MACHINE LEARNING

MINI PROJECT ONE

PARTICIPANTS	CODE SECTION	REPORT SECTION	DOCUMENTATION SECTION	PRESENTATION SECTION
NATHAN FELIZZOLA	33.3	33.3	33.3	33.3
PHELLIPPE SOUZA-HEROD	33.3	33.3	33.3	33.3
CONNOR CODDINGTON	33.3	33.3	33.3	33.3

STEP ONE

Getting the Data

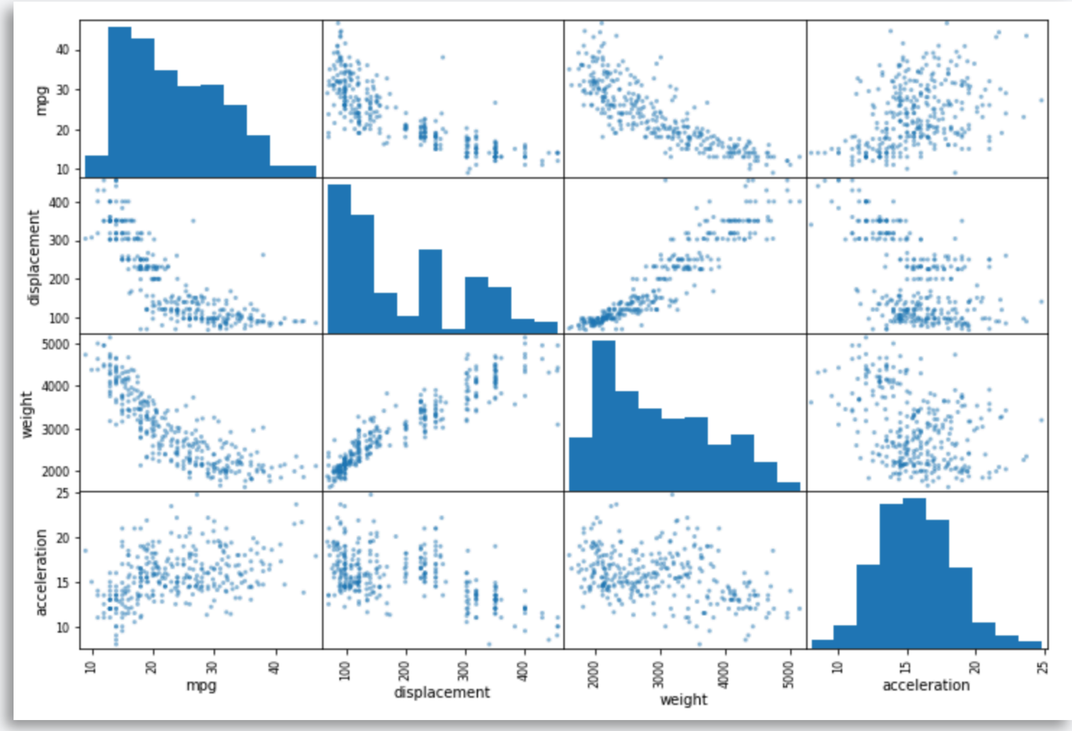
- Downloaded the Auto MPG Data Set from UCI ML Repository
- Used read_csv to import data into the notebook

```
headers = ['mpg','cylinders','displacement','horsepower','weight','acceleration','model year
','origin','car name']

auto_data = pd.read_csv('Data-CSVs/auto-mpg.data', delimiter='\s+', names=headers)
```

STEP TWO

Discover and visualize the Data to Gain Insight

- Scatter matrix of MPG, displacement, weight, and acceleration was created
- Noticed a possible predictor between the weight and MPG
- Function is scatter_matrix from pandas



STEP THREE

Prepare the data for Machine Learning algorithms

- No missing values to fix
- Decided to used MPG and Weight for the model
  - Found that MPG and weight had a high negative correlation
- Created a training set of 80% with 20% set aside for testing
  - Used train_test_split

```
from sklearn.model_selection import train_test_split

train_set, test_set = train_test_split(auto_data, test_size=0.2, random_state=42)
#splits the training and testing set for us
#function has two returns so we need to give two variables at the beginning
```

