IAML (Level 10) Assignment 1

Huacheng Song

TOTAL POINTS

70.5 / 80

QUESTION 1

Question 1 : Linear Regression 22 pts

1.1 Data properties 3 / 3

✓ **- 0 pts Correct**

- 1 pts You failed to mention the correct range of the revision time attributes (2.72 to 48.01)

- 1 pts You failed to mention the correct range of the exam score targets (14.73 to 94.94)

- 1 pts You failed to mention the size of the dataset (50 datapoints)

- 1 pts You failed to mention that the attributes are continuous

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 0.5 pts Your reported values for the data ranges are incorrect

- 1 pts The range has been specified correctly, but the min and max values have been omitted.

1.2 Linear Model 3 / 3

✓ **- 0 pts Correct**

- 2 pts Your reported model parameters are incorrect. They should be 17.898, and 1.441

- 1 pts You did not explain that the model parameters represented the intercept and slope

- 0.5 pts You included too many decimal places in your answer. Four or less is more appropriate

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 0.5 pts Your explanation of the model parameters goes towards the right direction, but is not quite there: you could simply say the model parameters represent the intercept and slope

1.3 Display 3 / 3

✓ **- 0 pts Correct**

- 1 pts Your linear model is a poor fit to the data

- 1 pts The fitted line is not continuous

- 1 pts You did not label the axis

- 0.5 pts You did not add a legend

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 1 pts You should not represent your input data using a line

1.4 Custom implementation 3 / 3

✓ **- 0 pts Correct**

- 3 pts Your code is obviously wrong e.g. you did not perform the pseudo inversion

- 2 pts You did not implement the solution using basic matrix operations e.g. you used np.linalg.lstsq

- 1 pts Your is overly long and does not make use of numpy expressions e.g. not using np.dot or np.linalg.inv

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 2 pts You did not implement the solution using basic matrix operations e.g. you used np.linalg.solve

- 2 pts You implemented linear regression only for 1D input attributes.

- 3 pts You fitted a polynomial.

1.5 MSE 3 / 3

✓ **- 0 pts Correct**

- **2 pts** Missing square term
- **1 pts** y-hat and/or y should have a consistent lowerscript
- **2 pts** You added a square root term
- **1 pts** Missing a limitation e.g. distorted by outliers or not in the same units as the data
- **1 pts** You did not use the suggested notation for the ground truth and model prediction i.e. $$y$$ and $$\hat{y}$$
- **3 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized
- **2 pts** Missing $$\frac{1}{N}$$ term

1.6 MSE 2 2.5 / 3

- **0 pts** Correct

✓ **- 0.5 pts** **You could comment on why MSEs are the same/different**

- **0.5 pts** You included too many decimal places in your answer. Four or less is more appropriate
- **1 pts** You MSE values is wrong. It should be 30.985
- **1 pts** For having different MSE scores for both approaches. They should be very similar for the most signi□ant decimal places
- **3 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized

1.7 Analysis 2 / 4

- **0 pts** Correct
- **2 pts** You failed to include the plot.
- **1 pts** The range of MSE is incorrect (should be around 8000 for $$w_1 = -2.0$$)
- **2 pts** The plot does not match the correct one in some aspect of the data
- **1 pts** The plot is unclear or insufficiently labelled i.e. you did not label the axis.
- **1 pts** You failed to identify the minimum value of MSE being close to 32.48 or the corresponding $$w_1$$ being close to 1.35

✓ **- 1 pts** **You failed to mention that the plot was convex/that there is a single minimum rather than multiple minima.**

- **0.5 pts** You have identified the plot as concave rather than convex.

✓ **- 1 pts** **You failed to mention that the resulting minimum $$w_1$$ value was to be expected to be similar to one we found earlier (i.e. close to 1.4)/otherwise refer to the previously calculated result of $$w_1$$ in your argument.**

- **0.5 pts** You included too many decimal places in your answer. Four or fewer is more appropriate.
- **1 pts** Answer too long/answer box resized
- **4 pts** You did not answer the question

You are justifying the global minimum with monotonic increase after 1.35. This argument is incomplete, as it should have been preceded with noting that the function is decreasing until 1.35 - alternatively you could have said that the function is convex.

QUESTION 2

Question 2 : Nonlinear Regression 18 pts

2.1 Polynomial regression 5 / 5

✓ **- 0 pts** **Correct**

- **2 pts** Your model predictions look very dfferent from the expected answer
- **1 pts** The plots for M = 3 and M = 4 are very different. In this example, they should look almost identical.
- **2 pts** You did not plot a continuous line for the models
- **1 pts** You did not plot the input data
- **1 pts** Your plot is unclear e.g. missing/wrong axes or a legend
- **5 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized
- **0.5 pts** The input data should be plotted as points as there is no connection among them
- **3 pts** You did not plot some of the lines for the models

2.2 Bar plot 3 / 3

✓ **- 0 pts Correct**

- 2 pts Your bar plot looks very different from the expected answer e.g. M = 3 and M = 4 should have very similar values for MSE, but both are less than M = 1 or M = 2

- 1 pts Your estimated MSE values are incorrect e.g. M = 1 should be around 24.7

- 1 pts Your plot is not clear e.g. missing axes

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

2.3 Analysis 4 / 4

✓ **- 0 pts Correct**

- 1 pts You failed to mention that M = 3 and M = 4 give very similar predictions.

- 2 pts You said that the M = 4 model is better. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

- 1 pts You did not mention that the weight vector entry corresponding to x 4 i is very small.

- 4 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 1 pts you did not explain correctly why M = 3 model is better. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

- 2 pts you did not answer which model to choose. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

- 1 pts you did not explain correctly why M=3 is better. You don't have sufficient evidence to use overfitting as an argument. To comment on overfitting a model should be evaluated on heldout/unseen data. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

Here you should not mention of overfitting. To comment on overfitting a model should be evaluated on held-out/unseen data. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

2.4 RBF 4 / 6

- 0 pts Correct

- 3 pts Your plot looks very different from the expected answer

- 2 pts You only plotted your model predictions where the input data was. You should have used more input points to better visualize the predictions i.e. to make the plot more continuous

✓ **- 1 pts You failed to mention what would happen if the width parameter is too large i.e. more datapoints further away from the basis center are included resulting in underfitttting (or overly smooth predictions)**

✓ **- 1 pts You failed to mention what would happen if the width parameter is too small i.e. model predictions will be a constant for points that are not close to the kernel center**

- 1 pts Your plot is not very clear e.g. you did not plot the input data, it is missing a legend, axis labels or the plot is not complete

- 6 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 3 pts You failed to provide the figure

- 2 pts Some of the plotted graphs are different from the expected answer

QUESTION 3

Question 3 : Decision Trees 26 pts

3.1 Dataset analysis 4 / 4

✓ **- 0 pts Correct**

- 1 pts You reported the wrong number of attributes. It should have been 136

- 0.5 pts You did not report the correct train set size (4800)

- 0.5 pts You did not report the correct test set size (1200)

- 0.5 pts You did not report the correct number of positive labels in the train set (2335)

- 0.5 pts You did not report the correct number of negative labels in the train set (2465)

- 0.5 pts You did not report the correct number of positive labels in the test set (592)

- 0.5 pts You did not report the correct number of negative labels in the test set (608)

- 4 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 1 pts You only reported the ratio of smiling faces to total faces, not the total number of smiling and not smiling faces.

- 1 pts You only reported the ratio of smiling faces to not smiling faces, not the total number of smiling and not smiling faces.

