value_counts of       Unnamed: 0  customerID  gender  SeniorCitizen Partner Dependents  \
0           1869  7010-BRBUU    Male              0     Yes        Yes
1           4528  9688-YGXVR  Female              0      No         No
2           6344  9286-DOJGF  Female              1     Yes         No
3           6739  6994-KERXL    Male              0      No         No
4            432  2181-UAESM    Male              0      No         No
...          ...         ...     ...            ...     ...        ...
5981        3772  0684-AOSIH    Male              0     Yes         No
5982        5191  5982-PSMKW  Female              0     Yes        Yes
5983        5226  8044-BGWPI    Male              0     Yes        Yes
5984        5390  7450-NWRTR    Male              1      No         No
5985         860  4795-UXVCJ    Male              0      No         No

      tenure PhoneService MultipleLines InternetService  ...  \
0         72          Yes           Yes              No  ...
1         44          Yes            No     Fiber optic  ...
2         38          Yes           Yes     Fiber optic  ...
3          4          Yes            No             DSL  ...
4          2          Yes            No             DSL  ...
...      ...          ...           ...             ...  ...
5981       1          Yes            No     Fiber optic  ...
5982      23          Yes           Yes             DSL  ...
5983      12          Yes            No              No  ...
5984      12          Yes           Yes     Fiber optic  ...
5985      26          Yes            No              No  ...

         DeviceProtection          TechSupport          StreamingTV  \
0     No internet service  No internet service  No internet service
1                     Yes                   No                  Yes
2                      No                   No                   No
3                      No                   No                   No
4                     Yes                   No                   No
...                   ...                  ...                  ...
5981                   No                   No                  Yes
5982                  Yes                  Yes                  Yes
5983  No internet service  No internet service  No internet service
5984                  Yes                   No                  Yes
5985  No internet service  No internet service  No internet service

          StreamingMovies        Contract PaperlessBilling  \
0     No internet service        Two year               No
1                      No  Month-to-month              Yes
2                      No  Month-to-month              Yes
3                     Yes  Month-to-month              Yes
4                      No  Month-to-month               No
...                   ...             ...              ...
5981                  Yes  Month-to-month              Yes
5982                  Yes        Two year              Yes
5983  No internet service  Month-to-month              Yes
5984                  Yes  Month-to-month              Yes
5985  No internet service        One year               No

                PaymentMethod MonthlyCharges TotalCharges  Churn
0     Credit card (automatic)          24.10      1734.65     No
```

```
1       Credit card (automatic)          88.15        3973.2     No
2     Bank transfer (automatic)          74.95       2869.85    Yes
3              Electronic check          55.90         238.5     No
4              Electronic check          53.45         119.5     No
...                         ...            ...           ...    ...
5981           Electronic check          95.00            95    Yes
5982    Credit card (automatic)          91.10        2198.3     No
5983           Electronic check          21.15        306.05     No
5984           Electronic check          99.45       1200.15    Yes
5985    Credit card (automatic)          19.80         457.3     No

[5986 rows x 22 columns]>
```

In [47]:

```
data.describe()
```

Out[47]:

	Unnamed: 0	SeniorCitizen	tenure	MonthlyCharges
count	5986.000000	5986.000000	5986.000000	5986.000000
mean	3533.561310	0.161377	32.468760	64.802213
std	2035.705666	0.367909	24.516391	30.114702
min	0.000000	0.000000	0.000000	18.250000
25%	1777.250000	0.000000	9.000000	35.650000
50%	3546.500000	0.000000	29.000000	70.400000
75%	5291.750000	0.000000	56.000000	89.900000
max	7042.000000	1.000000	72.000000	118.750000

In [48]:

```
data['InternetService'].value_counts()
```

Out[48]:

```
Fiber optic    2627
DSL            2068
No             1291
Name: InternetService, dtype: int64
```

In [49]:

```
data.columns
```

Out[49]:

