# EDA No. 4 AAA Project Martin George mgeorgevienna@gmail.com

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
df = pd.read_csv('member_sample.csv', index_col = 0)
```

# Application of classification model on AAA data

In [2]:
```
df.head()
df.info()
df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21344 entries, 0 to 99998
Columns: 112 entries, Individual Key to Was Towed To AAR Referral
dtypes: float64(35), object(77)
memory usage: 18.4+ MB
```

Out[2]:
```
Index(['Individual Key', 'Household Key', 'Member Flag', 'City',
       'State - Grouped', 'ZIP5', 'ZIP9', 'FSV CMSI Flag',
       'FSV Credit Card Flag', 'FSV Deposit Program Flag',
       ...
       'SC Vehicle Manufacturer Name', 'SC Vehicle Model Name',
       'SVC Facility Name', 'SVC Facility Type', 'Total Cost',
       'Tow Destination Latitude', 'Tow Destination Longitude',
       'Tow Destination Name', 'Was Duplicated', 'Was Towed To AAR Referral'],
      dtype='object', length=112)
```

In [3]:
```
df.head()
```

Out[3]:

| | Individual Key | Household Key | Member Flag | City | State - Grouped | ZIP5 | ZIP9 | FSV CMSI Flag | FSV Credit Card Flag | FSV Deposit Program Flag | ... | SC Vehicle Manufacturer Name | SC Vehicle Model Name | SVC Facility Name | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 10000003.0 | 10462590.0 | Y | NEW HAVEN | CT | 6511.0 | 65111349.0 | N | N | N | ... | NaN | NaN | NaN | |
| **1** | 52211550.0 | 4500791.0 | Y | WEST WARWICK | RI | 2893.0 | 28933850.0 | N | Y | N | ... | TOYOTA | CAMRY | ASTRO WRECKER SERVICE | indepe |
| **2** | 52211550.0 | 4500791.0 | Y | WEST WARWICK | RI | 2893.0 | 28933850.0 | N | Y | N | ... | TOYOTA | CAMRY | Astro Wrecker Service | indepe |
| **3** | 52211550.0 | 4500791.0 | Y | WEST WARWICK | RI | 2893.0 | 28933850.0 | N | Y | N | ... | TOYOTA | CAMRY | ASTRO WRECKER SERVICE | indepe |
| **4** | 52211550.0 | 4500791.0 | Y | WEST WARWICK | RI | 2893.0 | 28933850.0 | N | Y | N | ... | TOYOTA | CAMRY | ASTRO WRECKER SERVICE | indepe |

5 rows × 112 columns

In [4]:
```
df['Do Not Direct Mail Solicit'].isna().sum()
```

Out[4]:
```
1
```

In [5]:
```
df.groupby('Do Not Direct Mail Solicit')['Total Cost'].mean()
```

Out[5]:
```
Do Not Direct Mail Solicit
0.0    47.270066
1.0    47.363815
Name: Total Cost, dtype: float64
```

In [6]:
```
df[['Do Not Direct Mail Solicit','Email Available','Email Status']].head()
```

Out[6]:

| | Do Not Direct Mail Solicit | Email Available | Email Status |
|---|---|---|---|
| **0** | NaN | NaN | NaN |
| **1** | 0.0 | 0.0 | NaN |
| **2** | 0.0 | 0.0 | NaN |
| **3** | 0.0 | 0.0 | NaN |
| **4** | 0.0 | 0.0 | NaN |

In [7]:
```
sub_df = df[df['Email Available'] == 1.0]
```

In [8]:
```
sub_df.shape
```

Out[8]:
```
(11442, 112)
```

In [9]:
```
sub_df['Email Status'].value_counts(dropna = False)
```

Out[9]:
```
Active          4394
Unsubscribed    4376
NaN             2031
Held             515
Bounced          126
Name: Email Status, dtype: int64
```

In [10]:
```
#df.loc[(df['Email Available'] != 1.0 )]
df_email_miss= df.loc[(df['Do Not Direct Mail Solicit'] != 1.0 ) & (df['Email Available'] != 1.0 ) ]

