In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
transaction=pd.read_csv("C:/Users/Akshay/Desktop/gaurav datascience application/datasets/transaction.csv")
transaction
```

Out[2]:

	transaction_id	product_id	customer_id	transaction_date	online_order	order_status	brand	product_line	product_class	product_size	l
0	1	2	2950	25-02-2017	False	Approved	Solex	Standard	medium	medium	
1	2	3	3120	21-05-2017	True	Approved	Trek Bicycles	Standard	medium	large	
2	3	37	402	16-10-2017	False	Approved	OHM Cycles	Standard	low	medium	
3	4	88	3135	31-08-2017	False	Approved	Norco Bicycles	Standard	medium	medium	
4	5	78	787	01-10-2017	True	Approved	Giant Bicycles	Standard	medium	large	
...	...	...	...	...	...	...	...	...	...	...	
19995	19996	51	1018	24-06-2017	True	Approved	OHM Cycles	Standard	high	medium	
19996	19997	41	127	09-11-2017	True	Approved	Solex	Road	medium	medium	
19997	19998	87	2284	14-04-2017	True	Approved	OHM Cycles	Standard	medium	medium	
19998	19999	6	2764	03-07-2017	False	Approved	OHM Cycles	Standard	high	medium	
19999	20000	11	1144	22-09-2017	True	Approved	Trek Bicycles	Standard	medium	small	

20000 rows × 13 columns

In [3]:
```
customeraddress=pd.read_csv("C:/Users/Akshay/Desktop/gaurav datascience application/datasets/customeraddress.csv")
customeraddress
```

Out[3]:

	customer_id	address	postcode	state	country	property_valuation
0	1	060 Morning Avenue	2016	New South Wales	Australia	10
1	2	6 Meadow Vale Court	2153	New South Wales	Australia	10
2	4	0 Holy Cross Court	4211	QLD	Australia	9
3	5	17979 Del Mar Point	2448	New South Wales	Australia	4
4	6	9 Oakridge Court	3216	VIC	Australia	9
...	...	...	...	...	...	...
3994	3999	1482 Hauk Trail	3064	VIC	Australia	3
3995	4000	57042 Village Green Point	4511	QLD	Australia	6
3996	4001	87 Crescent Oaks Alley	2756	NSW	Australia	10
3997	4002	8194 Lien Street	4032	QLD	Australia	7
3998	4003	320 Acker Drive	2251	NSW	Australia	7

3999 rows × 6 columns

In [4]:
```
newcustomerlist=pd.read_csv("C:/Users/Akshay/Desktop/gaurav datascience application/datasets/new customer list.csv")
newcustomerlist
```

Out[4]:

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title	job_industry_category	wealth_segment	deceas
0	Chickie	Brister	Male	86	12-07-1957	General Manager	Manufacturing	Mass Customer	
1	Morly	Genery	Male	69	22-03-1970	Structural Engineer	Property	Mass Customer	
2	Ardelis	Forrester	Female	10	28-08-1974	Senior Cost Accountant	Financial Services	Affluent Customer	
3	Lucine	Stutt	Female	64	28-01-1979	Account Representative III	Manufacturing	Affluent Customer	
4	Melinda	Hadlee	Female	34	21-09-1965	Financial Analyst	Financial Services	Affluent Customer	
...	...	...	...	...	...	...	...	...	
995	Ferdinand	Romanetti	Male	60	07-10-1959	Paralegal	Financial Services	Affluent Customer	
996	Burk	Wortley	Male	22	17-10-2001	Senior Sales Associate	Health	Mass Customer	
997	Melloney	Temby	Female	17	05-10-1954	Budget/Accounting Analyst IV	Financial Services	Affluent Customer	
998	Dickie	Cubbini	Male	30	17-12-1952	Financial Advisor	Financial Services	Mass Customer	
999	Sylas	Duffill	Male	56	02-10-1955	Staff Accountant IV	Property	Mass Customer	

1000 rows × 23 columns

In [5]:
```
customerd=pd.read_csv("C:/Users/Akshay/Desktop/gaurav datascience application/datasets/customerd.csv")
customerd
```

Out[5]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	job_title	job_industry_category	wealth_segment	de
0	1	Laraine	Medendorp	F	93	Executive Secretary	Health	Mass Customer	
1	2	Eli	Bockman	Male	81	Administrative Officer	Financial Services	Mass Customer	
2	3	Arlin	Dearle	Male	61	Recruiting Manager	Property	Mass Customer	
3	4	Talbot	NaN	Male	33	NaN	IT	Mass Customer	
4	5	Sheila-kathryn	Calton	Female	56	Senior Editor	NaN	Affluent Customer	
...	...	...	...	...	...	...	...	...	
3995	3996	Rosalia	Halgarth	Female	8	VP Product Management	Health	Mass Customer	
3996	3997	Blanch	Nisuis	Female	87	Statistician II	Manufacturing	High Net Worth	
3997	3998	Sarene	Woolley	U	60	Assistant Manager	IT	High Net Worth	
3998	3999	Patrizius	NaN	Male	11	NaN	Manufacturing	Affluent Customer	
3999	4000	Kippy	Oldland	Male	76	Software Engineer IV	NaN	Affluent Customer	

4000 rows × 11 columns

In [6]:
```
transaction.describe()
```

Out[6]:

	transaction_id	product_id	customer_id	list_price	product_first_sold_date
count	20000.000000	20000.00000	20000.000000	20000.000000	19803.000000
mean	10000.500000	45.36465	1738.246050	1107.829449	38199.776549
std	5773.647028	30.75359	1011.951046	582.825242	2875.201110
min	1.000000	0.00000	1.000000	12.010000	33259.000000
25%	5000.750000	18.00000	857.750000	575.270000	35667.000000
50%	10000.500000	44.00000	1736.000000	1163.890000	38216.000000
75%	15000.250000	72.00000	2613.000000	1635.300000	40672.000000
max	20000.000000	100.00000	5034.000000	2091.470000	42710.000000

In [7]:
```
customerd.describe()
```

Out[7]:

	customer_id	past_3_years_bike_related_purchases	tenure
count	4000.000000	4000.000000	3913.000000
mean	2000.500000	48.890000	10.657041
std	1154.844867	28.715005	5.660146
min	1.000000	0.000000	1.000000
25%	1000.750000	24.000000	6.000000
50%	2000.500000	48.000000	11.000000
75%	3000.250000	73.000000	15.000000
max	4000.000000	99.000000	22.000000

In [8]:
```
customeraddress.describe()
```

Out[8]:

	customer_id	postcode	property_valuation
count	3999.000000	3999.000000	3999.000000
mean	2003.987997	2985.755939	7.514379
std	1154.576912	844.878364	2.824663
min	1.000000	2000.000000	1.000000
25%	1004.500000	2200.000000	6.000000
50%	2004.000000	2768.000000	8.000000
75%	3003.500000	3750.000000	10.000000
max	4003.000000	4883.000000	12.000000

In [9]:
```
newcustomerlist.describe()
```

Out[9]:

	past_3_years_bike_related_purchases	tenure	postcode	property_valuation	Unnamed: 16	Unnamed: 17	Unnamed: 18	Unnamed: 19	
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1(
mean	49.836000	11.388000	3019.227000	7.397000	0.750240	0.842208	0.947672	0.875643	4
std	27.796686	5.037145	848.895767	2.758804	0.205775	0.250128	0.298312	0.285795	2
min	0.000000	0.000000	2000.000000	1.000000	0.400000	0.400000	0.400000	0.340000	
25%	26.750000	7.000000	2209.000000	6.000000	0.570000	0.640000	0.712500	0.650586	2
50%	51.000000	11.000000	2800.000000	8.000000	0.750000	0.820000	0.925000	0.840750	5
75%	72.000000	15.000000	3845.500000	9.000000	0.930000	1.037500	1.164844	1.073594	7
max	99.000000	22.000000	4879.000000	12.000000	1.100000	1.375000	1.718750	1.718750	1(

In [10]:
```
df=pd.merge(customerd,customeraddress)
df
```

Out[10]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	job_title	job_industry_category	wealth_segment	de
0	1	Laraine	Medendorp	F	93	Executive Secretary	Health	Mass Customer	
1	2	Eli	Bockman	Male	81	Administrative Officer	Financial Services	Mass Customer	
2	4	Talbot	NaN	Male	33	NaN	IT	Mass Customer	
3	5	Sheila-kathryn	Calton	Female	56	Senior Editor	NaN	Affluent Customer	
4	6	Curr	Duckhouse	Male	35	NaN	Retail	High Net Worth	
...	...	...	...	...	...	...	...	...	
3991	3996	Rosalia	Halgarth	Female	8	VP Product Management	Health	Mass Customer	
3992	3997	Blanch	Nisuis	Female	87	Statistician II	Manufacturing	High Net Worth	
3993	3998	Sarene	Woolley	U	60	Assistant Manager	IT	High Net Worth	
3994	3999	Patrizius	NaN	Male	11	NaN	Manufacturing	Affluent Customer	
3995	4000	Kippy	Oldland	Male	76	Software Engineer IV	NaN	Affluent Customer	

3996 rows × 16 columns

In [11]:
```
df2=pd.merge(transaction,df)
df2.columns
```

Out[11]:
```
Index(['transaction_id', 'product_id', 'customer_id', 'transaction_date',
       'online_order', 'order_status', 'brand', 'product_line',
       'product_class', 'product_size', 'list_price', 'standard_cost',
       'product_first_sold_date', 'first_name', 'last_name', 'gender',
       'past_3_years_bike_related_purchases', 'job_title',
       'job_industry_category', 'wealth_segment', 'deceased_indicator',
       'owns_car', 'tenure', 'address', 'postcode', 'state', 'country',
       'property_valuation'],
      dtype='object')
```

In [12]:
```
df2
```

Out[12]:

	transaction_id	product_id	customer_id	transaction_date	online_order	order_status	brand	product_line	product_class	product_size	.
0	1	2	2950	25-02-2017	False	Approved	Solex	Standard	medium	medium	.
1	11065	1	2950	16-10-2017	False	Approved	Giant Bicycles	Standard	medium	medium	.
2	18923	62	2950	26-04-2017	False	Approved	Solex	Standard	medium	medium	.
3	2	3	3120	21-05-2017	True	Approved	Trek Bicycles	Standard	medium	large	.
4	6862	4	3120	05-10-2017	False	Approved	Giant Bicycles	Standard	high	medium	.
...	...	...	...	...	...	...	...	...	...	...	.
19963	19854	68	130	02-02-2017	True	Approved	OHM Cycles	Standard	medium	medium	.
19964	17966	17	2789	06-12-2017	False	Approved	Solex	Standard	high	medium	.
19965	18462	80	2789	20-06-2017	False	Approved	OHM Cycles	Touring	low	medium	.
19966	17981	69	3446	26-12-2017	True	Approved	Giant Bicycles	Road	medium	medium	.
19967	18165	86	3446	03-12-2017	False	Approved	OHM Cycles	Standard	medium	medium	.

19968 rows × 28 columns

In [13]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19968 entries, 0 to 19967
Data columns (total 28 columns):
transaction_id                          19968 non-null int64
product_id                              19968 non-null int64
customer_id                             19968 non-null int64
transaction_date                        19968 non-null object
online_order                            19609 non-null object
order_status                            19968 non-null object
brand                                   19773 non-null object
product_line                            19773 non-null object
product_class                           19773 non-null object
product_size                            19773 non-null object
list_price                              19968 non-null float64
standard_cost                           19773 non-null object
product_first_sold_date                 19773 non-null float64
first_name                              19968 non-null object
last_name                               19326 non-null object
gender                                  19968 non-null object
past_3_years_bike_related_purchases     19968 non-null int64
job_title                               17589 non-null object
job_industry_category                   16746 non-null object
wealth_segment                          19968 non-null object
deceased_indicator                      19968 non-null object
owns_car                                19968 non-null object
tenure                                  19522 non-null float64
address                                 19968 non-null object
postcode                                19968 non-null int64
state                                   19968 non-null object
country                                 19968 non-null object
property_valuation                      19968 non-null int64
dtypes: float64(3), int64(6), object(19)
memory usage: 4.4+ MB
```

In [14]:
```
df2['brand'] = df2.brand.astype(str)
df2['product_line'] = df2.product_line.astype(str)
df2['product_size'] = df2.product_size.astype(str)
df2['product_class'] = df2.product_class.astype(str)
df2['job_industry_category'] = df2.job_industry_category.astype(str)
df2['job_title'] = df2.job_title.astype(str)
df2['first_name'] = df2.first_name.astype(str)
df2['last_name'] = df2.last_name.astype(str)
df2['transaction_date'] = df2.last_name.astype(str)
```

In [17]:
```
from sklearn.preprocessing import LabelEncoder
le_online_order=LabelEncoder()
le_order_status=LabelEncoder()
le_brand=LabelEncoder()
le_product_line=LabelEncoder()
le_product_size=LabelEncoder()
le_product_class=LabelEncoder()
le_job_industry_category=LabelEncoder()
le_wealth_segment=LabelEncoder()
le_deceased_indicator=LabelEncoder()
le_owns_car=LabelEncoder()
le_country=LabelEncoder()
le_address=LabelEncoder()
le_postcode=LabelEncoder()
le_state=LabelEncoder()
le_first_name=LabelEncoder()
le_last_name=LabelEncoder()
le_gender=LabelEncoder()
le_job_title=LabelEncoder()
le_transaction_date=LabelEncoder()
```

In [18]:
```
df2['online_order_n']=le_online_order.fit_transform(df2['online_order'])
df2['order_status_n']=le_order_status.fit_transform(df2['order_status'])
df2['brand_n']=le_brand.fit_transform(df2['brand'])
df2['product_line_n']=le_product_line.fit_transform(df2['product_line'])
df2['product_size_n']=le_product_size.fit_transform(df2['product_size'])
df2['product_class_n']=le_product_class.fit_transform(df2['product_class'])
df2['job_industry_category_n']=le_job_industry_category.fit_transform(df2['job_industry_category'])
df2['wealth_segment_n']=le_wealth_segment.fit_transform(df2['wealth_segment'])
df2['deceased_indicator_n']=le_deceased_indicator.fit_transform(df2['deceased_indicator'])
df2['owns_car_n']=le_owns_car.fit_transform(df2['owns_car'])
df2['address_n']=le_address.fit_transform(df2['address'])
df2['country_n']=le_country.fit_transform(df2['country'])
df2['state_n']=le_state.fit_transform(df2['state'])
df2['postcode_n']=le_postcode.fit_transform(df2['postcode'])
df2['first_name_n']=le_first_name.fit_transform(df2['first_name'])
df2['last_name_n']=le_last_name.fit_transform(df2['last_name'])
df2['gender_n']=le_gender.fit_transform(df2['gender'])
df2['job_title_n']=le_job_title.fit_transform(df2['job_title'])
df2['transaction_date_n']=le_transaction_date.fit_transform(df2['transaction_date'])
```

In [ ]:

In [ ]:

In [19]:
```
df3=df2.drop(['online_order', 'order_status','brand','product_line','product_size','product_class','job_industry_category','wealth_segment','deceased_indicator','owns_car','address','country','state','postcode','first_name','last_name','gender','job_title','transaction_date'], axis=1)

df3
```

Out[19]:

	transaction_id	product_id	customer_id	list_price	standard_cost	product_first_sold_date	past_3_years_bike_related_purchases	tenure	pr
0	1	2	2950	71.49	$53.62	41245.0	19	10.0	
1	11065	1	2950	1403.50	$954.82	37659.0	19	10.0	
2	18923	62	2950	478.16	$298.72	40487.0	19	10.0	
3	2	3	3120	2091.47	$388.92	41701.0	89	10.0	
4	6862	4	3120	1129.13	$677.48	40649.0	89	10.0	
...	...	...	...	...	...	...	...	...	
19963	19854	68	130	1636.90	$44.71	40410.0	32	1.0	
19964	17966	17	2789	1024.66	$614.80	35378.0	66	7.0	
19965	18462	80	2789	1073.07	$933.84	42226.0	66	7.0	
19966	17981	69	3446	792.90	$594.68	33879.0	8	14.0	
19967	18165	86	3446	235.63	$125.07	38206.0	8	14.0	

19968 rows × 28 columns

In [20]:
```
df3.fillna(df3.mean())
```

Out[20]:

	transaction_id	product_id	customer_id	list_price	standard_cost	product_first_sold_date	past_3_years_bike_related_purchases	tenure	pr
0	1	2	2950	71.49	$53.62	41245.0	19	10.0	
1	11065	1	2950	1403.50	$954.82	37659.0	19	10.0	
2	18923	62	2950	478.16	$298.72	40487.0	19	10.0	
3	2	3	3120	2091.47	$388.92	41701.0	89	10.0	
4	6862	4	3120	1129.13	$677.48	40649.0	89	10.0	
...	...	...	...	...	...	...	...	...	
19963	19854	68	130	1636.90	$44.71	40410.0	32	1.0	
19964	17966	17	2789	1024.66	$614.80	35378.0	66	7.0	
19965	18462	80	2789	1073.07	$933.84	42226.0	66	7.0	
19966	17981	69	3446	792.90	$594.68	33879.0	8	14.0	
19967	18165	86	3446	235.63	$125.07	38206.0	8	14.0	

19968 rows × 28 columns

In [23]:
```
df3.to_csv('C:/Users/Akshay/Desktop/gaurav datascience application/coding/internship/file1.csv')
df3
```

Out[23]:

	transaction_id	product_id	customer_id	list_price	standard_cost	product_first_sold_date	past_3_years_bike_related_purchases	tenure	pr
0	1	2	2950	71.49	$53.62	41245.0	19	10.0	
1	11065	1	2950	1403.50	$954.82	37659.0	19	10.0	
2	18923	62	2950	478.16	$298.72	40487.0	19	10.0	
3	2	3	3120	2091.47	$388.92	41701.0	89	10.0	
4	6862	4	3120	1129.13	$677.48	40649.0	89	10.0	
...	...	...	...	...	...	...	...	...	
19963	19854	68	130	1636.90	$44.71	40410.0	32	1.0	
19964	17966	17	2789	1024.66	$614.80	35378.0	66	7.0	
19965	18462	80	2789	1073.07	$933.84	42226.0	66	7.0	
19966	17981	69	3446	792.90	$594.68	33879.0	8	14.0	
19967	18165	86	3446	235.63	$125.07	38206.0	8	14.0	

19968 rows × 28 columns