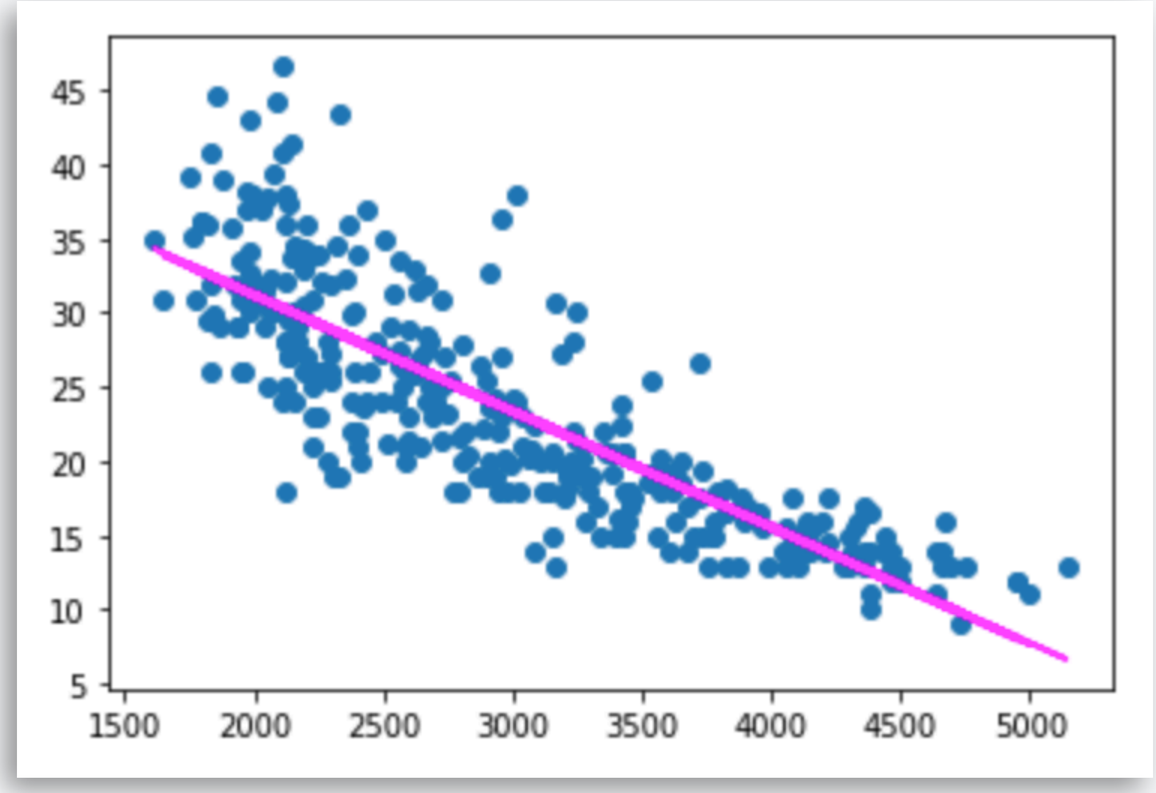
Selecting and Training the Model

- Model to be used is Linear Regression
- We needed to add a unique value to each weight to properly process the linear regression
- A fit was done using the training set of MPG and weight
- Predictions of MPG were gathered from the weight

```
for e, i in enumerate(train_set['weight'],start=1):
    train_set_weight.append([e,i])

train_set_mpg = train_set['mpg'].values #grab the mpg to be our output variable


#print(train_set_weight)

model = LinearRegression()

model.fit(train_set_weight, train_set_mpg)
#create a linear regression and train it with weight x mpg
```

Final Accuracy

- Our regression line was plotted
- Root mean squared was calculated and is 3.8
- Highest accuracy of 92% with cross validation



Questions?

THANK YOU FOR YOUR TIME