CoxAssignment05

Task: This assignment will help you understand multiple linear regression. You will also be required to implement these evaluation metrics.

Dataset: Airfares.csv

In [52]:
```
# import required functionality
import pandas as pd
import numpy as np


import matplotlib.pylab as plt

# may need to install dmba in your python env --> in anaconda powershell type pip inst
from dmba import regressionSummary, exhaustive_search
from dmba import adjusted_r2_score, AIC_score

%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Multiple Linear Regression

Please use the file Airfares.csv, which contains real data that were collected between Q3-1996 and Q2-1997.

The following problem takes place in the United States in the late 1990s, when many major US cities were facing issues with airport congestion, partly as a result of the 1978 deregulation of airlines. Both fares and routes were freed from regulation, and low-fare carriers such as Southwest (SW) began competing on existing routes and starting nonstop service on routes that previously lacked it. Building completely new airports is generally not feasible, but sometimes decommissioned military bases or smaller municipal airports can be reconfgured as regional or larger commercial airports. There are numerous players and interests involved in the issue (airlines, city, state and federal authorities, civic groups, the military, airport operators), and an aviation consulting firm is seeking advisory contracts with these players. The firm needs predictive models to support its consulting service. One thing the firm might want to be able to predict is fares, in the event a new airport is brought into service. The variables in these data are listed in below, and are believed to be important in predicting FARE. Some airport-to-airport data are available, but most data are at the city-to-city level. One question that will be of interest in the analysis is the effect that the presence or absence of Southwest has on FARE.

S_CODE	Starting airport's code
S_CITY	Starting city
E_CODE	Ending airport's code
E_CITY	Ending city
COUPON	Average number of coupons (a one-coupon flight is a non-stop flight, A two-coupon flight is a one stop flight, etc.) for that route
NEW	Number of new carriers entering that route between Q3-96 and Q2-97
VACATION	Whether a vacation route (Yes) or not (No).
SW	Whether Southwest Airlines serves that route (Yes) or not (No)
HI	Herfindel Index - measure of market concentration
S_INCOME	Starting city's average personal income
E_INCOME	Ending city's average personal income
S_POP	Starting city's population
E_POP	Ending city's population
SLOT	Whether either endpoint airport is slot controlled or not; This is a measure of airport congestion
GATE	Whether either endpoint airport has gate constraints or not; This is another measure of airport congestion
DISTANCE	Distance between two endpoint airports in miles
PAX	Number of passengers on that route during period of data collection
FARE (the response)	Average fare on that route

Exploratory Regression

Load the data and print the dimension, the first five records, and the data types.

In [53]:
```
air = pd.read_csv('Airfares.csv')
air.head(5)
```

Out[53]:

	S_CODE	S_CITY	E_CODE	E_CITY	COUPON	NEW	VACATION	SW	HI	S_INCOME
0	*	Dallas/Fort Worth TX	*	Amarillo TX	1.00	3	No	Yes	5291.99	28637.0
