Question 1: Simple linear regression

1. Plot the data using matplotlib. Do midterm and final exam seem to have a linear relationship? Discuss the data and their relationship in your report.

(1) Set the image title to 'Mid-term exam marks and final exam marks'

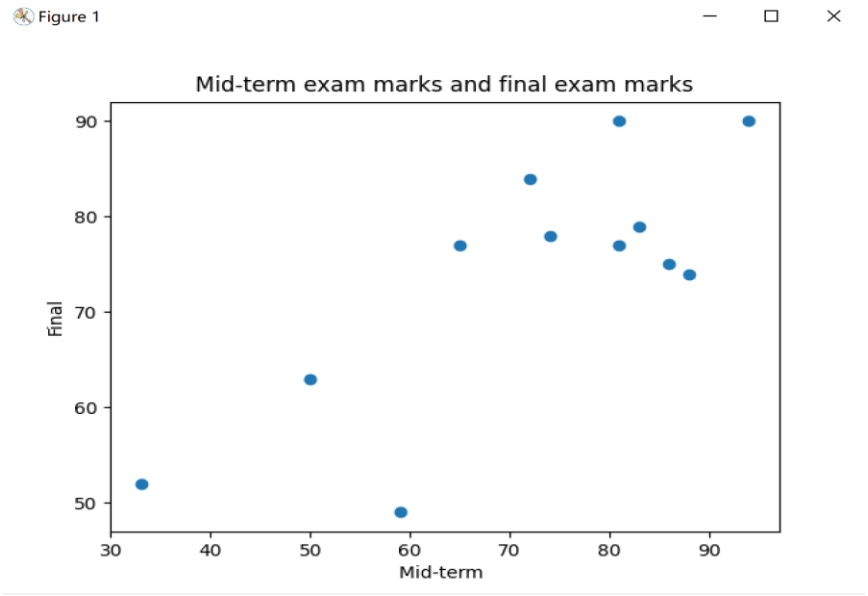
X axis title is set to Mid-term

Y axis title is set to Final

(2) sort values by midterm.

(3) save Picture

(4) show a scatter plot of the final grades versus mid-term grades



From the picture, it is very likely that there is a linear correlation between the mid-term marks and the final marks. Because the scattered points are roughly arranged up and down in a straight line.

By Calculating its Pearson correlation coefficient r and p.

```
r:0.7828090360394566
p:0.002608697385666579
```

The correlation coefficient is close to 0.8, and p is less than 0.05. Therefore, there is a greater certainty that it is linearly correlated.

2. Use linear regression to generate a model for the prediction of a students' final exam grade based on the students' midterm grade in the course, then describe the model in your report.

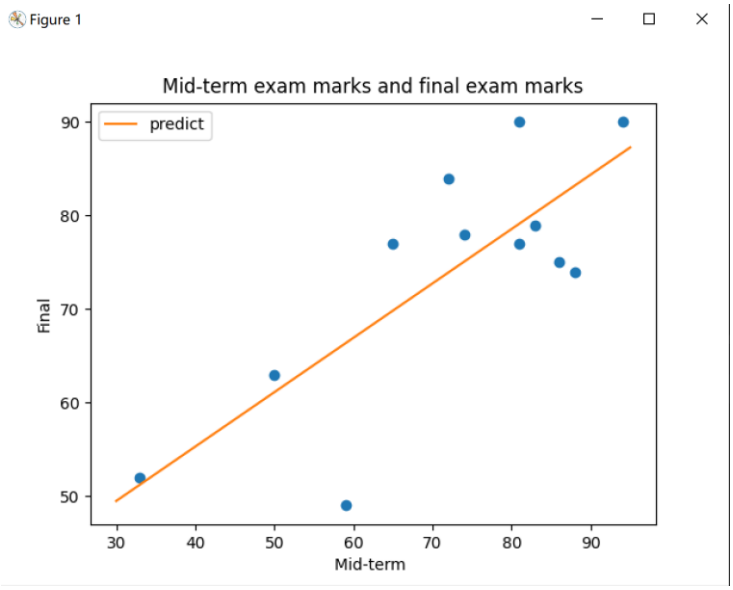
(1) Convert one-dimensional DataFrame to two-dimensional array. For example, midterm [72,50,81,74,…] → [[72],[50],[81],[74]…].

(2) Train the model using the midterm marks and final marks.

(3) Show intercept and coefficients of the unary linear regression model.

```
The intercept of the unary linear regression model is [32.02786108]
The coefficients of the unary linear regression model is [[0.58160008]]
```

(4) Draw the picture



3. According to your model, what will be the final exam grade of a student who received an 86 on the midterm exam?

(1) Predict: a = model.predict([[86]])

(2) Show the result:

```
The predicted final mark of a student who received an 86 on the midterm exam is 82.05

Process finished with exit code 0
```

The final exam grade of a student who received an 86 on the midterm exam is supposed to 82.05. (Round to the nearest hundredth)

Question 2: Classification with Decision Tree

1. Filter out the TID attribute, as it is not useful for decision making.

(1) read CSV file.