3.2 Analysis 4 / 4

✓ **- 0 pts Correct**

- 2 pts You failed to mention any difference between the two sets of points e.g. the mouth is wider and the chin is lower on the smiling face

- 1 pts You did not mention the difference clearly

- 3 pts The plot does not contain two faces

- 1 pts You did not plot both faces on the same figure. This makes it difficult to see the differences

- 1 pts You did not include a legend or way to indicate which points correspond to the train set and which correspond to the test

- 1 pts You plot is missing axes labels, you plotted lines instead of points, etc.

- 4 pts You did not answer the question

- 1 pts Answer too long/answer box resized

3.3 Decision Trees 2 / 2

✓ **- 0 pts Correct**

- 1 pts You did not specify the correct measure used by sklearn by default i.e. gini

- 1 pts You did not specify an advantage of gini over entropy e.g. computing entropy
requires more computation as you need to take logarithms

- 2 pts You did not answer the question

- 1 pts Answer too long/answer box resized

3.4 DT Depth 2 / 3

- 0 pts Correct

- 1 pts You did not mention what happens when you use a maxiumum depth that is
too small e.g. underfitting

- 2 pts You did not give two examples of what happens when you use a maxiumum
depth that is too large e.g. overfitting, large trees requiring larger storage,
and can be slow to evaluate

✓ **- 1 pts You only gave one example of what happens when you use a maxiumum depth that is too large e.g. overfitting, large trees requiring larger storage, and can be slow to evaluate**

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

3.5 Hyperparameter tuning 6 / 6

✓ **- 0 pts Correct**

- 2 pts Your reported accuracy numbers are significantly different from what is expected. Are you sure you used the correct random seed and the correct version
of sklearn?

- 1 pts You did not report the train set accuracy

- 1 pts You did not report the test set accuracy

- 2 pts You did not correctly identify why max_depth = 8 is the better model i.e.
you did not mention the overfitting that happens in the case of max_depth = 20 which clarifies why max_depth = 8 is best.

- **1 pts** You reported results with too many digits after the decimal place. Less than three would have been sufficient.
- **6 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized

3.6 Attribute importance 3 / 5

- **0 pts** Correct
- **2 pts** Your reported attributes are different from the expected answer of \`x50' \`y48', and \`y29'
- **1 pts** The order of your attributes is incorrect. It should be \`x50' \`y48', and \`y29'.
- **2 pts** You reported the indices of the attributes and not their names i.e. 100, 97, and 59 instead of \`x50' \`y48', and \`y29'.

✓ **- 2 pts You did not give a reason why the attributes were likely to be good choice in the context of the task. The most important attribute, \`x50', corresponds to the upper right lip. This is a part of the face that is likely to move a lot when someone smiles.**

- **5 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized
- **1 pts** You answered that the most important attribute does not make sense. \`x50' corresponds to the upper right lip and it is a part of the face that is likely to move a lot when someone smiles.
- **1 pts** You reported one incorrect attribute. The expected answer was \`x50' \`y48', and \`y29'.

3.7 Analysis 2 / 2

✓ **- 0 pts Correct**

- **2 pts** You did not give a sensible limitation of the choice of input feature encodings e.g. they are based on absolute pixel locations, it might be better to use relative distances between points or they are subject to noise if detected poorly.
- **1 pts** In your answer, you incorrectly said that the data only contains frontal views of faces, it actually contains side views as well
- **2 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized

QUESTION 4

Question 4 : Evaluating Binary Classifiers 14 pts

4.1 Classification accuracy 2 / 4

- **0 pts** Correct
- **2 pts** Your accuracy numbers are incorrect (or missing)
- **1 pts** You failed to state that alg_1 has the best performance
- **1 pts** You reported numbers in the range of 0 to 1. You should have used 0 to 100 as the questions asked for %

✓ **- 1 pts Your answer is too generic or you failed to mention a limitation of using a FIXED threshold i.e. the threshold might not be optimal for each of the different set of predictions**

✓ **- 1 pts You failed to give a better way of choosing the threshold for each model e.g. using a held out validation set**

- **0.5 pts** You included too few or many decimal places in your answer. Between one and two is more appropriate
- **4 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized

4.2 AUC 4 / 4

✓ **- 0 pts Correct**

- **2 pts** Your AUC numbers are wrong
- **1 pts** You did not correctly state that the model with the best accuracy does not have the best AUC score
- **2 pts** You did not identify the reason why alg_4 has a poor accuracy i.e. it is because 0.5 is a poor choice of threshold of this particular model
- **0.5 pts** You included too many decimal places in your answer. Four or less is more

appropriate, or two or less if you report area in %

- 4 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 2 pts This is not about overfitting or imbalance, it is about classification thresholds and impact on TPR/FPR.

- 0.5 pts Algorithm 3 AUC is 6.4%, not 64%.

- 1 pts Your answer would have been fine if you had not gone on to incorrectly discuss class imbalance - this is about classification thresholds.

4.3 ROC plots 6 / 6

✓ **- 0 pts Correct**

- 2 pts Your ROC curves do not look like what is expected

- 2 pts Your ROC curves are not smooth lines i.e. you only created the plot for the thresholded predictions

- 1 pts You did not plot the ROC curves for all four models.

- 1 pts You did not plot the four curves on the same plot, making it more difficult to compare them

- 1 pts Your plot is not clear i.e. you failed to label the axis and to provide a legend

- 1 pts You failed to describe the performance of alg_3 i.e. it performs much worse than random guessing

- 2 pts You failed to identify that alg_3 can be improved by inverting its predictions i.e. a prediction of 0 would become a prediction of 1

- 1 pts Answer too long/answer box resized

- 6 pts You did not answer the question

Question 1 : (22 total points) Linear Regression

In this question we will fit linear regression models to data.

(a) (3 points) Describe the main properties of the data, focusing on the size, data ranges, and data types.

The size of the data is (50,2) which means there are 50 rows and 2 columns.
The mean of *revision_time* is 22.22 and *exam_score* is 49.92. The standard deviation of *revision_time* is 13.99 and *exam_score* is 20.93.
The range of *revision_time* is from 2.72(min) to 48.01(max) and *exam_score* is from 14.73(min) to 94.95(max).
The datatype of both columns are float64(2).
All the float here are rounded to 2 decimal places.

Question 1 continued on next page...

1.1 Data properties 3 / 3

✓ **- 0 pts Correct**

- 1 pts You failed to mention the correct range of the revision time attributes (2.72 to 48.01)

- 1 pts You failed to mention the correct range of the exam score targets (14.73 to 94.94)

- 1 pts You failed to mention the size of the dataset (50 datapoints)

- 1 pts You failed to mention that the attributes are continuous

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 0.5 pts Your reported values for the data ranges are incorrect

- 1 pts The range has been specified correctly, but the min and max values have been omitted.

(b) (3 points) Fit a linear model to the data so that we can predict `exam_score` from `revision_time`. Report the estimated model parameters $\mathbf{w}$. Describe what the parameters represent for this 1D data. For this part, you should use the sklearn implementation of Linear Regression.
Hint: By default in sklearn `fit_intercept = True`*. Instead, set* `fit_intercept = False` *and pre-pend* 1 *to each value of* x_i *yourself to create* $\boldsymbol{\phi}(x_i) = [1, x_i]$.

$\mathbf{w}$ = [17.90, 1.44](rounded to 2 decimal places)
w0 = 17.90. **w0** is the intercept where our line intercepts the y-axis.
w1 = 1.44. **w1** is the coefficient for the Radio independent variable which is the revision_time in this question.

Question 1 continued on next page...

1.2 Linear Model 3 / 3

✓ **- 0 pts Correct**

- 2 pts Your reported model parameters are incorrect. They should be 17.898, and 1.441

- 1 pts You did not explain that the model parameters represented the intercept and slope

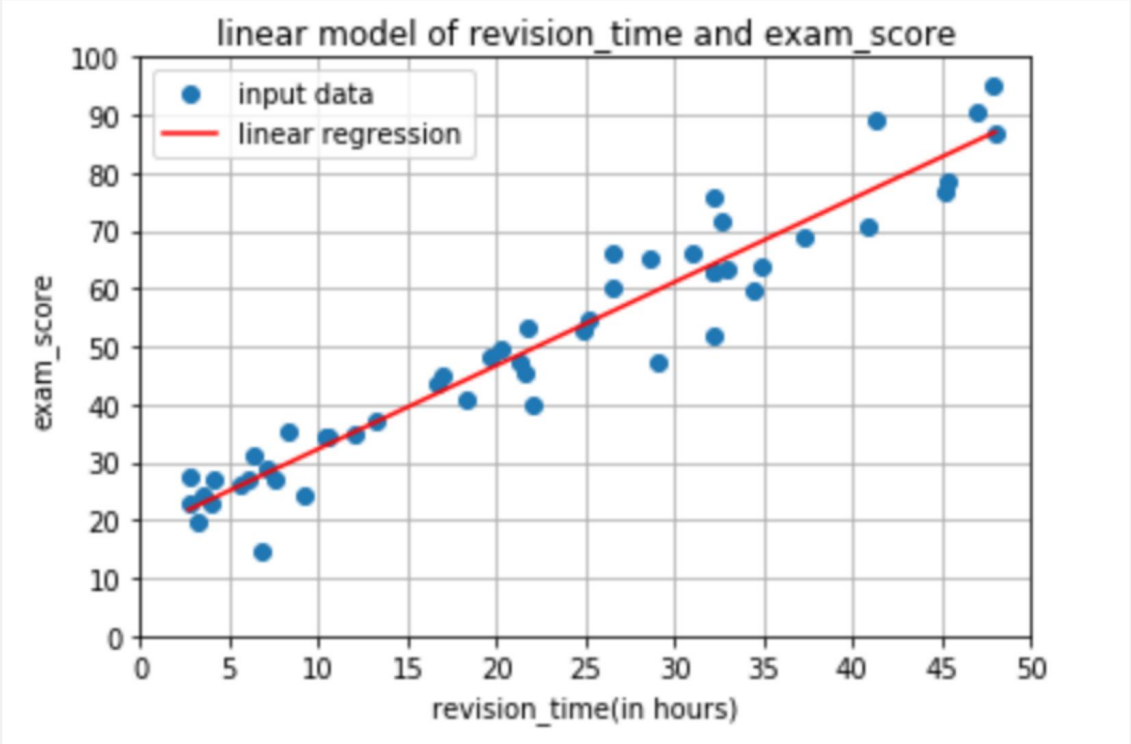
- 0.5 pts You included too many decimal places in your answer. Four or less is more appropriate

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 0.5 pts Your explanation of the model parameters goes towards the right direction, but is not quite there: you could simply say the model parameters represent the intercept and slope

(c) (3 points) Display the fitted linear model and the input data on the same plot.



Question 1 continued on next page. . .

1.3 Display 3 / 3

✓ **- 0 pts** **Correct**

- 1 pts Your linear model is a poor fit to the data

- 1 pts The fitted line is not continuous

- 1 pts You did not label the axis

- 0.5 pts You did not add a legend

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 1 pts You should not represent your input data using a line

(d) (3 points) Instead of using sklearn, implement the closed-form solution for fitting a linear regression model yourself using numpy array operations. Report your code in the answer box. It should only take a few lines (i.e. <5).
Hint: Only report the relevant lines for estimating **w** *e.g. we do not need to see the data loading code. You can write the code in the answer box directly or paste in an image of it.*

```
time_train = np.insert(regression.loc[:, ['revision_time']].values, 0, values = 1, axis = 1)
score_train = regression.loc[:, ['exam_score']].values
a = (np.linalg.inv(time_train.transpose().dot(time_train)))
w = a.dot(time_train.transpose()).dot(score_train)
```

Where regression is the DataFrame read by file 'regression_part1.csv'.
The **w** calculated by the closed-form solution is [17.90, 1.44].(rounded to 2 decimal places)
I use the method called normal equation.