1	*	Atlanta GA	*	Baltimore/Wash Intl MD	1.06	3	No	No	5419.16	26993.0
2	*	Boston MA	*	Baltimore/Wash Intl MD	1.06	3	No	No	9185.28	30124.0
3	ORD	Chicago IL	*	Baltimore/Wash Intl MD	1.06	3	No	Yes	2657.35	29260.0
4	MDW	Chicago IL	*	Baltimore/Wash Intl MD	1.06	3	No	Yes	2657.35	29260.0

In [54]:
```
air.shape
```

Out[54]:
```
(638, 18)
```

In [55]:
```
air.dtypes
```

Out[55]:
```
S_CODE       object
S_CITY       object
E_CODE       object
E_CITY       object
COUPON      float64
NEW           int64
VACATION     object
SW           object
HI          float64
S_INCOME    float64
E_INCOME    float64
S_POP         int64
E_POP         int64
SLOT         object
GATE         object
DISTANCE      int64
PAX           int64
FARE        float64
dtype: object
```

Create an object holding the numerical variables in a list. Please exclude the first four variables 'S_CODE', 'S_CITY', 'E_CODE', 'E_CITY'. Print the new list.

In [56]:
```
predictors = ['COUPON', 'NEW', 'VACATION', 'SW', 'HI', 'S_INCOME', 'E_INCOME', 'S_POP'
             'E_POP', 'SLOT', 'GATE', 'DISTANCE', 'PAX']
outcome = 'FARE'
```

In [57]:
```
X = pd.get_dummies(air[predictors], drop_first=True)
y = air[outcome]
```

In [58]:
```
X
```

Out[58]:

	COUPON	NEW	HI	S_INCOME	E_INCOME	S_POP	E_POP	DISTANCE	PAX	VACATIO
0	1.00	3	5291.99	28637.0	21112.0	3036732	205711	312	7864	
1	1.06	3	5419.16	26993.0	29838.0	3532657	7145897	576	8820	
2	1.06	3	9185.28	30124.0	29838.0	5787293	7145897	364	6452	
3	1.06	3	2657.35	29260.0	29838.0	7830332	7145897	612	25144	
4	1.06	3	2657.35	29260.0	29838.0	7830332	7145897	612	25144	
...	...	...	...	...	...	...	...	...	...	
633	1.08	3	2216.70	32991.0	37375.0	8621121	991717	1030	34324	
634	1.08	0	2216.70	32991.0	37375.0	8621121	991717	1030	34324	
635	1.17	3	6797.80	27994.0	37375.0	4948339	991717	960	6016	
636	1.28	3	5566.43	31981.0	37375.0	4549784	991717	858	4877	
637	1.28	3	5566.43	31981.0	37375.0	4549784	991717	858	4877	

638 rows × 13 columns

Remove the categorical variables from the list (object datatypes).

In [73]:
```
air2 = air.drop(['VACATION', 'SW', 'SLOT', 'GATE'], axis=1)
air2
```

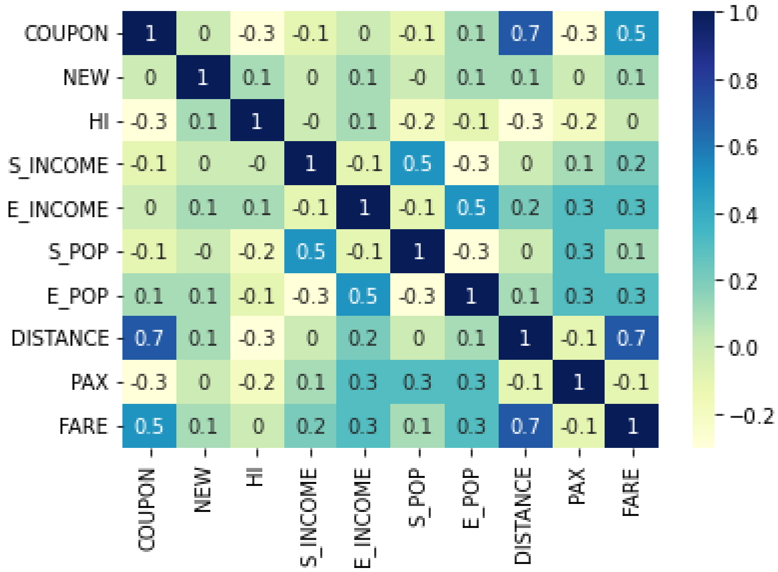
Out[73]:

	S_CODE	S_CITY	E_CODE	E_CITY	COUPON	NEW	HI	S_INCOME	E_IN
0	*	Dallas/Fort Worth TX	*	Amarillo TX	1.00	3	5291.99	28637.0	
1	*	Atlanta GA	*	Baltimore/Wash Intl MD	1.06	3	5419.16	26993.0	
2	*	Boston MA	*	Baltimore/Wash Intl MD	1.06	3	9185.28	30124.0	
3	ORD	Chicago IL	*	Baltimore/Wash Intl MD	1.06	3	2657.35	29260.0	
4	MDW	Chicago IL	*	Baltimore/Wash Intl MD	1.06	3	2657.35	29260.0	
...	...	...	...	...	...	...	...	...	
633	LGA	New York/Newark NY	*	West Palm Beach FL	1.08	3	2216.70	32991.0	
634	EWR	New York/Newark NY	*	West Palm Beach FL	1.08	0	2216.70	32991.0	
635	*	Philadelphia/Camden PA	*	West Palm Beach FL	1.17	3	6797.80	27994.0	
636	IAD	Washington DC	*	West Palm Beach FL	1.28	3	5566.43	31981.0	
637	DCA	Washington DC	*	West Palm Beach FL	1.28	3	5566.43	31981.0	

638 rows × 14 columns

Explore the numerical predictors and response (FARE) by creating a correlation table and examining a scatterplot between FARE and that predictor.

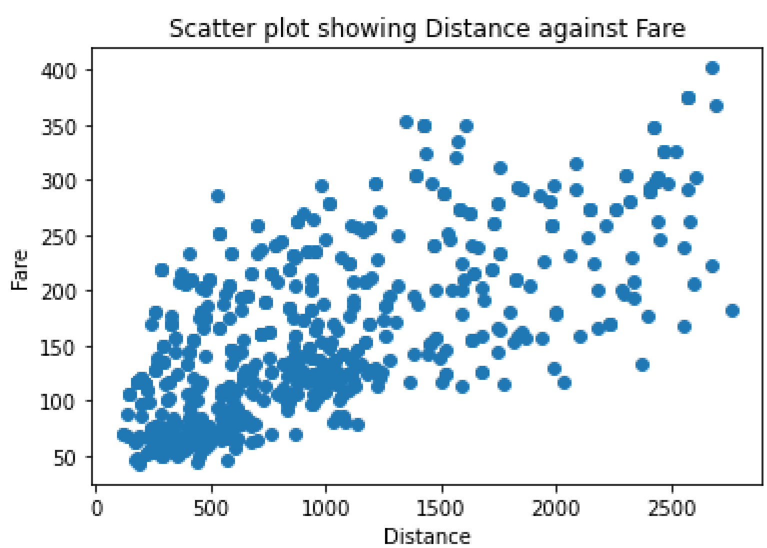
In [60]:
```
import seaborn as sns
dp = sns.heatmap(air2.corr().round(1), cmap="YlGnBu", annot=True)
```



What seems to be the best single predictor of FARE in the plot below? **hint the cell below may give you a clue to the answer - you must determine if there is a best predictor of FARE**