u_hh = df_email_miss["Household Key"].unique()
len(u_hh)
#n_by_state = df.groupby("state")["last_name"].count()
```

Out[10]:
```
3836
```

In [11]:
```
df_mail=df[df['Do Not Direct Mail Solicit'] != 1.0]
df_mail_ava=df_mail[df['Email Available'] != 1.0]
df_mail_ava.shape
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:2: UserWarning: Boolean Seri
es key will be reindexed to match DataFrame index.
```

Out[11]:
```
(9221, 112)
```

In [12]:
```
df_mail_ava.shape
```

Out[12]:
```
(9221, 112)
```

# Recommendation for AAA on email address

There are 9221 rows with email addresses are missing , even though these c ustomers have not blocked direct emails from AAA. Out of these 9221, 3836 are unique householders

In [13]:
```
df['Reason Joined'].shape
```

Out[13]:
```
(21344,)
```

In [14]:
```
df['Reason Joined'].value_counts(dropna = False)
```

Out[14]:
```
NaN                      20956
U                          168
Dependable Services        127
5                           45
Family Plan Avail           19
Nation Wide Rd Srv           7
Gift Membership              5
Free Membership              4
Club Reputation              3
3                            3
Convenient Offices           1
Variety of Services          1
Direct Mail                  1
Prior Family Exp             1
7                            1
Other                        1
Recommend/Referral           1
Name: Reason Joined, dtype: int64
```

# Study on 'Reason Joined' field

# Using KNN algorithm, we can fill the missing values based on another field, example 'Total Cost'

In [15]:
```
test_df = df.loc[df['Reason Joined'].isna()]
```

In [16]:
```
test_df.shape
```

Out[16]:
```
(20956, 112)
```

In [17]:
```
train_df = df.loc[df['Reason Joined'].notna()]
```

In [18]:
```
train_df.shape
```

Out[18]:
```
(388, 112)
```

In [19]:
```
train_df['Reason Joined'].value_counts(dropna = False)
```

Out[19]:
```
U                         168
Dependable Services       127
5                          45
Family Plan Avail          19
Nation Wide Rd Srv          7
Gift Membership             5
Free Membership             4
3                           3
Club Reputation             3
Other                       1
Convenient Offices          1
Recommend/Referral          1
7                           1
Variety of Services         1
Direct Mail                 1
Prior Family Exp            1
Name: Reason Joined, dtype: int64
```

In [20]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [21]:
```
knn = KNeighborsClassifier()
```

In [22]:
```
train_df = train_df[['Total Cost','Reason Joined']].dropna()
```

In [23]:
```
X= train_df[['Total Cost']]
y = train_df['Reason Joined']
```

In [24]:
```
X.shape
```

Out[24]:
```
(277, 1)
```

In [25]:
```
y.shape
```

Out[25]:
```
(277,)
```

In [26]:
```
knn.fit(X,y)
```

Out[26]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=5, p=2,
           weights='uniform')
```

In [27]:
```
fill_x = test_df[['Total Cost','Reason Joined']].dropna(subset = ['Total Cost'])
```

In [28]:
```
fill_x['Reason Joined'].value_counts(dropna = False)
```

Out[28]:
```
NaN    13667
Name: Reason Joined, dtype: int64
```

In [29]:
```
fill_x.shape
```

Out[29]:
```
(13667, 2)
```

In [30]:
```
test_df.shape
```

Out[30]:
```
(20956, 112)
```

In [31]:
```
x_test = fill_x[['Total Cost']]
```

In [32]:
```
knn.predict(x_test)
```

Out[32]:
```
array(['Dependable Services', 'Dependable Services',
       'Dependable Services', ..., 'Dependable Services',
       'Dependable Services', 'Dependable Services'], dtype=object)
```

In [33]:
```
x_test.shape
```

Out[33]:
```
(13667, 1)
```

In [34]:
```
fill_b = fill_x
```

In [35]:
```
fill_x['Reason Joined'].value_counts(dropna = False)
```

Out[35]:
```
NaN    13667
Name: Reason Joined, dtype: int64
```

In [36]:
```
fill_x['Reason Joined'] = knn.predict(x_test)
```

In [37]:

```
fill_x[['Total Cost','Reason Joined']]
```

Out[37]:

| | Total Cost | Reason Joined |
|---|---|---|
| 1 | 32.50 | Dependable Services |
| 2 | 30.00 | Dependable Services |
| 3 | 32.50 | Dependable Services |
| 4 | 30.00 | Dependable Services |
| 5 | 53.00 | Dependable Services |
| 6 | 30.00 | Dependable Services |
| 7 | 32.00 | Dependable Services |
| 8 | 32.00 | Dependable Services |
| 9 | 32.50 | Dependable Services |
| 11 | 58.85 | Dependable Services |
| 12 | 53.00 | Dependable Services |
| 13 | 53.00 | Dependable Services |
| 16 | 53.00 | Dependable Services |
| 17 | 53.00 | Dependable Services |
| 18 | 53.00 | Dependable Services |
| 19 | 29.00 | Dependable Services |
| 20 | 28.00 | U |
| 21 | 53.00 | Dependable Services |
| 22 | 53.00 | Dependable Services |
| 23 | 53.00 | Dependable Services |
| 24 | 53.00 | Dependable Services |
| 25 | 53.00 | Dependable Services |
| 26 | 58.85 | Dependable Services |
| 27 | 53.00 | Dependable Services |
| 28 | 53.00 | Dependable Services |
| 29 | 53.00 | Dependable Services |
| 32 | 58.85 | Dependable Services |
| 34 | 58.85 | Dependable Services |
| 36 | 29.00 | Dependable Services |