```
Index(['Unnamed: 0', 'customerID', 'gender', 'SeniorCitizen', 'Partner',
       'Dependents', 'tenure', 'PhoneService', 'MultipleLines',
       'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtectio
n',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharge
s',
       'Churn'],
      dtype='object')
```

In [50]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5986 entries, 0 to 5985
Data columns (total 22 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Unnamed: 0        5986 non-null   int64  
 1   customerID        5986 non-null   object 
 2   gender            5986 non-null   object 
 3   SeniorCitizen     5986 non-null   int64  
 4   Partner           5986 non-null   object 
 5   Dependents        5986 non-null   object 
 6   tenure            5986 non-null   int64  
 7   PhoneService      5986 non-null   object 
 8   MultipleLines     5986 non-null   object 
 9   InternetService   5986 non-null   object 
 10  OnlineSecurity    5986 non-null   object 
 11  OnlineBackup      5986 non-null   object 
 12  DeviceProtection  5986 non-null   object 
 13  TechSupport       5986 non-null   object 
 14  StreamingTV       5986 non-null   object 
 15  StreamingMovies   5986 non-null   object 
 16  Contract          5986 non-null   object 
 17  PaperlessBilling  5986 non-null   object 
 18  PaymentMethod     5986 non-null   object 
 19  MonthlyCharges    5986 non-null   float64
 20  TotalCharges      5986 non-null   object 
 21  Churn             5986 non-null   object 
dtypes: float64(1), int64(3), object(18)
memory usage: 1.0+ MB
```

In [ ]:

```
#Encoding data
```

In [51]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
data['PhoneService']=encoder.fit_transform(data['PhoneService'])
data['MultipleLines']=encoder.fit_transform(data['MultipleLines'])
data['InternetService']=encoder.fit_transform(data['InternetService'])
data['OnlineSecurity']=encoder.fit_transform(data['OnlineSecurity'])
data['OnlineBackup']=encoder.fit_transform(data['OnlineBackup'])
data['DeviceProtection']=encoder.fit_transform(data['DeviceProtection'])
data['InternetService']=encoder.fit_transform(data['InternetService'])
data['TechSupport']=encoder.fit_transform(data['TechSupport'])
data['StreamingTV']=encoder.fit_transform(data['StreamingTV'])
data['StreamingMovies']=encoder.fit_transform(data['StreamingMovies'])
data['PaymentMethod']=encoder.fit_transform(data['PaymentMethod'])
data['Churn']=encoder.fit_transform(data['Churn'])
data['gender']=encoder.fit_transform(data['gender'])
data['Partner']=encoder.fit_transform(data['Partner'])
data['TotalCharges']=encoder.fit_transform(data['TotalCharges'])
data.head()
```

Out[51]:

	Unnamed: 0	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	I
0	1869	7010-BRBUU	1	0	1	Yes	72	1	
1	4528	9688-YGXVR	0	0	0	No	44	1	
2	6344	9286-DOJGF	0	1	1	No	38	1	
3	6739	6994-KERXL	1	0	0	No	4	1	
4	432	2181-UAESM	1	0	0	No	2	1	

5 rows × 22 columns

In [52]:

```
data.describe()
```

Out[52]:

	Unnamed: 0	gender	SeniorCitizen	Partner	tenure	PhoneService	Multip
count	5986.000000	5986.000000	5986.000000	5986.000000	5986.000000	5986.000000	5986
mean	3533.561310	0.509522	0.161377	0.485132	32.468760	0.901437	0
std	2035.705666	0.499951	0.367909	0.499821	24.516391	0.298100	0
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
25%	1777.250000	0.000000	0.000000	0.000000	9.000000	1.000000	0
50%	3546.500000	1.000000	0.000000	0.000000	29.000000	1.000000	1
75%	5291.750000	1.000000	0.000000	1.000000	56.000000	1.000000	2
max	7042.000000	1.000000	1.000000	1.000000	72.000000	1.000000	2

In [ ]:

```
#data correlation
```

In [53]:

```
data.corr()
```

Out[53]:

	Unnamed: 0	gender	SeniorCitizen	Partner	tenure	PhoneService	Multi
Unnamed: 0	1.000000	-0.006931	-0.000331	-0.001498	0.009620	-0.024496	-
gender	-0.006931	1.000000	-0.007447	-0.007791	0.003207	-0.004913	
SeniorCitizen	-0.000331	-0.007447	1.000000	0.014867	0.005468	0.009464	
Partner	-0.001498	-0.007791	0.014867	1.000000	0.381976	0.024926	
tenure	0.009620	0.003207	0.005468	0.381976	1.000000	0.010392	
PhoneService	-0.024496	-0.004913	0.009464	0.024926	0.010392	1.000000	-
MultipleLines	-0.011430	0.000654	0.135743	0.147910	0.350499	-0.017479	
InternetService	-0.013718	-0.001858	-0.033614	0.006319	-0.030184	0.390024	-
OnlineSecurity	0.001855	-0.020759	-0.127915	0.161958	0.328139	-0.011130	
OnlineBackup	-0.004541	-0.018045	-0.017460	0.162627	0.367155	0.027494	
DeviceProtection	-0.010189	-0.003010	-0.019911	0.165875	0.372424	0.002764	
TechSupport	0.009791	-0.008468	-0.152474	0.133171	0.326081	-0.012189	
StreamingTV	-0.001738	-0.008193	0.026869	0.143919	0.292981	0.055582	
StreamingMovies	-0.019630	-0.007297	0.042940	0.134082	0.307437	0.047035	
PaymentMethod	0.023577	0.015745	-0.037653	-0.151566	-0.359652	-0.003853	-
MonthlyCharges	-0.009830	-0.014286	0.219387	0.104006	0.256983	0.251029	
TotalCharges	0.017044	-0.011860	0.041596	0.072476	0.160293	0.081162	
Churn	0.006630	-0.009548	0.150097	-0.146840	-0.348469	0.009421	

In [54]:

```
corr=data.corr()
corr.style.background_gradient(cmap='coolwarm').set_precision(4)
```

Out[54]:

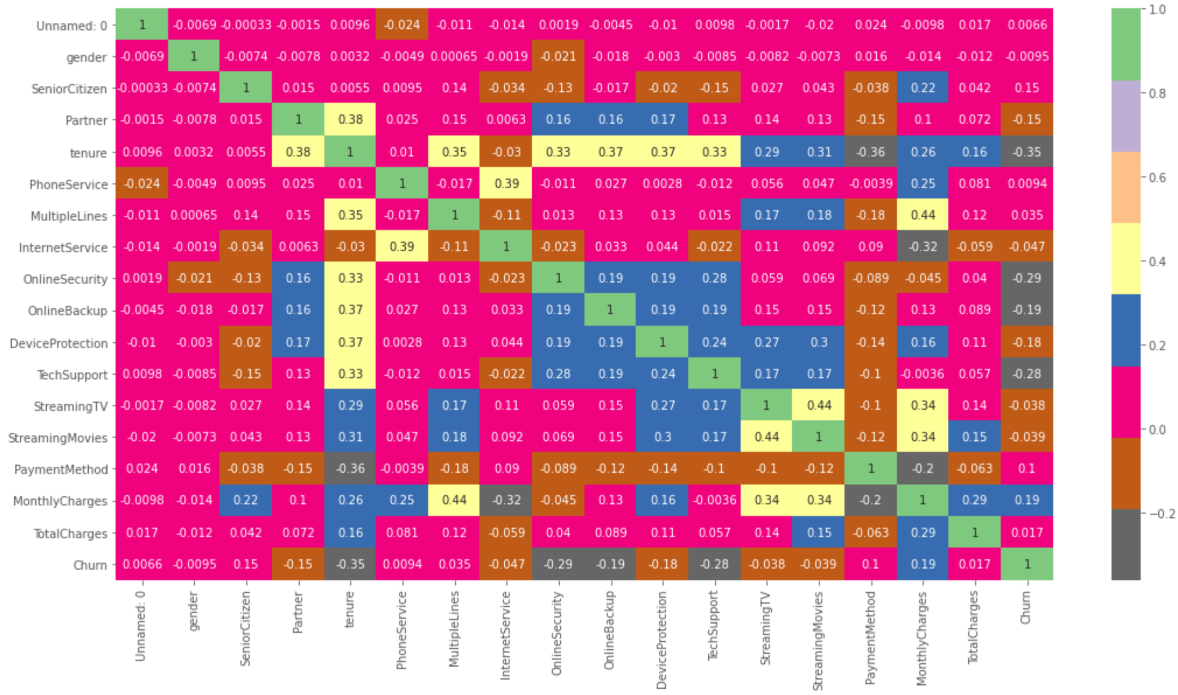
	Unnamed: 0	gender	SeniorCitizen	Partner	tenure	PhoneService	MultipleLine
Unnamed: 0	1.0000	-0.0069	-0.0003	-0.0015	0.0096	-0.0245	-0.01
gender	-0.0069	1.0000	-0.0074	-0.0078	0.0032	-0.0049	0.00
SeniorCitizen	-0.0003	-0.0074	1.0000	0.0149	0.0055	0.0095	0.13
Partner	-0.0015	-0.0078	0.0149	1.0000	0.3820	0.0249	0.14
tenure	0.0096	0.0032	0.0055	0.3820	1.0000	0.0104	0.35
PhoneService	-0.0245	-0.0049	0.0095	0.0249	0.0104	1.0000	-0.01
MultipleLines	-0.0114	0.0007	0.1357	0.1479	0.3505	-0.0175	1.00
InternetService	-0.0137	-0.0019	-0.0336	0.0063	-0.0302	0.3900	-0.10
OnlineSecurity	0.0019	-0.0208	-0.1279	0.1620	0.3281	-0.0111	0.01
OnlineBackup	-0.0045	-0.0180	-0.0175	0.1626	0.3672	0.0275	0.12
DeviceProtection	-0.0102	-0.0030	-0.0199	0.1659	0.3724	0.0028	0.12
TechSupport	0.0098	-0.0085	-0.1525	0.1332	0.3261	-0.0122	0.01
StreamingTV	-0.0017	-0.0082	0.0269	0.1439	0.2930	0.0556	0.17
StreamingMovies	-0.0196	-0.0073	0.0429	0.1341	0.3074	0.0470	0.18
PaymentMethod	0.0236	0.0157	-0.0377	-0.1516	-0.3597	-0.0039	-0.17
MonthlyCharges	-0.0098	-0.0143	0.2194	0.1040	0.2570	0.2510	0.43
TotalCharges	0.0170	-0.0119	0.0416	0.0725	0.1603	0.0812	0.11
Churn	0.0066	-0.0095	0.1501	-0.1468	-0.3485	0.0094	0.03

In [55]:

```
plt.figure(figsize=(18,9))
sns.heatmap(data.corr(),annot = True, cmap ="Accent_r")
```

Out[55]:

```
<AxesSubplot:>
```

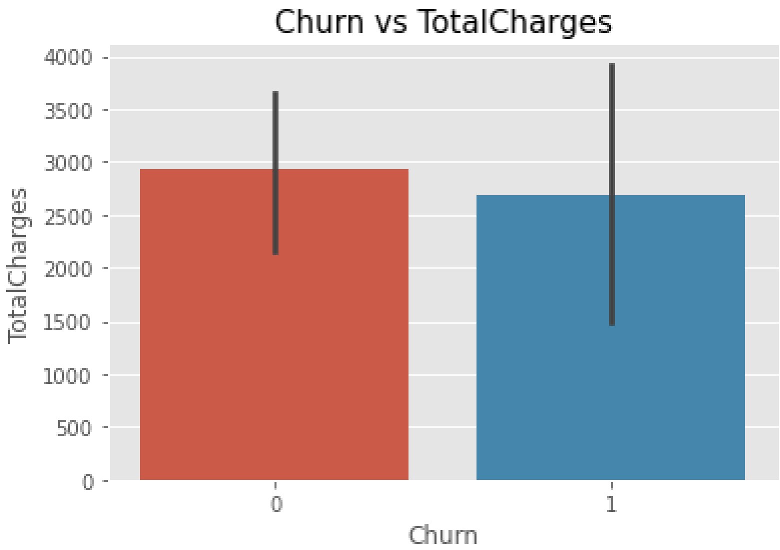


In [ ]:

```
#data visualization
```

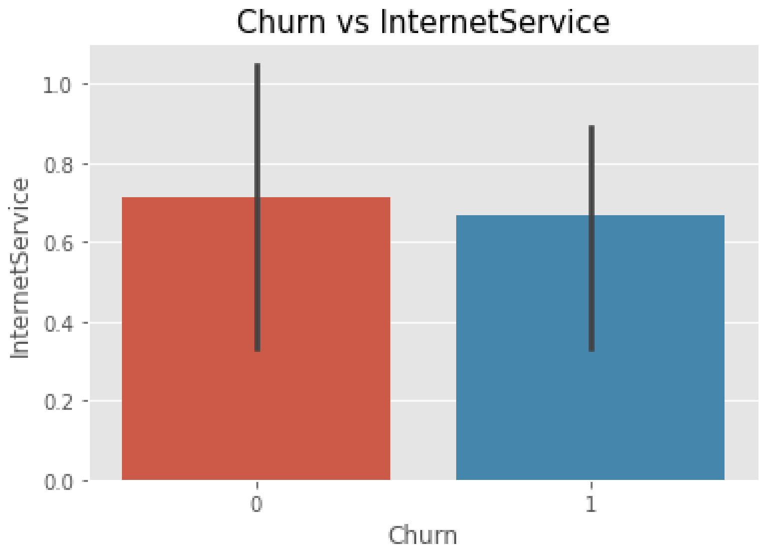
In [56]:

```
sns.barplot(x="Churn", y="TotalCharges",data=data[160:190])
plt.title("Churn vs TotalCharges",fontsize=15)
plt.xlabel("Churn")
plt.ylabel("TotalCharges")
plt.show()
plt.style.use("ggplot")
```



In [57]:

```
sns.barplot(x="Churn", y="InternetService",data=data[160:190])
plt.title("Churn vs InternetService",fontsize=15)
plt.xlabel("Churn")
plt.ylabel("InternetService")
plt.show()
plt.style.use("ggplot")
```

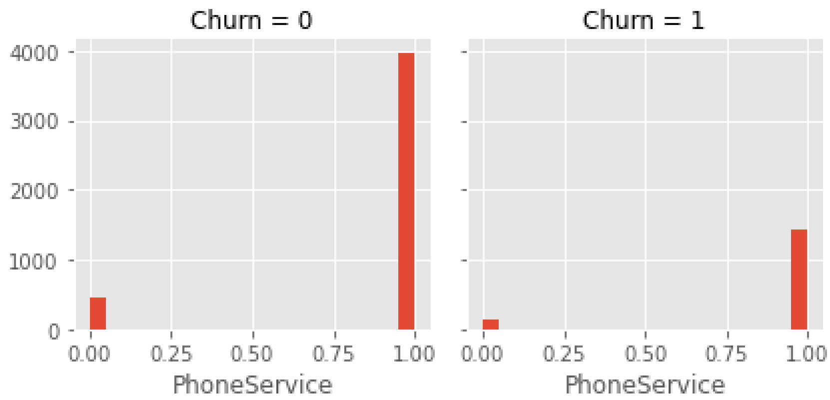


In [58]:

```
g=sns.FacetGrid(data,col='Churn')
g.map(plt.hist,'PhoneService',bins=20)
```

Out[58]:

```
<seaborn.axisgrid.FacetGrid at 0x6a30214490>
```

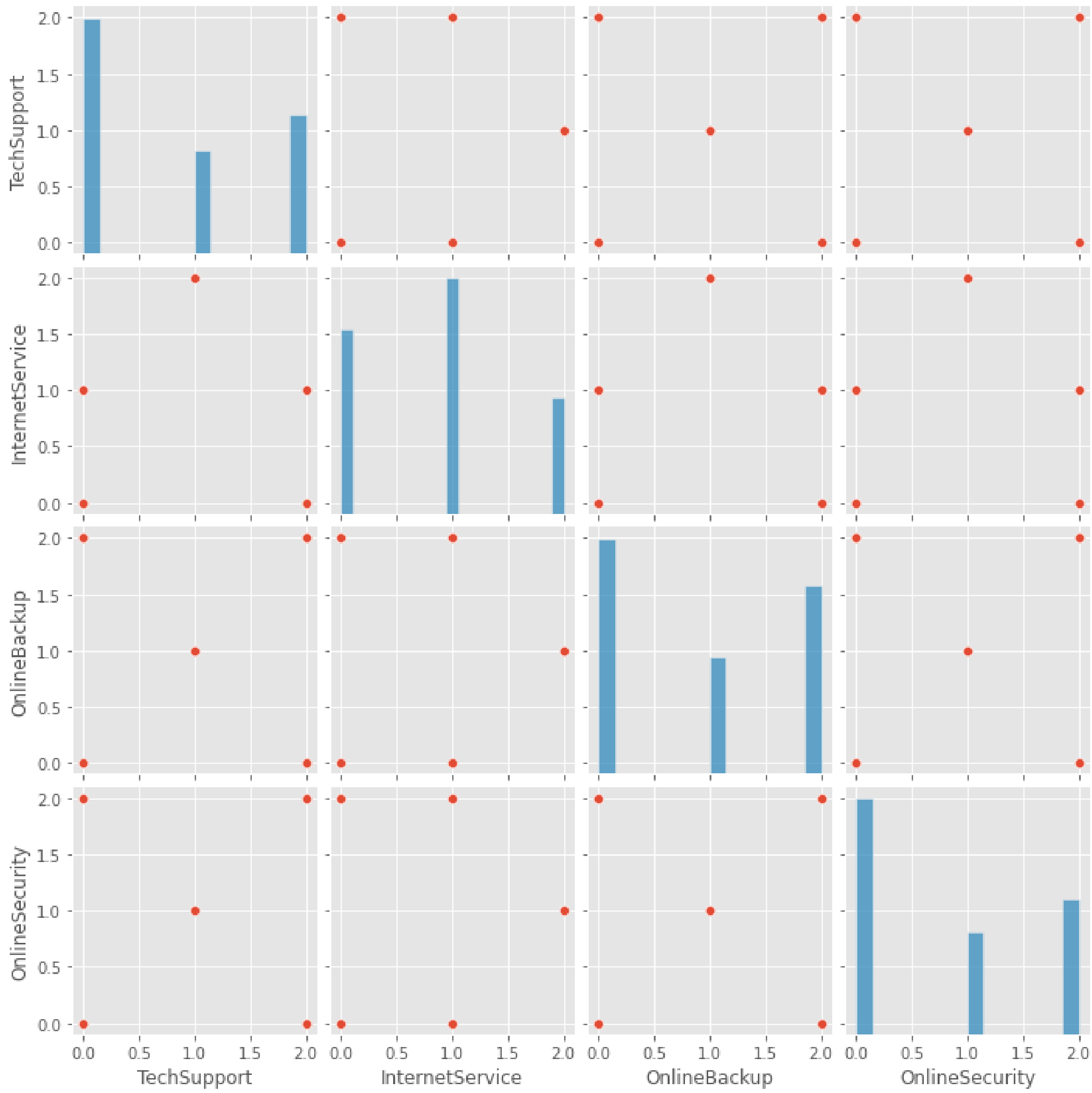


In [59]:

```
sns.pairplot(data[['TechSupport','InternetService','OnlineBackup','OnlineSecurity']])
```

Out[59]:

```
<seaborn.axisgrid.PairGrid at 0x6a2bd68a60>
```



In [60]:

```
#Logistic Regression Code
#import relevant libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics

#features extraction
x = data[['tenure','PhoneService','MultipleLines','InternetService','OnlineSecurity','Onlin
y = data['Churn']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=0)  #split

logreg = LogisticRegression()   #build our logistic model
logreg.fit(x_train, y_train)  #fitting training data
y_pred  = logreg.predict(x_test)    #testing model’s performance
print("Accuracy={:.2f}".format(logreg.score(x_test, y_test)))
```

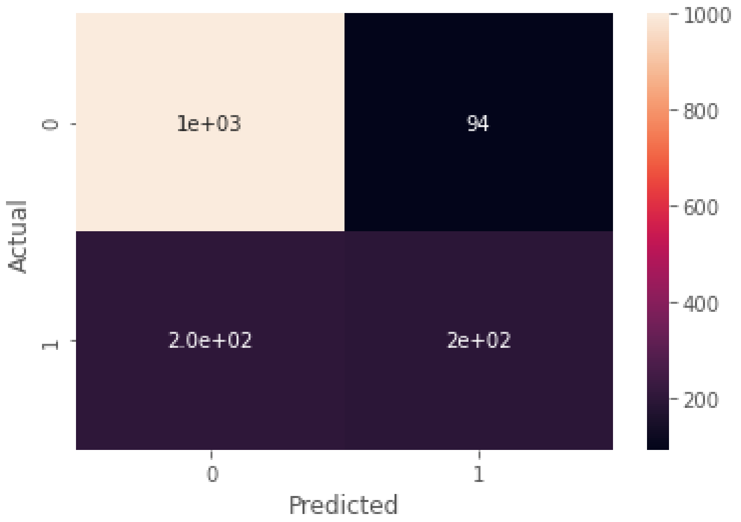
```
Accuracy=0.80
```

In [38]:

```
#2. Use confusion matrix to validate your model.
confusion_matrix = pd.crosstab(y_test, y_pred, rownames=['Actual'], colnames=['Predicted'])
sns.heatmap(confusion_matrix, annot=True)
```

Out[38]:

```
<AxesSubplot:xlabel='Predicted', ylabel='Actual'>
```



In [34]:

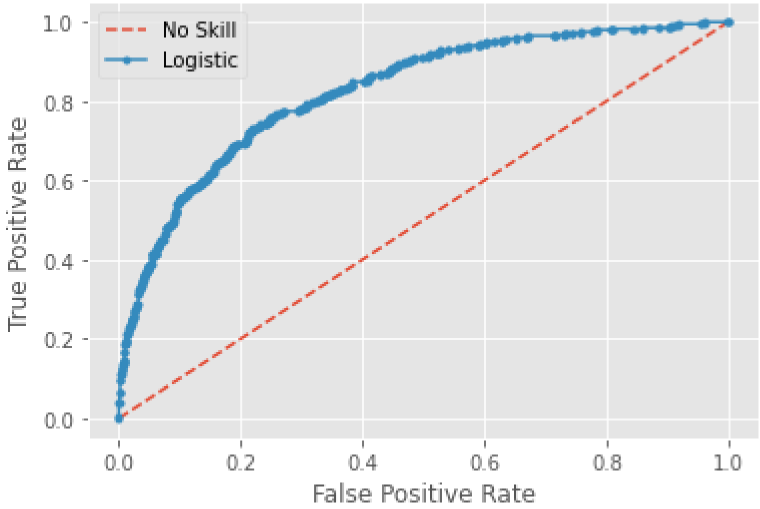
```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
print (y_pred) #predicted values
```

```
              precision    recall  f1-score   support

           0       0.83      0.91      0.87      1094
           1       0.68      0.49      0.57       403

    accuracy                           0.80      1497
   macro avg       0.75      0.70      0.72      1497
weighted avg       0.79      0.80      0.79      1497

[0 0 0 ... 1 0 0]
```

In [36]:

```
# 3. Another validation matrix for classification is ROC / AUC , do your research on them e
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from matplotlib import pyplot
#features extraction
x = data[['tenure','PhoneService','MultipleLines','InternetService','OnlineSecurity','Onlin
y = data['Churn']
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=0)  #split
# generate a no skill prediction (majority class)
ns_probs = [0 for _ in range(len(y_test))]
# fit a model
model = LogisticRegression(solver='lbfgs')
model.fit(x_train, y_train)
# predict probabilities
lr_probs = model.predict_proba(x_test)
# keep probabilities for the positive outcome only
lr_probs = lr_probs[:, 1]
# calculate scores
ns_auc = roc_auc_score(y_test, ns_probs)
lr_auc = roc_auc_score(y_test, lr_probs)
# summarize scores
print('No Skill: ROC AUC=%.3f' % (ns_auc))
print('Logistic: ROC AUC=%.3f' % (lr_auc))
# calculate roc curves
ns_fpr, ns_tpr, _ = roc_curve(y_test, ns_probs)
lr_fpr, lr_tpr, _ = roc_curve(y_test, lr_probs)
# plot the roc curve for the model
pyplot.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
pyplot.plot(lr_fpr, lr_tpr, marker='.', label='Logistic')
# axis labels
pyplot.xlabel('False Positive Rate')
pyplot.ylabel('True Positive Rate')
# show the legend
pyplot.legend()
# show the plot
pyplot.show()
```

```
No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.827
```


No Skill
Logistic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

In [ ]:

In [ ]: