```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from scipy.stats import zscore
from sklearn.preprocessing import Imputer
from sklearn.metrics import accuracy_score
import seaborn as sns
import os
%matplotlib inline
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
data = pd.read_csv('/content/drive/My Drive/Colab Notebooks/Project18aug/data.csv')
```

```
data
```

	City1	City2	Average Fare	Distance	Average weekly passengers	market leading airline	market share	Average fare	Low price airline
0	CAK	ATL	114.47	528	424.56	FL	70.19	111.03	FL
1	CAK	MCO	122.47	860	276.84	FL	75.10	123.09	DL
2	ALB	ATL	214.42	852	215.76	DL	78.89	223.98	CO
3	ALB	BWI	69.40	288	606.84	WN	96.97	68.86	WN
4	ALB	ORD	158.13	723	313.04	UA	39.79	161.36	WN
5	ALB	FLL	135.17	1204	199.02	WN	40.68	137.97	DL
6	ALB	LAS	152.85	2237	237.17	WN	59.94	148.59	WN
7	ALB	LAX	190.73	2467	191.95	DL	17.89	205.06	US
8	ALB	MCO	129.35	1073	550.54	WN	76.84	127.69	WN
9	ALB	TPA	134.17	1130	202.93	US	35.40	132.91	DL
10	ABQ	ATL	212.49	1269	198.80	DL	68.39	226.79	AA
11	ABQ	BWI	173.56	1670	312.39	WN	49.16	180.49	AA
12	ABQ	ORD	170.67	1121	364.78	AA	45.94	174.62	WN
13	ABQ	DFW	120.24	580	839.78	WN	71.91	117.20	WN
14	ABQ	DEN	168.69	349	308.26	UA	59.55	181.34	F9
15	ABQ	IAH	154.40	767	372.93	WN	50.48	152.03	WN
16	ABQ	LAS	114.24	487	620.86	WN	93.92	113.82	WN
17	ABQ	LAX	132.29	677	655.00	WN	89.46	130.44	WN
18	ABQ	MSP	181.99	981	187.28	NW	65.00	182.27	CO
19	ABQ	LGA	233.05	1825	344.45	AA	31.33	233.26	DL
20	ABQ	OAK	162.21	889	388.15	WN	88.63	164.27	HP
21	ABQ	MCO	161.74	1552	190.65	WN	72.29	151.81	WN
22	ABQ	PHX	71.57	328	1252.39	WN	77.65	70.99	WN
23	ABQ	PDX	163.63	1111	222.93	WN	57.24	167.46	HP
24	ABQ	SAN	134.42	628	346.30	WN	82.92	136.70	HP
25	ABQ	SEA	165.69	1180	284.34	WN	47.82	164.81	HP
26	ABQ	TUS	77.82	321	191.19	WN	86.29	73.50	WN
27	ABQ	IAD	227.93	1650	301.84	AA	46.59	213.13	AA
28	AMA	DFW	74.82	324	615.10	WN	82.04	71.51	WN
29	AMA	IAH	120.50	545	229.78	WN	58.79	119.76	CO
...	...	...	...	...	...	...	...	...	...

970	SLC	SEA	110.64	689	892.39	DL	60.93	111.80	WN
971	SLC	GEG	98.36	546	204.56	DL	62.27	97.48	DL
972	SLC	IAD	263.37	1851	305.97	DL	68.95	268.78	DL
973	SAT	SAN	165.13	1129	248.80	WN	57.92	165.01	AA
974	SAT	SEA	177.77	1774	229.89	AA	26.71	169.59	AA
975	SAT	IAD	215.04	1381	345.10	DL	48.59	191.34	DL
976	SAN	SFO	122.02	447	937.93	UA	95.92	122.11	AA
977	SAN	SJC	82.34	417	2176.19	WN	71.53	83.52	AA
978	SAN	SEA	148.62	1050	1303.15	AS	73.50	149.91	AS
979	SAN	TPA	177.91	2087	273.47	DL	27.10	172.14	WN
980	SAN	TUS	74.62	367	469.67	WN	94.07	73.10	WN
981	SAN	IAD	330.21	2276	582.17	UA	42.55	440.35	AA
982	SFO	SNA	134.18	372	703.36	UA	66.35	140.50	AA
983	SFO	SEA	116.78	678	1545.54	UA	52.32	122.59	AS
984	SFO	TPA	237.26	2392	193.26	DL	21.54	265.35	CO
985	SFO	IAD	401.23	2442	955.65	UA	64.45	490.03	TZ
986	SJC	SNA	77.11	342	1970.76	WN	49.97	77.54	AA
987	SJC	SEA	105.84	697	1390.65	AS	75.35	106.30	WN
988	SJC	TUS	144.22	721	204.56	AS	56.16	143.53	AS
989	SJC	IAD	322.83	2424	229.02	UA	37.01	408.10	TZ
990	SNA	SEA	156.01	978	1191.84	AS	86.77	157.43	DL
991	SEA	GEG	70.61	224	1423.15	AS	68.40	72.60	WN
992	SEA	TPA	162.46	2520	312.93	DL	33.03	147.60	AA
993	SEA	TUS	131.47	1216	359.23	AS	73.25	126.38	AS
994	SEA	IAD	288.14	2329	787.50	UA	46.51	329.20	AS
995	SYR	TPA	136.16	1104	184.34	US	33.37	135.82	DL
996	TLH	TPA	83.28	200	232.71	FL	99.57	82.55	FL
997	TPA	IAD	159.97	814	843.80	US	46.19	159.65	DL
998	TPA	PBI	73.57	174	214.45	WN	99.74	73.44	WN
999	IAD	PBI	126.67	859	475.65	US	56.28	129.92	DL

```
data.drop(["City1 ", "City2    "  , "market leading airline","Low price airline"], axis = 1, inpla
```

```
data
```

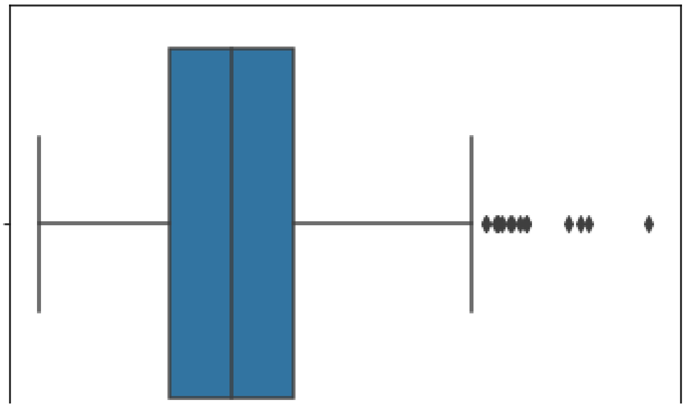
	Average Fare	Distance	Average weekly passengers	market share	Average fare	market
0	114.47	528	424.56	70.19	111.03	
1	122.47	860	276.84	75.10	123.09	
2	214.42	852	215.76	78.89	223.98	
3	69.40	288	606.84	96.97	68.86	
4	158.13	723	313.04	39.79	161.36	
5	135.17	1204	199.02	40.68	137.97	
6	152.85	2237	237.17	59.94	148.59	
7	190.73	2467	191.95	17.89	205.06	
8	129.35	1073	550.54	76.84	127.69	
9	134.17	1130	202.93	35.40	132.91	
10	212.49	1269	198.80	68.39	226.79	
11	173.56	1670	312.39	49.16	180.49	
12	170.67	1121	364.78	45.94	174.62	
13	120.24	580	839.78	71.91	117.20	
14	168.69	349	308.26	59.55	181.34	
15	154.40	767	372.93	50.48	152.03	
16	114.24	487	620.86	93.92	113.82	
17	132.29	677	655.00	89.46	130.44	
18	181.99	981	187.28	65.00	182.27	
19	233.05	1825	344.45	31.33	233.26	
20	162.21	889	388.15	88.63	164.27	
21	161.74	1552	190.65	72.29	151.81	
22	71.57	328	1252.39	77.65	70.99	
23	163.63	1111	222.93	57.24	167.46	
24	134.42	628	346.30	82.92	136.70	
25	165.69	1180	284.34	47.82	164.81	
26	77.82	321	191.19	86.29	73.50	
27	227.93	1650	301.84	46.59	213.13	
28	74.82	324	615.10	82.04	71.51	
29	120.50	545	229.78	58.79	119.76	
...	...	...	...	...	...	
970	110.64	689	892.39	60.93	111.80	

971	98.36	546	204.56	62.27	97.48
972	263.37	1851	305.97	68.95	268.78
973	165.13	1129	248.80	57.92	165.01
974	177.77	1774	229.89	26.71	169.59
975	215.04	1381	345.10	48.59	191.34
976	122.02	447	937.93	95.92	122.11
977	82.34	417	2176.19	71.53	83.52
978	148.62	1050	1303.15	73.50	149.91
979	177.91	2087	273.47	27.10	172.14
980	74.62	367	469.67	94.07	73.10
981	330.21	2276	582.17	42.55	440.35
982	134.18	372	703.36	66.35	140.50
983	116.78	678	1545.54	52.32	122.59
984	237.26	2392	193.26	21.54	265.35
985	401.23	2442	955.65	64.45	490.03
986	77.11	342	1970.76	49.97	77.54
987	105.84	697	1390.65	75.35	106.30
988	144.22	721	204.56	56.16	143.53
989	322.83	2424	229.02	37.01	408.10
990	156.01	978	1191.84	86.77	157.43
991	70.61	224	1423.15	68.40	72.60
992	162.46	2520	312.93	33.03	147.60
993	131.47	1216	359.23	73.25	126.38
994	288.14	2329	787.50	46.51	329.20
995	136.16	1104	184.34	33.37	135.82
996	83.28	200	232.71	99.57	82.55
997	159.97	814	843.80	46.19	159.65
998	73.57	174	214.45	99.74	73.44
999	126.67	859	475.65	56.28	129.92

1000 rows × 7 columns

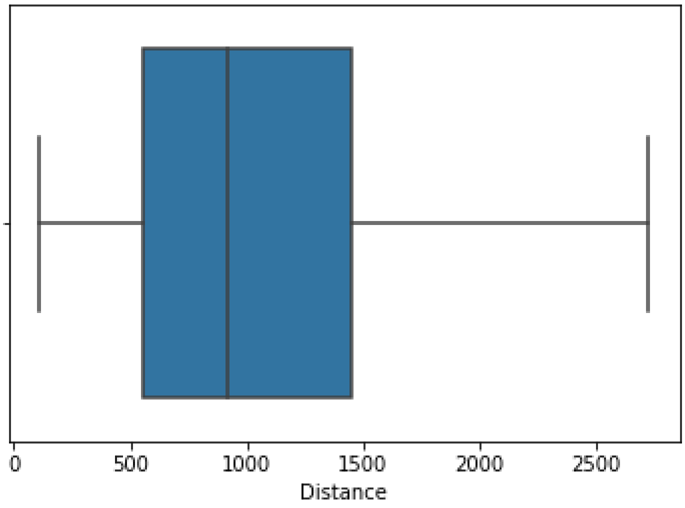
```
sns.boxplot(x=data['Average Fare '])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba06c694a8>
```



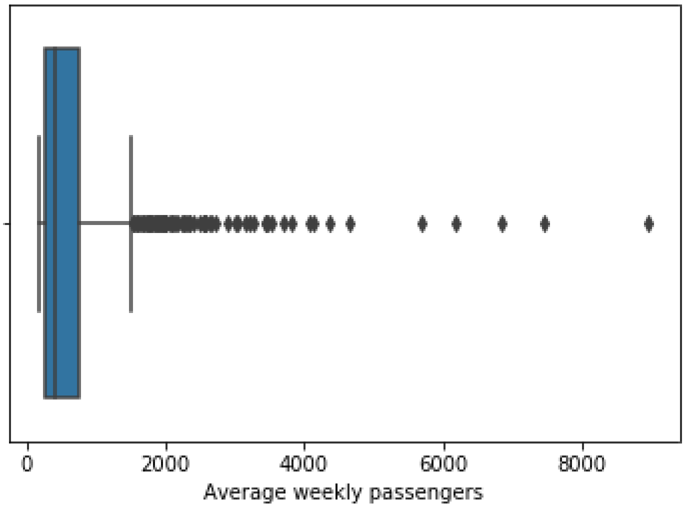
```
sns.boxplot(x=data['Distance'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba03f49588>
```



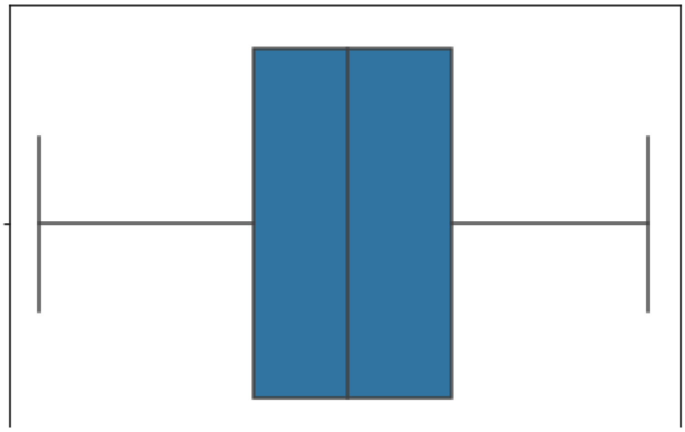
```
sns.boxplot(x=data['Average weekly passengers'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba03f17668>
```



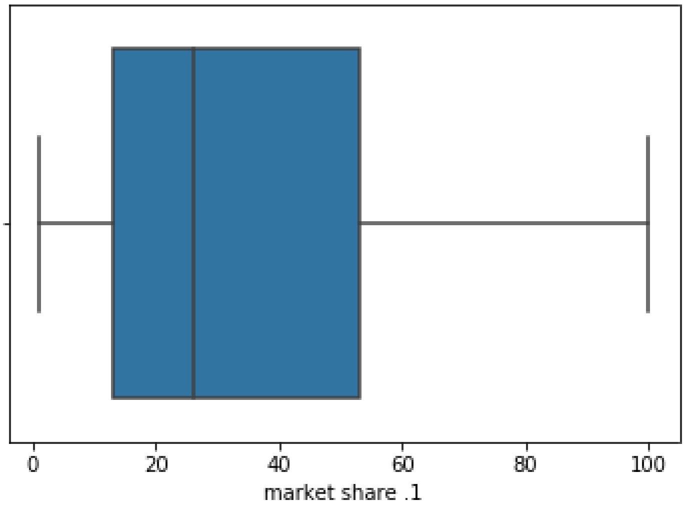
```
sns.boxplot(x=data['market share '])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba03e85828>
```



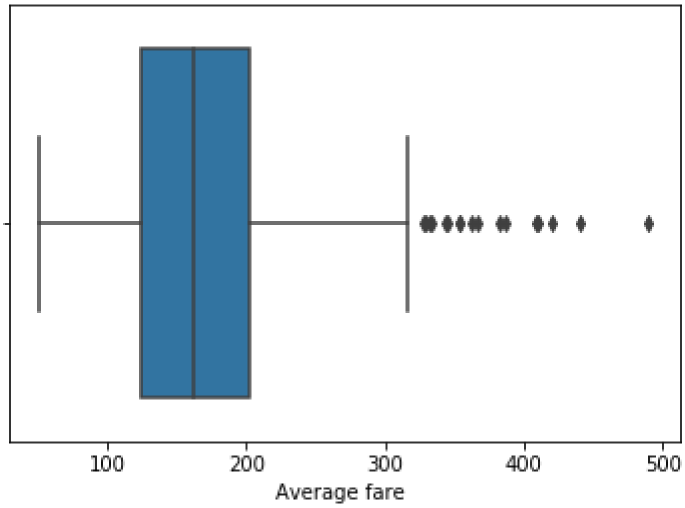
```
sns.boxplot(x=data['market share .1'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba03e57a20>
```



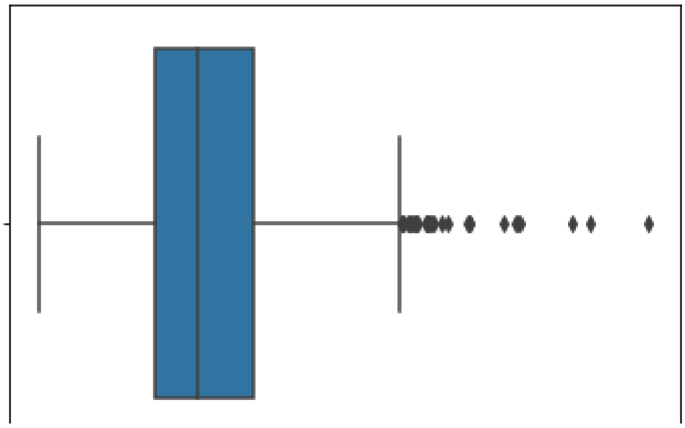
```
sns.boxplot(x=data['Average fare '])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba03e35ef0>
```



```
sns.boxplot(x=data['price '])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba03d7ab70>
```



```
Q1 = np.percentile(data, 25, interpolation = 'midpoint')


Q3 = np.percentile(data, 75, interpolation = 'midpoint')


IQR = Q3 - Q1


print(IQR)
```

```
200.38
```

```
def remove_outlier(df_in, col_name):
    q1 = df_in[col_name].quantile(0.25)
    q3 = df_in[col_name].quantile(0.75)
    iqr = q3-q1 #Interquartile range
    fence_low  = q1-1.5*iqr
    fence_high = q3+1.5*iqr
    df_out = df_in.loc[(df_in[col_name] > fence_low) & (df_in[col_name] < fence_high)]
    return df_out
```

```
from scipy import stats
import numpy as np
```

```
z = np.abs(stats.zscore(data))
print(z)
```

```
[[0.88376186 0.82281562 0.32333577 ... 0.88805083 1.31587826 0.68134108]
 [0.73919517 0.3063908  0.51614751 ... 0.69548767 0.63955964 0.51377609]
 [0.92241827 0.31883477 0.59587227 ... 0.91543248 1.17346508 0.50686631]
 ...
 [0.06153879 0.37794364 0.22387785 ... 0.1117307  0.7628822  0.33527637]
 [1.62285908 1.37346137 0.59758215 ... 1.48825391 2.40695059 1.47764552]
 [0.66329765 0.3079463  0.25665047 ... 0.58643242 0.1483755  0.45022426]]
```

```
data_o = data[(z < 3).all(axis=1)]
print(data_o)
```

```
     Average Fare  Distance  ...  market share .1  price
0           114.47       528  ...            70.19  111.03
1           122.47       860  ...            17.23  118.94
2           214.42       852  ...             2.77  167.12
3            69.40       288  ...            96.97   68.86
4           158.13       723  ...            15.34  145.42
5           135.17      1204  ...            17.09  127.69
6           152.85      2237  ...            59.94  148.59
7           190.73      2467  ...            16.59  174.00
8           129.35      1073  ...            76.84  127.69
9           134.17      1130  ...            26.40  124.78
10          212.49      1269  ...            11.91  200.93
11          173.56      1670  ...            14.37  161.39
12          170.67      1121  ...            33.87  163.22
13          120.24       580  ...            71.91  117.20
14          168.69       349  ...            39.95  149.41
15          154.40       767  ...            50.48  152.03
16          114.24       487  ...            93.92  113.82
17          132.29       677  ...            89.46  130.44
18          181.99       981  ...             2.37  109.14
19          233.05      1825  ...            19.50  222.08
20          162.21       889  ...             7.19  144.51
21          161.74      1552  ...            72.29  151.81
22           71.57       328  ...            77.65   70.99
23          163.63      1111  ...            20.77  155.82
24          134.42       628  ...            13.05  119.03
25          165.69      1180  ...            25.07  159.64
26           77.82       321  ...            86.29   73.50
27          227.93      1650  ...            46.59  213.13
28           74.82       324  ...            82.04   71.51
29          120.50       545  ...            37.55  118.05
..             ...       ...  ...              ...     ...
967         142.71       599  ...            33.83  133.85
968         143.99       585  ...            17.23  114.46
969         102.95       588  ...            89.93  101.01
970         110.64       689  ...            36.44  105.36
971          98.36       546  ...            62.27   97.48
972         263.37      1851  ...            68.95  268.78
973         165.13      1129  ...            12.45  160.60
974         177.77      1774  ...            26.71  169.59
975         215.04      1381  ...            48.59  191.34
976         122.02       447  ...             1.35  102.68
977          82.34       417  ...            26.94   79.37
978         148.62      1050  ...            73.50  149.91
979         177.91      2087  ...            15.38  165.25
980          74.62       367  ...            94.07   73.10
982         134.18       372  ...            30.82  119.80
983         116.78       678  ...            45.62  109.45
984         237.26      2392  ...            16.02  200.96
986          77.11       342  ...            48.91   76.66
987         105.84       697  ...            22.04  103.02
988         144.22       721  ...            56.16  143.53
990         156.01       978  ...             2.19  100.08
991          70.61       224  ...            31.30   66.32
992         162.46      2520  ...            23.75  141.21
993         131.47      1216  ...            73.25  126.38
994         288.14      2329  ...            30.43  252.34
995         136.16      1104  ...            28.65  118.51
996          83.28       200  ...            99.57   82.55
997         159.97       814  ...            13.89  159.02
998          73.57       174  ...            99.74   73.44
```

```
999          126.67       859  ...            38.57  121.94

[966 rows x 7 columns]
```

```
data.describe().transpose()
```

```
  #Treat “Average Fare” – 3rdColumn as your Dependent Variable and Rest of the columns as Indepen
X = data_o.drop('Average fare ', axis=1)
y = data_o[['Average fare ']]
```

```
y
```

```
data_o.corr(method ='pearson')
```

```
#Drop the independent variables which has less than 0.1 correlation with the dependent variable
data_o.drop(["Average weekly passengers", "market share "], axis = 1, inplace = True)
```

```
data_o
```

```
#Create scatter Plot of Independent Variable vs Dependent Variable
ax1 = data_o.plot.scatter(x='Average Fare ',y='Distance', c='DarkBlue')
```

```
#Create scatter Plot of Independent Variable vs Dependent Variable
  ax2 = data_o.plot.scatter(x='Average Fare ',y='price ', c='DarkBlue')
```

```
import numpy as np
import matplotlib.pyplot as plt  # To visualize
import pandas as pd  # To read data
from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=1)


linear_regressor = LinearRegression()  # create object for the class
linear_regressor.fit(X_train, y_train)  # perform linear regression
Y_pred = linear_regressor.predict(X)  # make predictions


for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, linear_regressor.coef_[0][idx]))
```

```
plt.plot(y, Y_pred, color='red')
plt.show()
```

```
print('intercept:', linear_regressor.intercept_)
```

```
r_sq = linear_regressor.score(X, y)
print('coefficient of determination:', r_sq)
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
Dataframe.score(y, Y_pred)
```

```
acc=accuracy_score(y, Y_pred)
```