| | Total Cost | Reason Joined |
|---|---|---|
| **41** | 33.00 | U |
| **...** | ... | ... |
| **99515** | 53.00 | Dependable Services |
| **99516** | 53.00 | Dependable Services |
| **99539** | 39.00 | U |
| **99540** | 58.85 | Dependable Services |
| **99541** | 0.00 | U |
| **99542** | 53.00 | Dependable Services |
| **99543** | 53.00 | Dependable Services |
| **99544** | 53.00 | Dependable Services |
| **99545** | 53.00 | Dependable Services |
| **99966** | 58.85 | Dependable Services |
| **99967** | 53.00 | Dependable Services |
| **99969** | 47.00 | Dependable Services |
| **99970** | 58.85 | Dependable Services |
| **99971** | 58.85 | Dependable Services |
| **99972** | 53.00 | Dependable Services |
| **99973** | 53.00 | Dependable Services |
| **99974** | 58.85 | Dependable Services |
| **99977** | 34.00 | U |
| **99983** | 29.00 | Dependable Services |
| **99984** | 44.00 | Dependable Services |
| **99985** | 29.00 | Dependable Services |
| **99986** | 47.00 | Dependable Services |
| **99987** | 44.00 | Dependable Services |
| **99988** | 82.00 | Dependable Services |
| **99989** | 29.43 | Dependable Services |
| **99993** | 53.00 | Dependable Services |
| **99994** | 36.00 | U |
| **99995** | 53.00 | Dependable Services |

| | Total Cost | Reason Joined |
|---|---|---|
| **99996** | 58.85 | Dependable Services |
| **99997** | 58.85 | Dependable Services |

13667 rows × 2 columns

In [38]:
```
fill_x['Reason Joined'].value_counts(dropna = False)
```

Out[38]:
```
Dependable Services    10679
U                       2840
5                        144
Family Plan Avail          4
Name: Reason Joined, dtype: int64
```

In [39]:
```
fill_b['Reason Joined'].value_counts(dropna = False)
```

Out[39]:
```
Dependable Services    10679
U                       2840
5                        144
Family Plan Avail          4
Name: Reason Joined, dtype: int64
```

# KNN is an optional algorithm to fill the missing values of a feature column, based on an another relaible feature. In this aabove example, based on the Total cost, we could predict the missing values of " Reason Joined" feature.

# sklearn imputer is also a method for filling missing values. Testing follows.

In [137]:
```
from sklearn.impute import SimpleImputer
```

In [149]:
```
imputer = SimpleImputer(strategy = 'median')
```

In [146]:
```
df_to_impute = df.select_dtypes(['int','float'])
```

In [147]:
```
df_to_impute
```

Out[147]:

| | Individual Key | Household Key | ZIP5 | ZIP9 | Length Of Residence | Do Not Direct Mail Solicit | Email Available | ERS ENT Count Year 1 | ERS ENT Count Year 2 | ERS ENT Count Year 3 | ... | Member Match Flag | Member Number and Associate ID | Plus Cost | Premier Cost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 10000003.0 | 10462590.0 | 6511.0 | 65111349.0 | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **1** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **2** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **3** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **4** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **5** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **6** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **7** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **8** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **9** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 15300.0 | 0.0 | 0.0 |
| **10** | 1606764.0 | 4317516.0 | 2878.0 | 28781026.0 | NaN | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **11** | 2766867.0 | 11622991.0 | 2889.0 | 28892920.0 | 13.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.0 | ... | 1.0 | 16300.0 | 0.0 | 0.0 |
| **12** | 2766867.0 | 11622991.0 | 2889.0 | 28892920.0 | 13.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.0 | ... | 1.0 | 16300.0 | 0.0 | 0.0 |
| **13** | 2766867.0 | 11622991.0 | 2889.0 | 28892920.0 | 13.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.0 | ... | 1.0 | 16300.0 | 0.0 | 0.0 |
| **14** | 2766869.0 | 11622991.0 | 2889.0 | 28892920.0 | 13.0 | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **15** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **16** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 16322.0 | 0.0 | 0.0 |
| **17** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 16322.0 | 0.0 | 0.0 |
| **18** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 16322.0 | 0.0 | 0.0 |
| **19** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 16322.0 | 0.0 | 0.0 |
| **20** | 13746947.0 | 579810.0 | 2863.0 | 28631322.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 18200.0 | 0.0 | 0.0 |
| **21** | 1788453.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 18800.0 | 0.0 | 0.0 |
| **22** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 18820.0 | 0.0 | 0.0 |
| **23** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 18820.0 | 0.0 | 0.0 |
| **24** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 18820.0 | 0.0 | 0.0 |
| **25** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 18820.0 | 0.0 | 0.0 |
| **26** | 1788455.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | ... | 1.0 | 18821.0 | 0.0 | 0.0 |

| | Individual Key | Household Key | ZIP5 | ZIP9 | Length Of Residence | Do Not Direct Mail Solicit | Email Available | ERS ENT Count Year 1 | ERS ENT Count Year 2 | ERS ENT Count Year 3 | ... | Member Match Flag | Member Number and Associate ID | Plus Cost | Premier Cost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **27** | 14243585.0 | 7728088.0 | 2806.0 | 28065003.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | ... | 1.0 | 19100.0 | 0.0 | 0.0 |
| **28** | 14243587.0 | 7728088.0 | 2806.0 | 28065003.0 | 13.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 19120.0 | 0.0 | 0.0 |
| **29** | 14243587.0 | 7728088.0 | 2806.0 | 28065003.0 | 13.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 19120.0 | 0.0 | 0.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **99968** | 4458026.0 | 1588987.0 | 2886.0 | 28861711.0 | NaN | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99969** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 1.0 | 54348200.0 | 0.0 | 0.0 |
| **99970** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 1.0 | 54348200.0 | 0.0 | 0.0 |
| **99971** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 1.0 | 54348200.0 | 0.0 | 0.0 |
| **99972** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 1.0 | 54348200.0 | 0.0 | 0.0 |
| **99973** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 1.0 | 54348200.0 | 0.0 | 0.0 |
| **99974** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 1.0 | 54348200.0 | 0.0 | 0.0 |
| **99975** | 12849941.0 | 16604128.0 | 2906.0 | 29063709.0 | NaN | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99976** | 22426406.0 | 45466286.0 | 2809.0 | 28092304.0 | 12.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | NaN | NaN | NaN | NaN |
| **99977** | 22426405.0 | 45466286.0 | 2809.0 | 28092304.0 | 15.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 1.0 | 54349620.0 | 0.0 | 0.0 |
| **99979** | 19764804.0 | 15397653.0 | 2871.0 | 28712143.0 | NaN | 0.0 | 1.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99980** | 19764802.0 | 15397653.0 | 2871.0 | 28712143.0 | NaN | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99981** | 19764801.0 | 15397653.0 | 2871.0 | 28712143.0 | NaN | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99982** | 19764793.0 | 15397653.0 | 2871.0 | 28712143.0 | 0.0 | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99983** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 54353700.0 | 0.0 | 0.0 |
| **99984** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 54353700.0 | 18.0 | 0.0 |
| **99985** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 54353700.0 | 0.0 | 0.0 |
| **99986** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 54353700.0 | 18.0 | 0.0 |
| **99987** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 54353700.0 | 18.0 | 0.0 |
| **99988** | 16521336.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 54353720.0 | 39.0 | 0.0 |
| **99989** | 1619870.0 | 5462399.0 | 2879.0 | 28791421.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 54365300.0 | 0.0 | 0.0 |
| **99990** | 1619868.0 | 5462399.0 | 2879.0 | 28791421.0 | NaN | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99991** | 1619869.0 | 5462399.0 | 2879.0 | 28791421.0 | NaN | 0.0 | 0.0 | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **99992** | 54745437.0 | 5462399.0 | 2816.0 | 28167132.0 | 10.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | NaN | NaN | NaN | NaN |

| | Individual Key | Household Key | ZIP5 | ZIP9 | Length Of Residence | Do Not Direct Mail Solicit | Email Available | ERS ENT Count Year 1 | ERS ENT Count Year 2 | ERS ENT Count Year 3 | ... | Member Match Flag | Member Number and Associate ID | Plus Cost | Premier Cost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **99993** | 25797262.0 | 20330346.0 | 2886.0 | 28867552.0 | NaN | 0.0 | 0.0 | 1.0 | 0.0 | 3.0 | ... | 1.0 | 54367300.0 | 0.0 | 0.0 |
| **99994** | 25797262.0 | 20330346.0 | 2886.0 | 28867552.0 | NaN | 0.0 | 0.0 | 1.0 | 0.0 | 3.0 | ... | 1.0 | 54367300.0 | 6.0 | 0.0 |
| **99995** | 25797262.0 | 20330346.0 | 2886.0 | 28867552.0 | NaN | 0.0 | 0.0 | 1.0 | 0.0 | 3.0 | ... | 1.0 | 54367300.0 | 0.0 | 0.0 |
| **99996** | 28273400.0 | 8325571.0 | 2886.0 | 28868235.0 | 13.0 | 0.0 | 1.0 | 2.0 | 0.0 | 3.0 | ... | 1.0 | 54369500.0 | 0.0 | 0.0 |
| **99997** | 28273400.0 | 8325571.0 | 2886.0 | 28868235.0 | 13.0 | 0.0 | 1.0 | 2.0 | 0.0 | 3.0 | ... | 1.0 | 54369500.0 | 0.0 | 0.0 |
| **99998** | 28273400.0 | 8325571.0 | 2886.0 | 28868235.0 | 13.0 | 0.0 | 1.0 | 2.0 | 0.0 | 3.0 | ... | 1.0 | 54369500.0 | NaN | NaN |

21344 rows × 35 columns

In [161]:
```
X = df_to_impute.dropna(subset = ['Total Cost']).drop('Total Cost', axis = 1)
```

In [177]:
```
y = df_to_impute.dropna(subset = ['Total Cost'])['Total Cost']
```

In [178]:
```
imputer.fit(X)
```

Out[178]:
```
SimpleImputer(copy=True, fill_value=None, missing_values=nan,
       strategy='median', verbose=0)
```

In [179]:
```
X = imputer.transform(X)
```

In [180]:

```
pd.DataFrame(X)
```

Out[180]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 23 | 24 | 25 | 26 | 27 | 28 | 29 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 9707320.0 | 41.00000 | -71.000 |
| **1** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 6361198.0 | 0.00000 | 0.000 |
| **2** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 9127495.0 | 0.00000 | 0.000 |
| **3** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 1593215.0 | 0.00000 | 0.000 |
| **4** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 3652711.0 | 0.00000 | 0.000 |
| **5** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 1.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 4646305.0 | 0.00000 | 0.000 |
| **6** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 7373094.0 | 0.00000 | 0.000 |
| **7** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 7460803.0 | 0.00000 | 0.000 |
| **8** | 52211550.0 | 4500791.0 | 2893.0 | 28933850.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 15300.0 | 0.0 | 0.0 | 9152542.0 | 0.00000 | 0.000 |
| **9** | 2766867.0 | 11622991.0 | 2889.0 | 28892920.0 | 13.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 16300.0 | 0.0 | 0.0 | 18842978.0 | 0.00000 | 0.000 |
| **10** | 2766867.0 | 11622991.0 | 2889.0 | 28892920.0 | 13.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 16300.0 | 0.0 | 0.0 | 2075597.0 | 0.00000 | 0.000 |
| **11** | 2766867.0 | 11622991.0 | 2889.0 | 28892920.0 | 13.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.0 | ... | 1.0 | 1.0 | 16300.0 | 0.0 | 0.0 | 5005248.0 | 0.00000 | 0.000 |
| **12** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 16322.0 | 0.0 | 0.0 | 1640562.0 | 0.00000 | 0.000 |
| **13** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 16322.0 | 0.0 | 0.0 | 1711324.0 | 0.00000 | 0.000 |
| **14** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 16322.0 | 0.0 | 0.0 | 1852044.0 | 0.00000 | 0.000 |
| **15** | 2766868.0 | 11622991.0 | 2889.0 | 28892920.0 | 5.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 16322.0 | 0.0 | 0.0 | 5441060.0 | 0.00000 | 0.000 |
| **16** | 13746947.0 | 579810.0 | 2863.0 | 28631322.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 18200.0 | 0.0 | 0.0 | 163014.0 | 0.00000 | 0.000 |
| **17** | 1788453.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 18800.0 | 0.0 | 0.0 | 7711905.0 | 0.00000 | 0.000 |
| **18** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 18820.0 | 0.0 | 0.0 | 1575870.0 | 0.00000 | 0.000 |
| **19** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 18820.0 | 0.0 | 0.0 | 1677303.0 | 0.00000 | 0.000 |
| **20** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 18820.0 | 0.0 | 0.0 | 3947898.0 | 0.00000 | 0.000 |
| **21** | 1788452.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 18820.0 | 0.0 | 0.0 | 3952912.0 | 0.00000 | 0.000 |
| **22** | 1788455.0 | 7187017.0 | 2888.0 | 28882811.0 | 15.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 18821.0 | 0.0 | 0.0 | 19656374.0 | 41.71028 | -71.491 |
| **23** | 14243585.0 | 7728088.0 | 2806.0 | 28065003.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | ... | 1.0 | 1.0 | 19100.0 | 0.0 | 0.0 | 385595.0 | 0.00000 | 0.000 |
| **24** | 14243587.0 | 7728088.0 | 2806.0 | 28065003.0 | 13.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 19120.0 | 0.0 | 0.0 | 3121195.0 | 41.00000 | -71.000 |
| **25** | 14243587.0 | 7728088.0 | 2806.0 | 28065003.0 | 13.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 19120.0 | 0.0 | 0.0 | 4997740.0 | 0.00000 | 0.000 |
| **26** | 4064211.0 | 8724100.0 | 2879.0 | 28791619.0 | 15.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 1.0 | 20500.0 | 0.0 | 0.0 | 17543023.0 | 0.00000 | 0.000 |
| **27** | 195833722.0 | 12394451.0 | 2920.0 | 29205904.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 20800.0 | 0.0 | 0.0 | 20438910.0 | 0.00000 | 0.000 |
| **28** | 7182300.0 | 300071.0 | 6111.0 | 61111421.0 | 15.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 20900.0 | 0.0 | 0.0 | 1892106.0 | 0.00000 | 0.000 |

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 23 | 24 | 25 | 26 | 27 | 28 | 29 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **29** | 53136272.0 | 37549482.0 | 2814.0 | 28140051.0 | 10.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 1.0 | 28200.0 | 0.0 | 0.0 | 17183027.0 | 41.92437 | -71.674 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **13914** | 205158507.0 | 95100521.0 | 2871.0 | 28714086.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 53449000.0 | 0.0 | 0.0 | 4583123.0 | 0.00000 | 0.000 |
| **13915** | 205158507.0 | 95100521.0 | 2871.0 | 28714086.0 | 15.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 53449000.0 | 0.0 | 0.0 | 1777244.0 | 0.00000 | 0.000 |
| **13916** | 56389710.0 | 4641147.0 | 2886.0 | 28863516.0 | 15.0 | 0.0 | 1.0 | 0.0 | 2.0 | 3.0 | ... | 0.0 | 1.0 | 53485600.0 | 6.0 | 0.0 | 18794375.0 | 41.74966 | -71.423 |
| **13917** | 56389710.0 | 4641147.0 | 2886.0 | 28863516.0 | 15.0 | 0.0 | 1.0 | 0.0 | 2.0 | 3.0 | ... | 0.0 | 1.0 | 53485600.0 | 0.0 | 0.0 | 18670374.0 | 0.00000 | 0.000 |
| **13918** | 56389710.0 | 4641147.0 | 2886.0 | 28863516.0 | 15.0 | 0.0 | 1.0 | 0.0 | 2.0 | 3.0 | ... | 0.0 | 1.0 | 53485600.0 | 0.0 | 0.0 | 5357333.0 | 0.00000 | 0.000 |
| **13919** | 8867560.0 | 4641147.0 | 2886.0 | 28863516.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 53485620.0 | 0.0 | 0.0 | 2910805.0 | 0.00000 | 0.000 |
| **13920** | 8867560.0 | 4641147.0 | 2886.0 | 28863516.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 53485620.0 | 0.0 | 0.0 | 2328658.0 | 0.00000 | 0.000 |
| **13921** | 8867560.0 | 4641147.0 | 2886.0 | 28863516.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 53485620.0 | 0.0 | 0.0 | 5263118.0 | 0.00000 | 0.000 |
| **13922** | 8867560.0 | 4641147.0 | 2886.0 | 28863516.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | ... | 0.0 | 1.0 | 53485620.0 | 0.0 | 0.0 | 6173883.0 | 0.00000 | 0.000 |
| **13923** | 4458025.0 | 1588987.0 | 2886.0 | 28861711.0 | 15.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 | ... | 0.0 | 1.0 | 54333400.0 | 0.0 | 0.0 | 17851393.0 | 0.00000 | 0.000 |
| **13924** | 4458025.0 | 1588987.0 | 2886.0 | 28861711.0 | 15.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 | ... | 0.0 | 1.0 | 54333400.0 | 0.0 | 0.0 | 3952261.0 | 41.00000 | -71.000 |
| **13925** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 54348200.0 | 0.0 | 0.0 | 20218515.0 | 41.75776 | -71.404 |
| **13926** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 54348200.0 | 0.0 | 0.0 | 19306815.0 | 0.00000 | 0.000 |
| **13927** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 54348200.0 | 0.0 | 0.0 | 19164491.0 | 0.00000 | 0.000 |
| **13928** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 54348200.0 | 0.0 | 0.0 | 5280579.0 | 0.00000 | 0.000 |
| **13929** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 54348200.0 | 0.0 | 0.0 | 7338895.0 | 0.00000 | 0.000 |
| **13930** | 12849942.0 | 16604128.0 | 2906.0 | 29063709.0 | 15.0 | 0.0 | 0.0 | 3.0 | 2.0 | 0.0 | ... | 0.0 | 1.0 | 54348200.0 | 0.0 | 0.0 | 16625008.0 | 0.00000 | 0.000 |
| **13931** | 22426405.0 | 45466286.0 | 2809.0 | 28092304.0 | 15.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 1.0 | 54349620.0 | 0.0 | 0.0 | 5880022.0 | 0.00000 | 0.000 |
| **13932** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 54353700.0 | 0.0 | 0.0 | 6416688.0 | 41.00000 | -71.000 |
| **13933** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 54353700.0 | 18.0 | 0.0 | 6307512.0 | 41.00000 | -71.000 |
| **13934** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 54353700.0 | 0.0 | 0.0 | 7027924.0 | 41.00000 | -71.000 |
| **13935** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 54353700.0 | 18.0 | 0.0 | 6837393.0 | 41.00000 | -71.000 |
| **13936** | 16521338.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 54353700.0 | 18.0 | 0.0 | 3042243.0 | 41.00000 | -71.000 |
| **13937** | 16521336.0 | 13735475.0 | 2809.0 | 28091350.0 | 10.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 54353720.0 | 39.0 | 0.0 | 3017750.0 | 41.00000 | -71.000 |
| **13938** | 1619870.0 | 5462399.0 | 2879.0 | 28791421.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 1.0 | 54365300.0 | 0.0 | 0.0 | 19887931.0 | 41.45056 | -71.525 |
| **13939** | 25797262.0 | 20330346.0 | 2886.0 | 28867552.0 | 15.0 | 0.0 | 0.0 | 1.0 | 0.0 | 3.0 | ... | 0.0 | 1.0 | 54367300.0 | 0.0 | 0.0 | 2284969.0 | 0.00000 | 0.000 |
| **13940** | 25797262.0 | 20330346.0 | 2886.0 | 28867552.0 | 15.0 | 0.0 | 0.0 | 1.0 | 0.0 | 3.0 | ... | 0.0 | 1.0 | 54367300.0 | 6.0 | 0.0 | 3623739.0 | 41.00000 | -71.000 |
| **13941** | 25797262.0 | 20330346.0 | 2886.0 | 28867552.0 | 15.0 | 0.0 | 0.0 | 1.0 | 0.0 | 3.0 | ... | 0.0 | 1.0 | 54367300.0 | 0.0 | 0.0 | 3743487.0 | 0.00000 | 0.000 |

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 23 | 24 | 25 | 26 | 27 | 28 | 29 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **13942** | 28273400.0 | 8325571.0 | 2886.0 | 28868235.0 | 13.0 | 0.0 | 1.0 | 2.0 | 0.0 | 3.0 | ... | 0.0 | 1.0 | 54369500.0 | 0.0 | 0.0 | 19404669.0 | 0.00000 | 0.000 |
| **13943** | 28273400.0 | 8325571.0 | 2886.0 | 28868235.0 | 13.0 | 0.0 | 1.0 | 2.0 | 0.0 | 3.0 | ... | 0.0 | 1.0 | 54369500.0 | 0.0 | 0.0 | 19614013.0 | 0.00000 | 0.000 |

13944 rows × 33 columns

In [181]:

```
pd.DataFrame(X).isna().sum()
```

Out[181]:

```
0     0
1     0
2     0
3     0
4     0
5     0
6     0
7     0
8     0
9     0
10    0
11    0
12    0
13    0
14    0
15    0
16    0
17    0
18    0
19    0
20    0
21    0
22    0
23    0
24    0
25    0
26    0
27    0
28    0
29    0
30    0
31    0
32    0
dtype: int64
```

In [186]:
```
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import SelectKBest
```

In [188]:
```
pipe = Pipeline([('imputer',SimpleImputer()),
                 ('kbest',SelectKBest()),
                 ('lgr',LinearRegression())])
```

In [184]:
```
pipe.fit(X,y)
```

Out[184]:
```
Pipeline(memory=None,
     steps=[('imputer', SimpleImputer(copy=True, fill_value=None, missing_values=nan, strategy='mean',
       verbose=0)), ('lgr', LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False))])
```

# KNN is better than mean or median imputer if more than 50% data is missing

In [190]:

```
df['ZIP5'].value_counts() > 2
```

```
Out[190]: 2920.0     True
          2886.0     True
          2889.0     True
          2888.0     True
          2816.0     True
          2919.0     True
          2818.0     True
          2852.0     True
          2904.0     True
          2910.0     True
          2893.0     True
          2864.0     True
          2906.0     True
          2882.0     True
          2879.0     True
          2911.0     True
          2908.0     True
          2915.0     True
          2865.0     True
          2905.0     True
          2914.0     True
          2871.0     True
          2838.0     True
          2840.0     True
          2860.0     True
          2861.0     True
          2806.0     True
          2828.0     True
          2917.0     True
          2921.0     True
                    ...
          6401.0    False
          6417.0    False
          6032.0    False
          6897.0    False
          6516.0    False
          6033.0    False
          6405.0    False
          6902.0    False
          6109.0    False
          6413.0    False
          6614.0    False
          6420.0    False
          6092.0    False
          6770.0    False
          6480.0    False
          2877.0    False
```

```
6605.0    False
6604.0    False
6825.0    False
6438.0    False
6512.0    False
6066.0    False
6484.0    False
6001.0    False
6517.0    False
6333.0    False
6019.0    False
6374.0    False
6119.0    False
2802.0    False
Name: ZIP5, Length: 182, dtype: bool
```

In [189]:

```
df['ZIP5'].value_counts()[df['ZIP5'].value_counts() > 2]
```

```
Out[189]: 2920.0    1518
          2886.0    1141
          2889.0     826
          2888.0     823
          2816.0     770
          2919.0     703
          2818.0     661
          2852.0     648
          2904.0     641
          2910.0     635
          2893.0     606
          2864.0     560
          2906.0     535
          2882.0     531
          2879.0     507
          2911.0     498
          2908.0     475
          2915.0     440
          2865.0     402
          2905.0     378
          2914.0     371
          2871.0     328
          2838.0     322
          2840.0     321
          2860.0     319
          2861.0     317
          2806.0     313
          2828.0     288
          2917.0     284
          2921.0     273
                    ...
          6250.0       4
          6082.0       4
          6840.0       4
          6058.0       4
          2808.0       4
          2801.0       4
          6831.0       4
          6416.0       4
          2862.0       4
          6084.0       4
          6384.0       4
          6877.0       3
          6340.0       3
          6415.0       3
          6108.0       3
          6460.0       3
```

```
6234.0       3
6457.0       3
6419.0       3
6880.0       3
6443.0       3
6807.0       3
6357.0       3
6525.0       3
6798.0       3
6067.0       3
6354.0       3
6243.0       3
6277.0       3
6057.0       3
Name: ZIP5, Length: 133, dtype: int64
```

In [191]:
```python
df.groupby('FSV CMSI Flag').mean()
```

Out[191]:

| FSV CMSI Flag | Individual Key | Household Key | ZIP5 | ZIP9 | Length Of Residence | Do Not Direct Mail Solicit | Email Available | ERS ENT Count Year 1 | ERS ENT Count Year 2 | ERS ENT Count Year 3 | ... | Member Match Flag | Me Numbe Associ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N | 3.403291e+07 | 1.600860e+07 | 2947.671848 | 2.948020e+07 | 11.552839 | 0.054041 | 0.52604 | 0.517824 | 0.921864 | 0.952447 | ... | 1.0 | 1.09198 |
| Y | 2.398762e+07 | 1.515128e+07 | 2885.457413 | 2.885794e+07 | 11.088766 | 0.027340 | 0.75184 | 0.531746 | 1.193878 | 1.090703 | ... | 1.0 | 1.07118 |

2 rows × 35 columns

# Python library SMOTE can be also used to populate samples

In [192]:
```python
df['FSV CMSI Flag'].value_counts(normalize = True)
```

Out[192]:
```
N    0.955444
Y    0.044556
Name: FSV CMSI Flag, dtype: float64
```

In [193]:
```python
yes = df.loc[df['FSV CMSI Flag'] == 'Y']
```

In [194]:
```
no = df.loc[df['FSV CMSI Flag'] == 'N']
```

In [195]:
```
no_sample = no.sample(951)
```

In [213]:
```
combined = pd.concat( [yes, no_sample]).dropna(subset = ['Total Cost'])
```

In [226]:
```
X = combined[['Total Cost']]
y = combined[['FSV CMSI Flag']]
```

In [227]:
```
from sklearn.linear_model import LogisticRegression
```

In [228]:
```
lgr = LogisticRegression()
```

In [229]:
```
lgr.fit(X,y)
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarnin
g: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\utils\validation.py:761: DataConversionWar
ning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for ex
ample using ravel().
  y = column_or_1d(y, warn=True)
```

Out[229]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [231]:
```
lgr.predict_proba(X)
```

Out[231]:
```
array([[0.45883936, 0.54116064],
       [0.43457182, 0.56542818],
       [0.44661667, 0.55338333],
       ...,
       [0.43900919, 0.56099081],
       [0.43900919, 0.56099081],
       [0.43457182, 0.56542818]])
```

In [232]:
```
lgr.score(X,y)
```

Out[232]:
```
0.5565280816921955
```

## A prediction algorithm can be evaluated by considering its predict probability and score.

In [ ]: