```
import numpy as np
import pandas as pd
import seaborn as sns
```

```
df = pd.read_csv('AdvWorksCusts.csv')
dff2 = pd.read_csv('AW_BikeBuyer.csv')
```

```
df.shape
```

```
(16519, 23)
```

```
dff2.shape
```

```
(16519, 2)
```

```
dff2.head()
```

	CustomerID	BikeBuyer
0	11000	0
1	11001	1
2	11002	0
3	11003	0
4	11004	1

```
df = df.merge(dff2,on=["CustomerID"])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16749 entries, 0 to 16748
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   CustomerID            16749 non-null  int64 
 1   Title                 88 non-null     object
 2   FirstName             16749 non-null  object
 3   MiddleName            9696 non-null   object
 4   LastName              16749 non-null  object
 5   Suffix                2 non-null      object
 6   AddressLine1          16749 non-null  object
 7   AddressLine2          281 non-null    object
 8   City                  16749 non-null  object
 9   StateProvinceName     16749 non-null  object
 10  CountryRegionName     16749 non-null  object
 11  PostalCode            16749 non-null  object
 12  PhoneNumber           16749 non-null  object
 13  BirthDate             16749 non-null  object
 14  Education             16749 non-null  object
 15  Occupation            16749 non-null  object
 16  Gender                16749 non-null  object
 17  MaritalStatus         16749 non-null  object
 18  HomeOwnerFlag         16749 non-null  int64 
 19  NumberCarsOwned       16749 non-null  int64 
 20  NumberChildrenAtHome  16749 non-null  int64 
 21  TotalChildren         16749 non-null  int64 
 22  YearlyIncome          16749 non-null  int64 
 23  BikeBuyer             16749 non-null  int64 
dtypes: int64(7), object(17)
memory usage: 3.2+ MB
```

```
df.head()
```

	CustomerID	Title	FirstName	MiddleName	LastName	Suffix	AddressLine1	AddressLine2	City
0	11000	NaN	Jon	V	Yang	NaN	3761 N. 14th St	NaN	Rockhampton
1	11001	NaN	Eugene	L	Huang	NaN	2243 W St.	NaN	Seaford
2	11002	NaN	Ruben	NaN	Torres	NaN	5844 Linden Land	NaN	Hobart
3	11003	NaN	Christy	NaN	Zhu	NaN	1825 Village Pl.	NaN	North Ryde

```
df.isnull().values.any()
```

```
True
```

```
df.isnull().sum()
```

```
CustomerID                 0
Title                  16661
FirstName                  0
MiddleName              7053
LastName                   0
Suffix                 16747
AddressLine1               0
AddressLine2           16468
City                       0
StateProvinceName          0
CountryRegionName          0
PostalCode                 0
PhoneNumber                0
BirthDate                  0
Education                  0
Occupation                 0
Gender                     0
MaritalStatus              0
HomeOwnerFlag              0
NumberCarsOwned            0
NumberChildrenAtHome       0
TotalChildren              0
YearlyIncome               0
BikeBuyer                  0
dtype: int64
```

```
#dropping duplicate value
df2=df.drop_duplicates()
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16429 entries, 0 to 16748
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   CustomerID            16429 non-null  int64 
 1   Title                 88 non-null     object
 2   FirstName             16429 non-null  object
 3   MiddleName            9465 non-null   object
 4   LastName              16429 non-null  object
 5   Suffix                2 non-null      object
 6   AddressLine1          16429 non-null  object
 7   AddressLine2          276 non-null    object
 8   City                  16429 non-null  object
 9   StateProvinceName     16429 non-null  object
 10  CountryRegionName     16429 non-null  object
 11  PostalCode            16429 non-null  object
 12  PhoneNumber           16429 non-null  object
 13  BirthDate             16429 non-null  object
 14  Education             16429 non-null  object
 15  Occupation            16429 non-null  object
 16  Gender                16429 non-null  object
 17  MaritalStatus         16429 non-null  object
 18  HomeOwnerFlag         16429 non-null  int64 
 19  NumberCarsOwned       16429 non-null  int64 
 20  NumberChildrenAtHome  16429 non-null  int64 
 21  TotalChildren         16429 non-null  int64 
 22  YearlyIncome          16429 non-null  int64 
 23  BikeBuyer             16429 non-null  int64 
dtypes: int64(7), object(17)
memory usage: 3.1+ MB
```

```
#removing all null value
df2 = df.dropna(axis=1)
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16749 entries, 0 to 16748
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   CustomerID            16749 non-null  int64 
 1   FirstName             16749 non-null  object
 2   LastName              16749 non-null  object
 3   AddressLine1          16749 non-null  object
 4   City                  16749 non-null  object
 5   StateProvinceName     16749 non-null  object
 6   CountryRegionName     16749 non-null  object
 7   PostalCode            16749 non-null  object
 8   PhoneNumber           16749 non-null  object
 9   BirthDate             16749 non-null  object
 10  Education             16749 non-null  object
 11  Occupation            16749 non-null  object
 12  Gender                16749 non-null  object
 13  MaritalStatus         16749 non-null  object
 14  HomeOwnerFlag         16749 non-null  int64 
 15  NumberCarsOwned       16749 non-null  int64 
 16  NumberChildrenAtHome  16749 non-null  int64 
 17  TotalChildren         16749 non-null  int64 
 18  YearlyIncome          16749 non-null  int64 
 19  BikeBuyer             16749 non-null  int64 
dtypes: int64(7), object(13)
memory usage: 2.7+ MB
```

```
df2.describe()
```

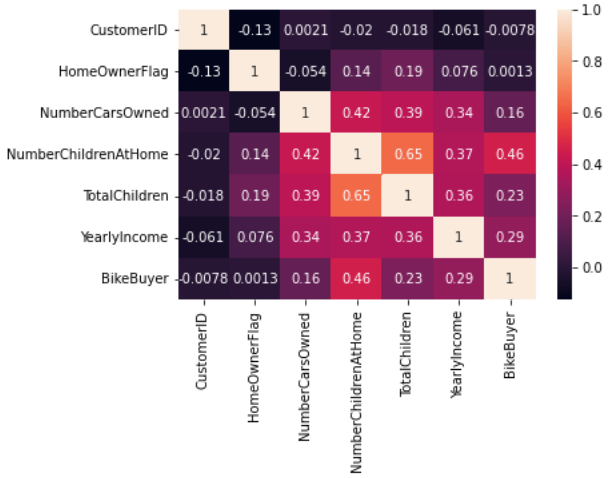
	CustomerID	HomeOwnerFlag	NumberCarsOwned	NumberChildrenAtHome	TotalChildren	YearlyIncome	Bike
	16749.000000	16749.000000	16749.000000	16749.000000	16749.000000	16749.000000	16749.0
	20222.633112	0.673473	1.503433	0.993791	2.009613	78109.602185	0.3
	5346.696692	0.468957	1.138620	1.516555	1.683549	39678.696234	0.4
	11000.000000	0.000000	0.000000	0.000000	0.000000	9482.000000	0.0
	15580.000000	0.000000	1.000000	0.000000	0.000000	47787.000000	0.0
	20200.000000	1.000000	2.000000	0.000000	2.000000	76120.000000	0.0
	24857.000000	1.000000	2.000000	2.000000	3.000000	105179.000000	1.0
	29482.000000	1.000000	4.000000	5.000000	5.000000	196511.000000	1.0

```
#correlation of whole table
d_cor=df2.corr()
d_cor
```

	CustomerID	HomeOwnerFlag	NumberCarsOwned	NumberChildrenAtHome	TotalChildren	Ye
CustomerID	1.000000	-0.126599	0.002115	-0.019848	-0.018416	
HomeOwnerFlag	-0.126599	1.000000	-0.053644	0.135171	0.188128	
NumberCarsOwned	0.002115	-0.053644	1.000000	0.424043	0.394739	
ımberChildrenAtHome	-0.019848	0.135171	0.424043	1.000000	0.647742	
TotalChildren	-0.018416	0.188128	0.394739	0.647742	1.000000	
YearlyIncome	-0.060852	0.076076	0.344480	0.369181	0.363084	
BikeBuyer	-0.007816	0.001302	0.164617	0.457332	0.233492	

```
sns.heatmap(d_cor, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fef68b63eb0>
```




```
#corelation between YearlyIncome and NumberCarsOwned
corr1 = df2['YearlyIncome'].corr(df2['NumberCarsOwned'])
print(corr1)
```

```
0.3448949340932411
```

```
#corelation between YearlyIncome and TotalChildren
corr2 = df2['YearlyIncome'].corr(df2['TotalChildren'])
print(corr2)
```

```
0.3626521990646709
```

```
#corelation between TotalChildren and NumberChildrenAtHome
corr3 = df2['TotalChildren'].corr(df2['NumberChildrenAtHome'])
print(corr3)
```

```
0.6475636644699255
```

```
#create a new data frame
df3 = df2[['CustomerID', 'HomeOwnerFlag', 'NumberCarsOwned', 'NumberChildrenAtHome', 'TotalChildren', 'YearlyIncome', 'BikeBuyer']]
```

```
df3.head()
```

	CustomerID	HomeOwnerFlag	NumberCarsOwned	NumberChildrenAtHome	TotalChildren	YearlyIncome	BikeB
0	11000	1	0	0	2	137947	
1	11001	0	1	3	3	101141	
2	11002	1	1	3	3	91945	
3	11003	0	1	0	0	86688	
4	11004	1	4	5	5	92771	

```
import pandas as pd
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB


# split the data into features and labels
X = df3.drop('BikeBuyer', axis=1)
y = df2['BikeBuyer']

# split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
# train a Naive Bayes model
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

# make predictions on the test set
y_pred = nb_model.predict(X_test)

# evaluation matrix
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred, average='weighted')
rec = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# print evaluation matrix
print("Accuracy:", acc)
print("Precision:", prec)
print("Recall:", rec)
print("F1 Score:", f1)
```

```
Accuracy: 0.7498507462686567
Precision: 0.7426191354066934
Recall: 0.7498507462686567
F1 Score: 0.726348488147498
```

Colab paid products - Cancel contracts here

✓ 0s completed at 15:27