```
- The best single predictor of FARE from the plot below seems to be
DISTANCE. From the correlation table above DISTANCE and Fare have the
highest realtionship with a .7 and the closest one after that is FARE
and COUPON with a .5
```

In [61]:
```python
# Plot of distance against fare
plt.scatter(air2.DISTANCE, air2.FARE)
plt.title('Scatter plot showing Distance against Fare')
plt.xlabel('Distance')
plt.ylabel('Fare')
plt.show()
```



In your opinion, does the plot provide further evidence there appears to be a relationship between Fare and Distance?

```
- The plot does provide further evidence that there is a relationship
between FARE and DISTANCE. The plot points of FARE rise consistently
with DISTANCE with few points that I would consider outliers that
fall below the mean distribution line. I am guessing that these
flights were not very popular and the airlines reduced the cost to
fill the seats.
```

Explore the categorical predictors by looking at the flights in each category. These are the variables you removed above. Please list the categorical variables you will be looking at.

Which categorical predictor seems best for predicting FARE? Remember, the intuition here is you are looking for what categorical variable has the most significant impact on FARE. I recommend using PivotTables.

Syntax: pd.pivot_table(input,index,values,aggfunc)

In [62]:
```
air.pivot_table('FARE', index='VACATION', aggfunc=np.mean)
```

Out[62]:

	FARE
VACATION	
No	173.552500
Yes	125.980882

In [63]:
```
air.pivot_table('FARE', index='SW', aggfunc=np.mean)
```

Out[63]:

	FARE
SW	
No	188.182793
Yes	98.382268

In [64]:
```
air.pivot_table('FARE', index='SLOT', aggfunc=np.mean)
```

Out[64]:

	FARE
SLOT	
Controlled	186.059396
Free	150.825680

In [65]:
```
air.pivot_table('FARE', index='GATE', aggfunc=np.mean)
```

Out[65]:

GATE	FARE
Constrained	193.129032
Free	153.095953

```
- The categorical variable that appears to be the best for predicting
FARE is SW, which is whether or not South West Airlines serviced the
flight. As it had the highest difference in mean when it comes to
FARE with almost double the FARE cost for those that did not fly
South West.
```

Develop a model for explaining the average fare on a new route.

Below you will see the categorical variables are set up as dummy variables.

In [66]:
```
air['VACATION'] = [1 if v == 'Yes' else 0 for v in air['VACATION']]
air['SW'] = [1 if v == 'Yes' else 0 for v in air['SW']]
air['SLOT'] = [1 if v == 'Controlled' else 0 for v in air['SLOT']]
air['GATE'] = [1 if v == 'Constrained' else 0 for v in air['GATE']]
```

Partition the data into training and validation sets using all the predictors. The outcome variable will be Fare. The model will eventually be fit to the training data and evaluated on the validation set.

In [67]:
```
train_X, valid_X, train_y, valid_y = train_test_split(X,y, test_size=0.4, random_state
```

In [74]:
```
air_lm= LinearRegression()
air_lm.fit(train_X, train_y)

