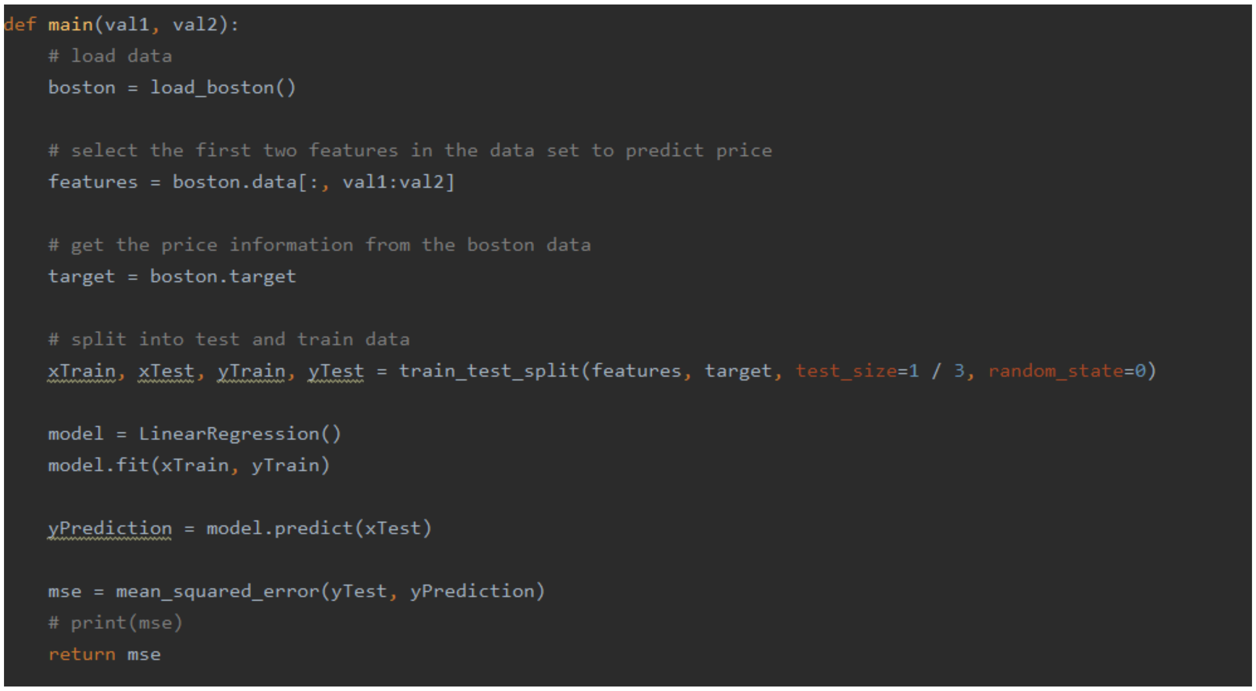
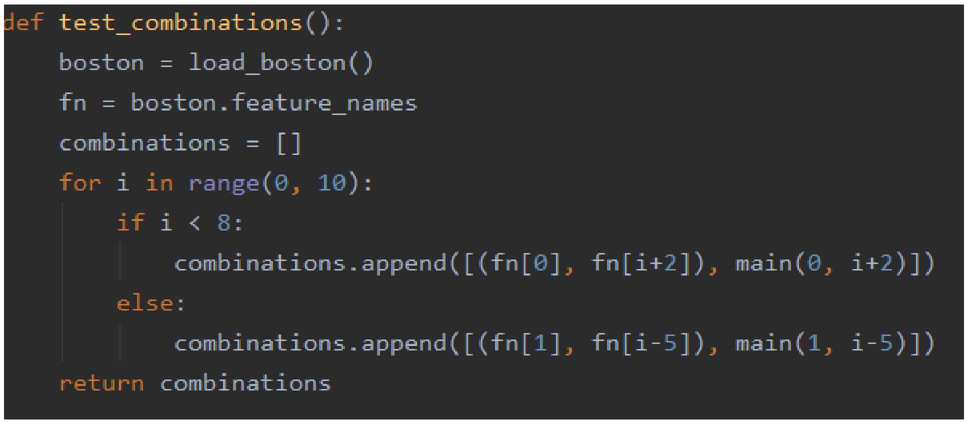
Andrew Breslauer

Boston Linear Regression Homework



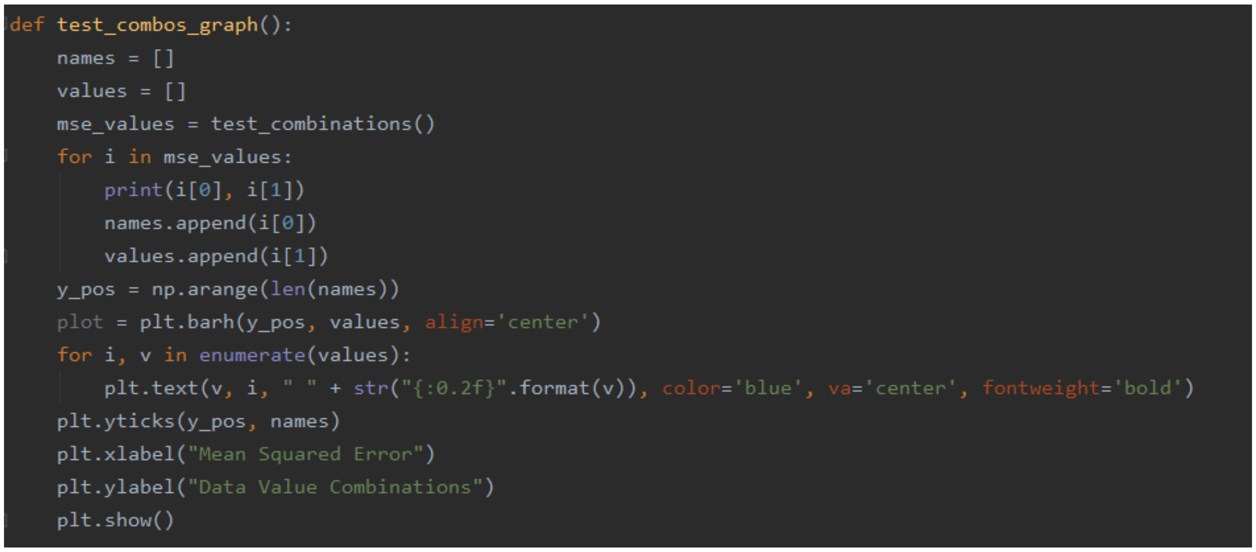
```
def main(val1, val2):
    # load data
    boston = load_boston()

    # select the first two features in the data set to predict price
    features = boston.data[:, val1:val2]

    # get the price information from the boston data
    target = boston.target

    # split into test and train data
    xTrain, xTest, yTrain, yTest = train_test_split(features, target, test_size=1 / 3, random_state=0)

    model = LinearRegression()
    model.fit(xTrain, yTrain)

    yPrediction = model.predict(xTest)

    mse = mean_squared_error(yTest, yPrediction)
    # print(mse)
    return mse
```

This function calculates the mse of a range of x-data values, and then returns that value, by splitting the dataset into test and training data.



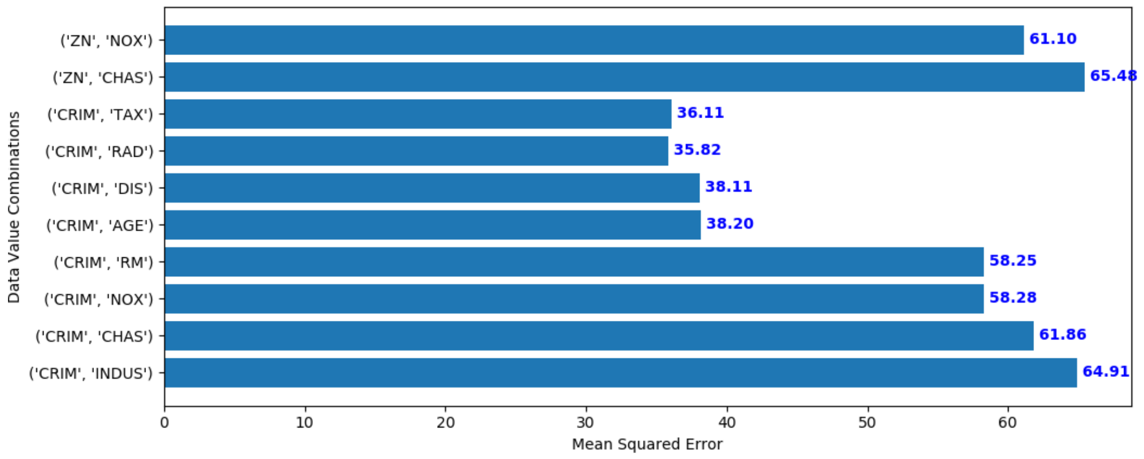
```
def test_combinations():
    boston = load_boston()
    fn = boston.feature_names
    combinations = []
    for i in range(0, 10):
        if i < 8:
            combinations.append([(fn[0], fn[i+2]), main(0, i+2)])
        else:
            combinations.append([(fn[1], fn[i-5]), main(1, i-5)])
    return combinations
```

This function will generate ten different combinations of x-data values for the main() to run on, returning a set of the combinations and the mse for each combination.



```python
def test_combos_graph():
    names = []
    values = []
    mse_values = test_combinations()
    for i in mse_values:
        print(i[0], i[1])
        names.append(i[0])
        values.append(i[1])
    y_pos = np.arange(len(names))
    plot = plt.barh(y_pos, values, align='center')
    for i, v in enumerate(values):
        plt.text(v, i, " " + str("{:0.2f}".format(v)), color='blue', va='center', fontweight='bold')
    plt.yticks(y_pos, names)
    plt.xlabel("Mean Squared Error")
    plt.ylabel("Data Value Combinations")
    plt.show()
```

This function will split the returned data-value/mse combinations and then create a horizontal bar chart visualizing the generated values.



These are the mean squared errors for 10 different data-value combinations, where the best performing combination is ‘CRIM’ (per capita Crime rate by town) and ‘RAD’ (index of accessibility to radial highways) at 35.82. There may or may not be a better combination among the other untested combinations, but ‘CRIM’ and ‘RAD’ generate the best linear regression line of the values tested.