Question 1 continued on next page...

1.4 Custom implementation 3 / 3

✓ **- 0 pts** **Correct**

- 3 pts Your code is obviously wrong e.g. you did not perform the pseudo inversion

- 2 pts You did not implement the solution using basic matrix operations e.g. you used np.linalg.lstsq

- 1 pts Your is overly long and does not make use of numpy expressions e.g. not using np.dot or np.linalg.inv

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 2 pts You did not implement the solution using basic matrix operations e.g. you used np.linalg.solve

- 2 pts You implemented linear regression only for 1D input attributes.

- 3 pts You fitted a polynomial.

(e) (3 points) Mean Squared Error (MSE) is a common metric used for evaluating the performance of regression models. Write out the expression for MSE and list one of its limitations.
Hint: For notation, you can use y for the ground truth quantity and $\hat{y}$ (`$\hat{y}$` in latex) in place of the model prediction.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

MSE will prone to outliers.
MSE calculates mean of square of errors. Mean will change a lot by some significant outliers, then MSE will also change a lot. MSE is prone to outliers.

Question 1 continued on next page...

1.5 MSE 3 / 3

✓ **- 0 pts** **Correct**

- 2 pts Missing square term

- 1 pts y-hat and/or y should have a consistent lowerscript

- 2 pts You added a square root term

- 1 pts Missing a limitation e.g. distorted by outliers or not in the same units as the data

- 1 pts You did not use the suggested notation for the ground truth and model prediction i.e. $$y$$ and $$\hat{y}$$

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 2 pts Missing $$\frac{1}{N}$$ term

(f) (3 points) Our next step will be to evaluate the performance of the fitted models using Mean Squared Error (MSE). Report the MSE of the data in `regression_part1.csv` for your prediction of `exam_score`. You should report the MSE for the linear model fitted using sklearn and the model resulting from your closed-form solution. Comment on any differences in their performance.

The MSE of sklearn is 30.9854726145413 and the MSE of closed_form is 30.98547261454129. The difference between two values is 1.07e14(rounded in 2 decimal places) difference. The reason why they have difference is for sklearn the w = [17.897680258350174, 1.441140905437971] and for normal equation the w = [17.897680258350192, 1.4411409054379702]. Also calculating normal equation is numerically unstable. Hence the two MSE have a slightly difference. Basically they have the same performance.

Question 1 continued on next page...

1.6 MSE 2 **2.5 / 3**

- **0 pts** Correct

✓ - **0.5 pts** **You could comment on why MSEs are the same/different**

- **0.5 pts** You included too many decimal places in your answer. Four or less is more appropriate