print('intercept ', air_lm.intercept_)
print(pd.DataFrame({'Predictor': X.columns, 'coefficient': air_lm.coef_}))

regressionSummary(train_y, air_lm.predict(train_X))
```

```
intercept  52.97284313801684
       Predictor  coefficient
0         COUPON   -11.744039
1            NEW    -2.508229
2             HI     0.006876
3       S_INCOME     0.000626
4       E_INCOME     0.001247
5          S_POP     0.000004
6          E_POP     0.000004
7       DISTANCE     0.077822
8            PAX    -0.000916
9   VACATION_Yes   -35.206099
10        SW_Yes   -42.157140
11     SLOT_Free   -13.533183
12     GATE_Free   -21.185849

Regression statistics

                      Mean Error (ME) : -0.0000
       Root Mean Squared Error (RMSE) : 34.7664
            Mean Absolute Error (MAE) : 27.1065
          Mean Percentage Error (MPE) : -4.5089
Mean Absolute Percentage Error (MAPE) : 20.0849
```

In [49]:

```
Training set: (382, 13) Validation set: (256, 13)
```

Identify the most prevalent predictors using an exhaustive search.

In [75]:

```
def train_model(variables):
    model = LinearRegression()
    model.fit(train_X[variables], train_y)
    return model
```

In [76]:

```
def score_model(model, variables):
    pred_y = model.predict(train_X[variables])
    return -adjusted_r2_score(train_y, pred_y, model)
```

In [77]:

```
allVariables = train_X.columns
results = exhaustive_search(allVariables, train_model, score_model)
```

In [81]:

```
data = []
for result in results:
    model = result['model']
    variables = result['variables']
    AIC = AIC_score(train_y, model.predict(train_X[variables]), model)
    d = {'n': result['n'], 'r2adj': -result['score'], 'AIC': AIC}
    d.update({var: var in result['variables'] for var in allVariables})
    data.append(d)
```

In [82]:

```
pd.set_option('display.width', 100)
print(pd.DataFrame(data, columns=('n', 'rdadj', 'AIC') + tuple(sorted(allVariables))))
```

```
    n  rdadj          AIC  COUPON  DISTANCE  E_INCOME  E_POP  GATE_Free     HI    NE
W    PAX  \
0    1    NaN  4149.881509   False      True     False  False      False  False  Fals
e  False
1    2    NaN  4025.892420   False      True     False  False      False  False  Fals
e  False
2    3    NaN  3913.585125   False      True     False  False      False  False  Fals
e  False
3    4    NaN  3890.268211   False      True     False  False      False   True  Fals
e  False
4    5    NaN  3873.328296   False      True     False  False       True   True  Fals
e  False
5    6    NaN  3852.808698   False      True     False  False       True   True  Fals
e  False
6    7    NaN  3843.170960   False      True     False   True      False   True  Fals
e   True
7    8    NaN  3833.945866   False      True     False   True       True   True  Fals
e   True
8    9    NaN  3826.433471   False      True     False   True       True   True  Fals
e   True
9   10    NaN  3821.876901   False      True      True   True       True   True  Fals
e   True
10  11    NaN  3822.798222   False      True      True   True       True   True  Fals
e   True
11  12    NaN  3823.825398   False      True      True   True       True   True   Tru
e   True
12  13    NaN  3825.237680    True      True      True   True       True   True   Tru
e   True

    SLOT_Free  SW_Yes  S_INCOME  S_POP  VACATION_Yes
0       False   False     False  False         False
1       False    True     False  False         False
2       False    True     False  False          True
3       False    True     False  False          True
4       False    True     False  False          True
5        True    True     False  False          True
6       False    True     False   True          True
7       False    True     False   True          True
8        True    True     False   True          True
9        True    True     False   True          True
10       True    True      True   True          True
11       True    True      True   True          True
12       True    True      True   True          True
```

Which variables are the most prevalent to include? **Hint** It is not all of them.