(2) delete TID attribute.

```
borrower_raw = pd.read_csv('./specs/borrower_question2.csv')
del borrower_raw['TID']
```

2. Using sklearn decision trees, generate a decision tree using information gain as splitting criterion, and a minimum impurity decrease of 0.5. Leave everything else to its default value. Plot the resulting decision tree, and discuss the classification results in your report.

(1) Split 'DefaultedBorrower' column as target (class label).

(2) Convert the dataset to fit the model (pd.get_dummies)

(3) Train the model

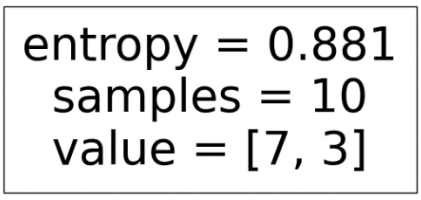
```
classifier_05=DecisionTreeClassifier(criterion='entropy', min_impurity_decrease=0.5)
tree_05 = classifier_05.fit(train_raw,target_raw)
```

train_raw:

AnnualIncome	HomeOwner_No	HomeOwner_Yes	MaritalStatus_Divorced	MaritalStatus_Married	MaritalStatus_Single
125	0	1	0	0	1
100	1	0	0	1	0
70	1	0	0	0	1
120	0	1	0	1	0
120	1	0	1	0	0
60	1	0	0	1	0
220	0	1	1	0	0
85	1	0	0	0	1
75	1	0	0	1	0
90	1	0	0	0	1

(4) plot decision tree

plot_tree in sklearn.tree



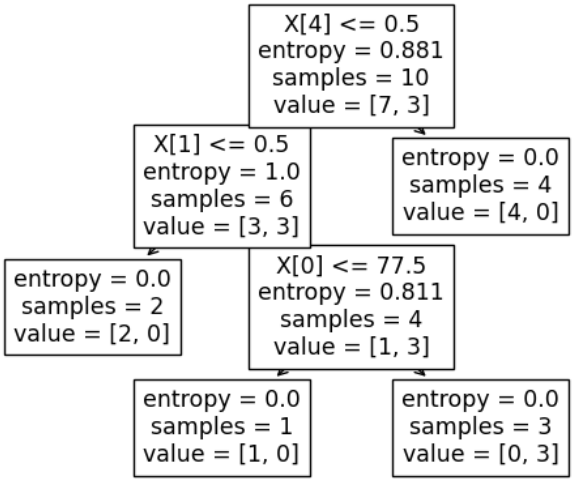
This decision tree has only the root node for the information gain is less than 0.5 no matter what kind of attribute is used.

3. Train another tree, but this time use a minimum impurity decrease of 0.1. Plot the resulting decision tree, and compare the results with the previous model you trained. Save the produced tree into ./output/tree_low.png.

(1) Train the model.

```
classifier_01 =DecisionTreeClassifier(criterion='entropy',min_impurity_decrease=0.1)
tree_01 = classifier_01.fit(train_raw,target_raw)
```

(2) plot decision tree



4. Discuss the generated models in your report.

AnnualIncome	HomeOwner_No	HomeOwner_Yes	MaritalStatus_Divorced	MaritalStatus_Married	MaritalStatus_Single
125	0	1	0	0	1
100	1	0	0	1	0
70	1	0	0	0	1
120	0	1	0	1	0
120	1	0	1	0	0
60	1	0	0	1	0
220	0	1	1	0	0
85	1	0	0	0	1
75	1	0	0	1	0
90	1	0	0	0	1

X[4] : MaritalStatus_Married

X[1] : HomeOwner_NO

X[0] : AnnualIncome

(1) If MaritalStatus_Married<0.5(Divorced or Single), then to (2).

Else DefaultedBorrower: NO

(If sample is Divorced or Single, then to (2), else If a sample gets married, then DefaultedBorrower: NO)

(2) If HomeOwner_NO<0.5(is a home owner) then DefaultedBorrower: NO.

Else to (3)

(If a sample is Divorced or Single, and she/he do not own a home, then to (3) else if sample is Divorced or Single, and she/he own a home, then DefaultedBorrower: NO.)

(3) If AnnualIncome <77.5 then DefaultedBorrower: NO. Else DefaultedBorrower: YES

(If a sample is Divorced or Single, and she/he do not own a home and her or his annual income >77.5 then DefaultedBorrower: YES)