Quiz 2
DataSci 347: Machine Learning 1

Instruction

This is an open book, 10-15 minute quiz. Answer all 9 questions and choose the correct answer. The first portion of the quiz uses a subset of 200 subjects that are randomly chosen from IQ.Full.csv. From this dataset we extracted their 4 AFQT tests: Arith, Word, Parag and Math. The dataset is named afqt.

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

data_full = pd.read_csv("IQ.Full.csv")
data1 = data_full[['Arith', 'Word', 'Parag', 'Math']]

np.random.seed(1)
n = len(data1)
afqt = data1.iloc[np.random.choice(n, 200, replace=False)]

print(afqt.columns.tolist())
```

```
## ['Arith', 'Word', 'Parag', 'Math']
```

```
afqt_mean = afqt.mean()
afqt_mean
```

```
## Arith    18.4
## Word     26.4
## Parag    11.1
## Math     14.2
```

```
afqt_sd = afqt.std(ddof=1)
afqt_sd
```

```
## Arith    7.07
## Word     7.37
## Parag    3.26
## Math     6.42
```

1. We first perform PCA to summarize the set of four tests. The four tests are first centered and scaled.

```
scaler = StandardScaler()
afqt_scaled = scaler.fit_transform(afqt)
afqt_pca = PCA()
afqt_pca.fit(afqt_scaled)
afqt_pca.components_.T
```

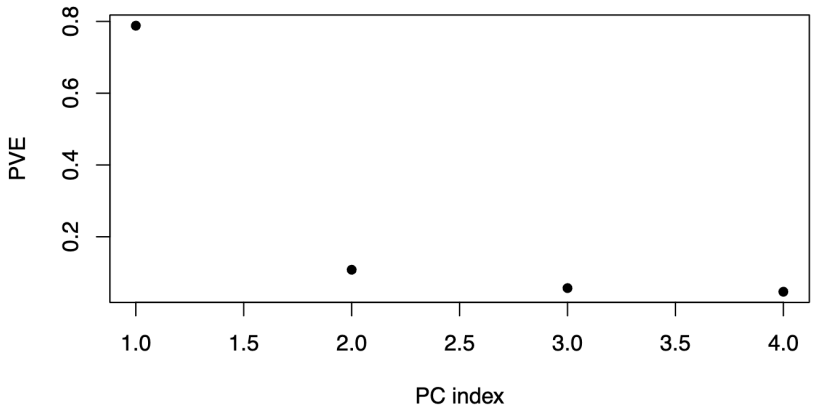
```
##         PC1    PC2      PC3     PC4
## Arith 0.502 -0.518  0.00136 -0.6928
## Word  0.503  0.394 -0.76584  0.0682
## Parag 0.489  0.603  0.62320 -0.0956
## Math  0.506 -0.461  0.15846  0.7115
```

PC1 scores are approximately equal to:
(A) .5 (Arith + Word + Parag + Math)
(B) .5 [(Arith - 18.41)/7.068) + (Word - 26.39)/7.374) + (Parag - 11.095)/3.256) + (Math - 14.21)/6.416)]

2. The PC1 score of afqt_pca in question 1 has the largest variance among all 4 PC scores.
(A) True
(B) False

3. Based on the following PVE plot we see that

```
pve = afqt_pca.explained_variance_ratio_
plt.plot(range(1, 5), pve, 'o', markersize=8)
plt.xlabel('PC index')
plt.ylabel('PVE')
plt.xticks(range(1, 5))
plt.show()
```



(A) PC1 accounts for approximately 80% of the total variance among the 4 PCs
(B) PC1 accounts for approximately 20% of the total variance among the 4 PCs

We next run a kmeans clustering analysis specifying 2 clusters.

```
afqt_kmeans = KMeans(n_clusters=2, random_state=0).fit(afqt)
unique, counts = np.unique(afqt_kmeans.labels_, return_counts=True)
counts
```

```
## [116  84]
```

4. Choose the correct answer:
(A) There are 100 subjects in cluster 1 and another 100 in cluster 2
(B) There are 116 in cluster 1 and 84 in cluster 2.

The remaining quiz questions are about regression. We will use a subset from the Cars_04 data that has been used in class. We will use MPG_Hwy as the response variable.

Let us first take a subset of the data and name it car.data.

