Problem 4:

(P8.43, textbook) Refer to University admissions data set in Appendix C.4. The director of admissions at a state university wished to determine how accurately students' grade-point averages at the end of their freshman year (Y) can be predicted from the entrance examination (ACT) test score ($X2$); the high school class rank ($X1$, a percentile where 99 indicates student is at or near the top of his or her class and 1 indicates student is at or near the bottom of the class); and the academic year ($X3$). The academic year variable covers the years 1996 through 2000. Develop a prediction model for the director of admissions. Justify your choice of model. Assess your model's ability to predict and discuss its use as a tool for admissions decisions.

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

In [5]:
```
df = pd.read_csv('APPENC04.txt', sep = '\s+', header =None, names=['Y','X1','X2','X3']
df.head()
```

Out[5]:

	Y	X1	X2	X3
1	0.98	61	20	1996
2	1.13	84	20	1996
3	1.25	74	19	1996
4	1.32	95	23	1996
5	1.48	77	28	1996

In [6]:
```
x1= df['X1']
x2= df['X2']
x3= df['X3']
y= df['Y']
```

Regression of Y on X1

In [41]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model1 = smf.ols('y ~ x1', data=df)
results1 = model1.fit()
sse1 = np.sum((results1.fittedvalues - df.Y)**2)
ssr1 = np.sum((results1.fittedvalues - df.Y.mean())**2)
sstoX1 = ssr1+sse1
R2_X1 = ssr1/sstoX1
print('R^2 =',R2_X1)
n=len(y)
p1=2
R2a_X1 = 1 - (sse1/(n-p1))/(sstoX1/(n-1))
print('R^2a =',R2a_X1)
```

```
R^2 = 0.15878535706175279
R^2a = 0.1575887501727936
```

Regression of Y on X2

In [42]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model2 = smf.ols('y ~ x2', data=df)
results2 = model2.fit()
sse2 = np.sum((results2.fittedvalues - df.Y)**2)
ssr2 = np.sum((results2.fittedvalues - df.Y.mean())**2)
sstoX2 = ssr2+sse2
R2_X2 = ssr2/sstoX2
print('R^2 =',R2_X2)
n=len(y)
p1=2
R2a_X2 = 1 - (sse2/(n-p1))/(sstoX2/(n-1))
print('R^2a =',R2a_X2)
```

```
R^2 = 0.1336711825899664
R^2a = 0.13243885141299627
```

Regression of Y on X3

In [43]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model3 = smf.ols('y ~ x3', data=df)
results3 = model3.fit()
sse3 = np.sum((results3.fittedvalues - df.Y)**2)
ssr3 = np.sum((results3.fittedvalues - df.Y.mean())**2)
sstoX3 = ssr3+sse3
R2_X3 = ssr3/sstoX3
print('R^2 =',R2_X3)
n=len(y)
p1=2
R2a_X3 = 1 - (sse3/(n-p1))/(sstoX3/(n-1))
print('R^2a =',R2a_X3)
```

```
R^2 = 0.0005780151264460155
R^2a = -0.0008436377681109164
```

Regression of Y on X1 and X2

In [44]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model12 = smf.ols('y ~ x1+x2', data=df)
results12 = model12.fit()
sse12 = np.sum((results12.fittedvalues - df.Y)**2)
ssr12 = np.sum((results12.fittedvalues - df.Y.mean())**2)
ssto12 = ssr12+sse12
R2_X1X2 = ssr12/ssto12
print('R^2 =',R2_X1X2)
n=len(y)
p2=3
R2a_X1X2 = 1 - (sse12/(n-p2))/(ssto12/(n-1))
print('R^2a =',R2a_X1X2)
```

```
R^2 = 0.2033362320191845
R^2a = 0.20106653467450986
```

Regression of Y on X1 and X3

In [45]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model13 = smf.ols('y ~ x1+x3', data=df)
results13 = model13.fit()
sse13 = np.sum((results13.fittedvalues - df.Y)**2)
ssr13 = np.sum((results13.fittedvalues - df.Y.mean())**2)
ssto13 = ssr13+sse13
R2_X1X3 = ssr13/ssto13
print('R^2 =',R2_X1X3)
n=len(y)
p2=3
R2a_X1X3 = 1 - (sse13/(n-p2))/(ssto13/(n-1))
print('R^2a =',R2a_X1X3)
```

```
R^2 = 0.1596954462049164
R^2a = 0.15730141613712412
```

Regression of Y on X2 and X3

In [46]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model23 = smf.ols('y ~ x2+x3', data=df)
results23 = model23.fit()
sse23 = np.sum((results23.fittedvalues - df.Y)**2)
ssr23 = np.sum((results23.fittedvalues - df.Y.mean())**2)
ssto23 = ssr23+sse23
R2_X2X3 = ssr23/ssto23
print('R^2 =',R2_X2X3)
n=len(y)
p2=3
R2a_X2X3 = 1 - (sse23/(n-p2))/(ssto23/(n-1))
print('R^2a =',R2a_X2X3)
```

```
R^2 = 0.13401058637869417
R^2a = 0.13154338007208088
```

Regression of Y on X1, X2 and X3

In [47]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model123 = smf.ols('y ~ x1+x2+x3', data=df)
results123 = model123.fit()
sse123 = np.sum((results123.fittedvalues - df.Y)**2)
ssr123 = np.sum((results123.fittedvalues - df.Y.mean())**2)
ssto123 = ssr123+sse123
R2_X1X2X3 = ssr123/ssto123
print('R^2 =',R2_X1X2X3)
n=len(y)
p3=4
R2a_X1X2X3 = 1 - (sse123/(n-p3))/(ssto123/(n-1))
print('R^2a =',R2a_X1X2X3)
```

```
R^2 = 0.20395902134699967
R^2a = 0.2005522839205247
```

- Model X1: R^2a= 0.1575
- Model X2: R^2a= 0.1324
- Model X3: R^2a= 0.0008
- Model X1&X2: R^2a= 0.201
- Model X1&X3: R^2a= 0.1573
- Model X2&X3: R^2a= 0.1315
- Model X1&X2&X3: R^2a= 0.2005

We can see that the adjusting R^2 (R^2a) of model X1&X2 is highest means that this model is best fit among those model above