```
- DISTANCE, E_INCOME, E_POP, GATE, HI, PAX, SLOT, SW, S_POP, and
VACATION. I chose these ones because these variables were all true at
the point where the AIC was the lowest.
```

Predictive Regression Modeling

Now that you know which variables to include, implement a linear model on those predictors. Your output should be a regression summary, the coefficients, and output your model performance on the training and test

dataset. Discuss the performance of your model. Based on the errors, did the model fit well on the new test data?

In [84]:
```
pred = ['DISTANCE', 'E_INCOME', 'E_POP', 'GATE', 'HI', 'PAX', 'SLOT', 'SW', 'S_POP', '
out = 'FARE'
```

In [85]:
```
X = pd.get_dummies(air[pred], drop_first=True)
y = air[out]
```

In [87]:
```
train_X, valid_X, train_y, valid_y = train_test_split(X,y, test_size=0.4, random_state
```

In [88]:
```
air_lm= LinearRegression()
air_lm.fit(train_X, train_y)

print('intercept ', air_lm.intercept_)
print(pd.DataFrame({'Predictor': X.columns, 'coefficient': air_lm.coef_}))

regressionSummary(train_y, air_lm.predict(train_X))
```

```
intercept  17.564893917150414
  Predictor  coefficient
0  DISTANCE     0.075558
1  E_INCOME     0.001148
2     E_POP     0.000004
3      GATE    21.410803
4        HI     0.007188
5       PAX    -0.000829
6      SLOT    13.915304
7        SW   -43.031272
8     S_POP     0.000004
9  VACATION   -35.865596

Regression statistics

                      Mean Error (ME) : -0.0000
       Root Mean Squared Error (RMSE) : 34.8867
            Mean Absolute Error (MAE) : 27.1374
          Mean Percentage Error (MPE) : -4.5313
Mean Absolute Percentage Error (MAPE) : 20.1672
```

In [89]:
```
pred_y = air_lm.predict(train_X)

print('adjusted r2: ', adjusted_r2_score(train_y, pred_y, air_lm))
print('AIC: ', AIC_score(train_y, pred_y, air_lm))
```

```
adjusted r2:  0.7735665926875468
AIC:  3821.876901268484
```

```
- With an r2 of 77.36% I would say that this model is a good fit.
Also with the mean error of 0 would help me confirm that this model
is a good fit.
```

Using the developed model, predict the average fare on a route with the following characteristics: COUPON = 1.202, NEW = 3, VACATION = No, SW = No, HI = 4442.141, S_INCOME = 28,760, E_INCOME = 27,664, S_POP = 4,557,004, E_POP = 3,195,503, SLOT = Free, GATE = Free, PAX = 12,782, DISTANCE = 1976 miles.

In [90]:
```
# Fare when SW does not cover this route
# enter new data in data frame format
new_data = pd.DataFrame([
    { 'COUPON': 1.202, 'NEW': 3, 'VACATION': 0, 'SW': 0, 'HI': 4442.141, 'E_INCOME': 2
     'PAX': 12782, 'DISTANCE': 1976}])

print(new_data)
```

```
   COUPON  NEW  VACATION  SW        HI  E_INCOME    S_POP    E_POP  SLOT  GATE    PAX
DISTANCE
0   1.202    3         0   0  4442.141     27664  4557004  3195503     1     1  12782
1976
```

In [96]:
```
# predict Fare when SW does not cover this route
preds = air_lm.predict(new_data[pred])
print(preds)
```

```
[287.04763737]
```

Comment on how airlines incorporating this model could assist them with their business model.

```
- Airlines using this model could help them better manage there
prices to drive up sales while still makeing sure they are able to
turn a profit. If they are able to predict prices based on this
criteria they will have an advantage when it comes to marketing their
tickets to those that are searching for the best deals months in
advance of the flight date.
```

Submission Instructions

1. Print your Notebook file as a PDF by completing the following: a. File>Print Preview b. Right-click>Print>Destination>Print as PDF
2. Save the PDF file as YourLastNameAssignment05.
3. Submit the PDF file to the dropbox "Assignment 05".

Supplemental Resources

Use the following to help if you get stuck:

1. Google the error you may be receiving. At a higher level, google what you are expected to do. For example: How do I subtract variables in Python?
2. Navigate through the slides and use the find feature to locate keywords.
3. Rewatch a lecture video or two.
4. Check out YouTube. This is a great resource for students.
5. Post in the discussion board. Remember, I receive emails of postings and other students will as well. If you are not receiving emails, subscribe to the assignment questions discussion board.

```
In [ ]:
```