```
np.random.seed(10)
car_temp = pd.read_csv("Cars_04.csv")
s_index = np.random.choice(len(car_temp), 200, replace=False)
car_data = car_temp.iloc[s_index]
car_data.describe()
```

```
##
## Acura_MDX                    :  1   Am:61
## Acura_NSX                    :  1   As:81
## Acura_RL                     :  1   E :58
## Acura_RSX                    :  1
## Acura_TSX                    :  1
## Aston_Martin_V12_Vanquish    :  1
## (Other)                      :194
##
```

```
##      MPG_City        Horsepower         Weight
##  Min.   :10.0    Min.   : 65      Min.   :1.98
##  1st Qu.:16.0    1st Qu.:160      1st Qu.:3.11
##  Median :19.0    Median :203      Median :3.54
##  Mean   :19.4    Mean   :226      Mean   :3.67
##  3rd Qu.:22.0    3rd Qu.:275      3rd Qu.:4.06
##
##     Seating          Length          MPG_Hwy         Origin
##  Min.   :2.00    Min.   :143      Min.   :14.0   Min.   :1.00
##  1st Qu.:5.00    1st Qu.:177      1st Qu.:22.0   1st Qu.:1.00
##  Median :5.00    Median :187      Median :26.0   Median :2.00
##  Mean   :4.93    Mean   :186      Mean   :25.9   Mean   :2.06
##  3rd Qu.:5.00    3rd Qu.:192      3rd Qu.:29.0   3rd Qu.:3.00
##  Max.   :8.00    Max.   :224      Max.   :60.0   Max.   :3.00
##
##  Transmission      EPA_Class          Width         Displacement
##  automatic:184   compact   :35   Min.   :65.4   Min.   :1.00
##  manual   :  6   midsize   :30   1st Qu.:69.5   1st Qu.:2.40
##                  suv2wd    :38   Median :71.7   Median :3.20
##                  two_seater:22   Mean   :72.1   Mean   :3.31
##                  suv4wd    :19   3rd Qu.:74.7   3rd Qu.:4.20
##                  large     :18   Max.   :80.5   Max.   :8.30
##                  (Other)   :38   NA's   :51
##
##  Cylinders          Make            Model          Turndiam
##  Min.   : 2.00   Chevrolet : 12   300M          :  1   Min.   :30.2
##  1st Qu.: 4.00   Toyota    : 12   3             :  1   1st Qu.:35.4
##  Median : 6.00   Volkswagen:  9   360_Modena    :  1   Median :37.1
##  Mean   : 5.88   Honda     :  8   4RunnerSR5    :  1   Mean   :37.2
##  3rd Qu.: 6.00   Mitsubishi:  8   525i          :  1   3rd Qu.:38.7
##  Max.   :12.00   Cadillac  :  7   575M_Maranello:  1   Max.   :43.5
##                  (Other)   :144   (Other)       :194
```

We then fit a linear model fit1: MPG_Hwy vs. Horsepower

```python
from sklearn.linear_model import LinearRegression
import scipy.stats as stats

X1 = car_data[['Horsepower']]
y = car_data['MPG_Hwy']

fit1 = LinearRegression()
fit1.fit(X1, y)

# Calculate statistics
y_pred = fit1.predict(X1)
residuals = y - y_pred
n = len(y)
p = 1  # number of predictors
residual_std_error = np.sqrt(np.sum(residuals**2) / (n - p - 1))
```

```

# R-squared
ss_res = np.sum(residuals**2)
ss_tot = np.sum((y - np.mean(y))**2)
r_squared = 1 - (ss_res / ss_tot)
adj_r_squared = 1 - (1 - r_squared) * (n - 1) / (n - p - 1)

# F-statistic
f_stat = (r_squared / p) / ((1 - r_squared) / (n - p - 1))
f_pvalue = 1 - stats.f.cdf(f_stat, p, n - p - 1)

print(f"Intercept: {fit1.intercept_:.5f}")
print(f"Horsepower coefficient: {fit1.coef_[0]:.5f}")
```

```
##
## Call:
## LinearRegression()
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.324  -2.785   0.042   2.322  23.024
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.82117    0.81791    43.8   <2e-16 ***
## Horsepower  -0.04377    0.00334   -13.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.44 on 198 degrees of freedom
## Multiple R-squared:  0.465,  Adjusted R-squared:  0.462
## F-statistic:  172 on 1 and 198 DF,  p-value: <2e-16
```

5. Based on summary of fit1, choose correct answer(s).

(A) On average MPG_Hwy decreases 0.044 when Horsepower increases by 1.

(B) Take two cars, car1 with Horsepower=220 and car2 with Horsepower=221; fit1 tells us MPG_Hwy is guaranteed to be higher in car1 than car2.

Next, we add one variable Weight to fit1 and store the result in fit2.