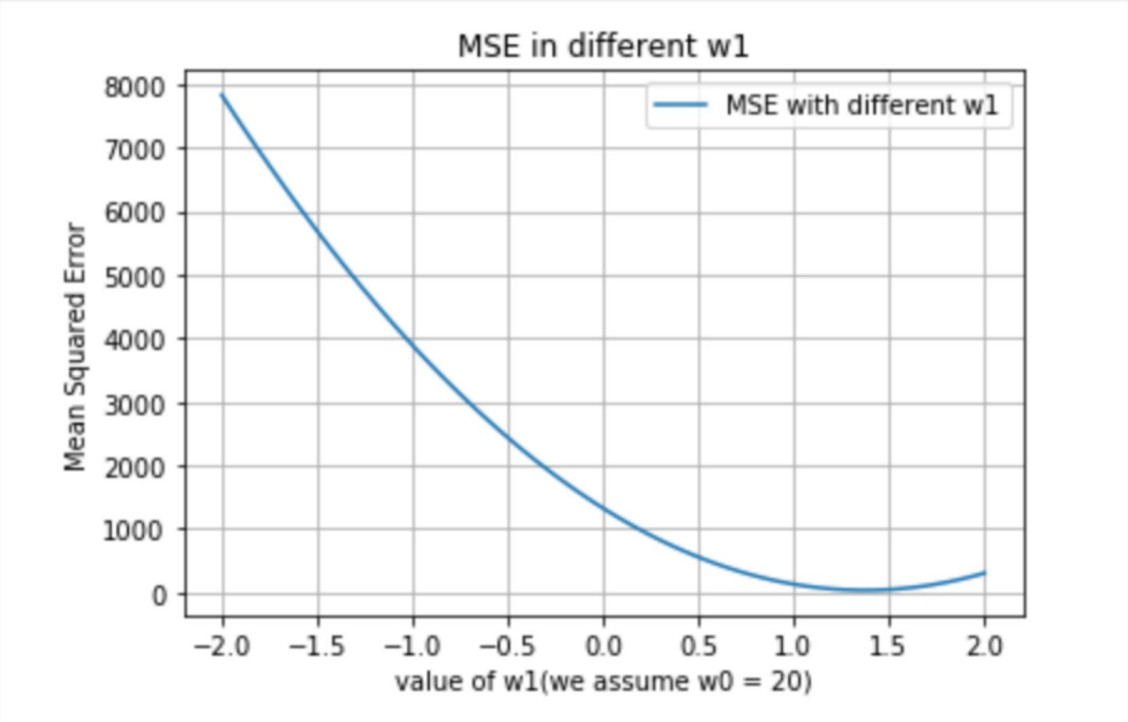
- **1 pts** You MSE values is wrong. It should be 30.985

- **1 pts** For having different MSE scores for both approaches. They should be very similar for the most significant decimal places

- **3 pts** You did not answer the question

- **1 pts** Answer too long/answer box resized

(g) (4 points) Assume that the optimal value of w_0 is 20, it is not but let's assume so for now. Create a plot where you vary w_1 from -2 to $+2$ on the horizontal axis, and report the Mean Squared Error on the vertical axis for each setting of $\mathbf{w} = [w_0, w_1]$ across the dataset. Describe the resulting plot. Where is its minimum? Is this value to be expected? *Hint: You can try 100 values of* w_1 *i.e.* `w1 = np.linspace(-2,2, 100)`.



The minimum MSE is 32.48(rouned in 2 decimal places) and the corresponding w1 is 1.35(rouned in 2 decimal places).
This value is expected. Because after this value, we can see that the graph begin to increasing monotonically. Hence the value is the global minimum in the MSE of this question.

1.7 Analysis 2 / 4

- **0 pts** Correct
- **2 pts** You failed to include the plot.
- **1 pts** The range of MSE is incorrect (should be around 8000 for $$w_1 = -2.0$$)
- **2 pts** The plot does not match the correct one in some aspect of the data
- **1 pts** The plot is unclear or insufficiently labelled i.e. you did not label the axis.
- **1 pts** You failed to identify the minimum value of MSE being close to 32.48 or the corresponding $$w_1$$ being close to 1.35

✓ **- 1 pts** **You failed to mention that the plot was convex/that there is a single minimum rather than multiple minima.**

- **0.5 pts** You have identified the plot as concave rather than convex.

