Assignment8

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
```

How Much is Your Car Worth?

Data about the retail price of 2005 General Motors cars can be found in `car_data.csv`.

The columns are:

1. Price: suggested retail price of the used 2005 GM car in excellent condition.
2. Mileage: number of miles the car has been driven
3. Make: manufacturer of the car such as Saturn, Pontiac, and Chevrolet
4. Model: specific models for each car manufacturer such as Ion, Vibe, Cavalier
5. Trim (of car): specific type of car model such as SE Sedan 4D, Quad Coupe 2D
6. Type: body type such as sedan, coupe, etc.
7. Cylinder: number of cylinders in the engine
8. Liter: a more specific measure of engine size
9. Doors: number of doors
10. Cruise: indicator variable representing whether the car has cruise control (1 = cruise)
11. Sound: indicator variable representing whether the car has upgraded speakers (1 = upgraded)
12. Leather: indicator variable representing whether the car has leather seats (1 = leather)

Tasks, Part 1

1. Find the linear regression equation for mileage vs price.
2. Chart the original data and the equation on the chart.
3. Find the equation's R^2 score (use the `.score` method) to determine whether the equation is a good fit for this data. (0.8 and greater is considered a strong correlation.)

Tasks, Part 2

1. Use mileage, cylinders, liters, doors, cruise, sound, and leather to find the linear regression equation.

2. Find the equation's R^2 score (use the `.score` method) to determine whether the equation is a good fit for this data. (0.8 and greater is considered a strong correlation.)
3. Find the combination of the factors that is the best predictor for price.

Tasks, Hard Mode

1. Research dummy variables in scikit-learn to see how to use the make, model, and body type.
2. Find the best combination of factors to predict price.

```
df = pd.read_csv("car_data.csv")
```

```
data.head()
```

```
x = data[['Mileage']]
y = data[['Price']]
```

```
linreg = linear_model.LinearRegression()
linreg.fit(x, y)

print('intercept:', linreg.intercept_)
print('coefficients:', linreg.coef_)
print('r-squared:', linreg.score(x, y))
```

```
plt.scatter(x, y, color='r')
plt.plot(x, linreg.predict(x))
plt.xlabel('Mileage')
plt.ylabel('Price')
plt.title('Mileage vs Price')
```

```
degrees = 5
fig, axs = plt.subplots(degrees, figsize = (10, 30))

for degree in range(degrees):
    model = Pipeline([
        ('poly', PolynomialFeatures(degree=degree)),
        ('linear', linear_model.LinearRegression(fit_intercept=False))])

    points = 50000


    model = model.fit(x, y)
    model_score = model.score(x,y)

    subplot = axs[degree]
    subplot.plot(model.predict([[j] for j in range(points)]), color='r')
    subplot.scatter(x, y)
```

```
    subplot.set_title('{} degrees {} R-squeard'.format(degree, model_score))

plt.show()


feature = ['Mileage', 'Cylinder', 'Liter', 'Doors', 'Cruise', 'Sound', 'Leather']
x = data[feature]
y = data[['Price']]


linreg = linear_model.LinearRegression()
linreg.fit(x, y)
print('intercept:', linreg.intercept_)
print('coefficients:', linreg.coef_)


print('r-squared: ', linreg.score(x, y))


combinations = []
for x in range(1,8):
    combinations.append(itertools.combinations(feature, x))


best_score = 0
for item in combinations:
    for group in item:
        group = list(group)
        x = data[feature]
        linreg = linear_model.LinearRegression()
        linreg.fit(x, price)
        if best_score <
```