```
X2 = car_data[['Horsepower', 'Weight']]
y = car_data['MPG_Hwy']

fit2 = LinearRegression()
fit2.fit(X2, y)

# Calculate statistics for fit2
y_pred2 = fit2.predict(X2)
residuals2 = y - y_pred2
n = len(y)
p = 2  # number of predictors
```

```python
residual_std_error2 = np.sqrt(np.sum(residuals2**2) / (n - p - 1))

# R-squared
ss_res2 = np.sum(residuals2**2)
ss_tot2 = np.sum((y - np.mean(y))**2)
r_squared2 = 1 - (ss_res2 / ss_tot2)
adj_r_squared2 = 1 - (1 - r_squared2) * (n - 1) / (n - p - 1)

# F-statistic
f_stat2 = (r_squared2 / p) / ((1 - r_squared2) / (n - p - 1))
f_pvalue2 = 1 - stats.f.cdf(f_stat2, p, n - p - 1)

print(f"Intercept: {fit2.intercept_:.5f}")
print(f"Horsepower coefficient: {fit2.coef_[0]:.5f}")
print(f"Weight coefficient: {fit2.coef_[1]:.5f}")
```

```
##
## Call:
## LinearRegression()
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.871 -1.815 -0.316  1.738 18.753
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.98899    1.20757   38.91   <2e-16 ***
## Horsepower  -0.02803    0.00301   -9.33   <2e-16 ***
## Weight      -4.01044    0.36626  -10.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.51 on 197 degrees of freedom
## Multiple R-squared:  0.667,  Adjusted R-squared:  0.664
## F-statistic: 197 on 2 and 197 DF, p-value: <2e-16
```

6. From fit2, we see that 1 unit increase in horsepower always results in a decrease in MPG_Hwy on average by 0.028.

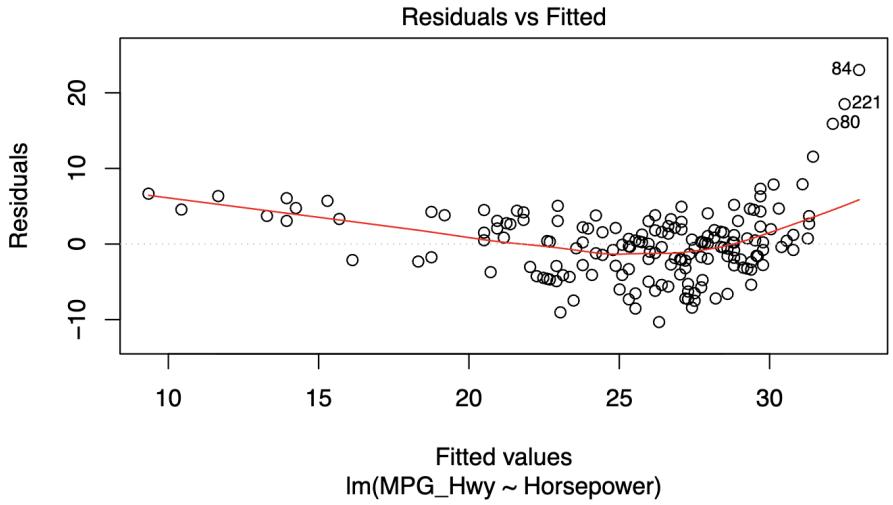
(A) True
(B) False

7. Based on fit2, we would like to estimate the mean of MPG_Hwy for all cars with the following measurements: Horsepower = 240, Weight = 3.5, with 4 seats and 180" long.

(A) We can not do it since Seats and Length are not included in the fit2
(B) It is $46.989 - 0.028 \times 240 - 4.01 \times 3.5$

Model diagnoses for fit2. Choose the correct answers.

```
# Residuals vs Fitted plot
plt.figure(figsize=(8, 6))
plt.scatter(y_pred, residuals, alpha=0.5)
plt.axhline(y=0, color='r', linestyle='--', linewidth=2)
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.title('Residuals vs
Fitted\nLinearRegression(MPG_Hwy ~ Horsepower)')
plt.grid(True, alpha=0.3)
plt.show()
```



8. Choose one answer.

(A) The linearity might be a problem since cars with smaller MPG_Hwy seem to be underestimated.

(B) The linearity might be a problem since cars with smaller MPG_Hwy seem to be overestimated.

9. fit2 can be used to reject $H_0 : \beta_1 = \beta_2 = 0$ at a significance level of 0.001 for the following reason:

(A) Because of a large R^2.

(B) Because the F test in the summary report has a p-value much smaller than .001.