✓ **- 1 pts** **You failed to mention that the resulting minimum $$w_1$$ value was to be expected to be similar to one we found earlier (i.e. close to 1.4)/otherwise refer to the previously calculated result of $$w_1$$ in your argument.**

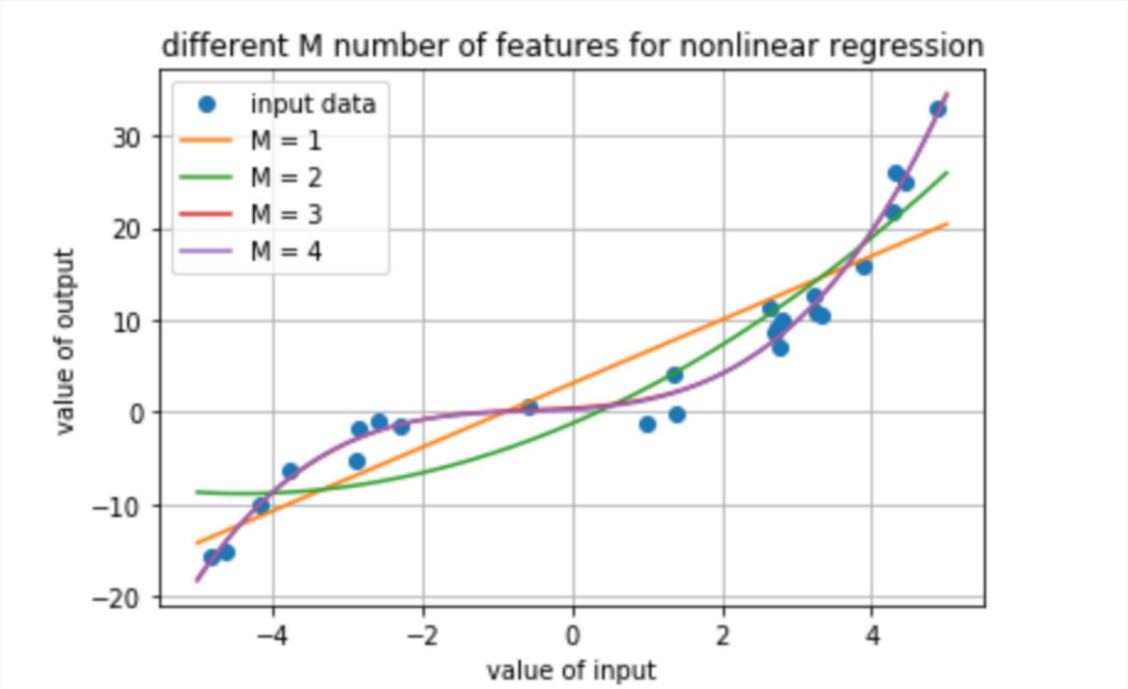
- **0.5 pts** You included too many decimal places in your answer. Four or fewer is more appropriate.
- **1 pts** Answer too long/answer box resized
- **4 pts** You did not answer the question

You are justifying the global minimum with monotonic increase after 1.35. This argument is incomplete, as it should have been preceded with noting that the function is decreasing until 1.35 - alternatively you could have said that the function is convex.

Question 2 : (18 total points) Nonlinear Regression

In this question we will tackle regression using basis functions.

(a) (5 points) Fit four different polynomial regression models to the data by varying the degree of polynomial features used i.e. $M = 1$ to 4. For example, $M = 3$ means that $\boldsymbol{\phi}(x_i) = [1, x_i, x_i^2, x_i^3]$. Plot the resulting models on the same plot and also include the input data.
Hint: You can again use the sklearn implementation of Linear Regression and you can also use PolynomialFeatures to generate the polynomial features. Again, set `fit_intercept = False`.



The regression line of M = 3 and M = 4 is basically the same line. So that is why in the graph we could not find thes red line.

Question 2 continued on next page...

2.1 Polynomial regression 5 / 5

✓ **- 0 pts** **Correct**

- 2 pts Your model predictions look very dfferent from the expected answer

- 1 pts The plots for M = 3 and M = 4 are very different. In this example, they should look almost identical.

- 2 pts You did not plot a continuous line for the models

- 1 pts You did not plot the input data

- 1 pts Your plot is unclear e.g. missing/wrong axes or a legend

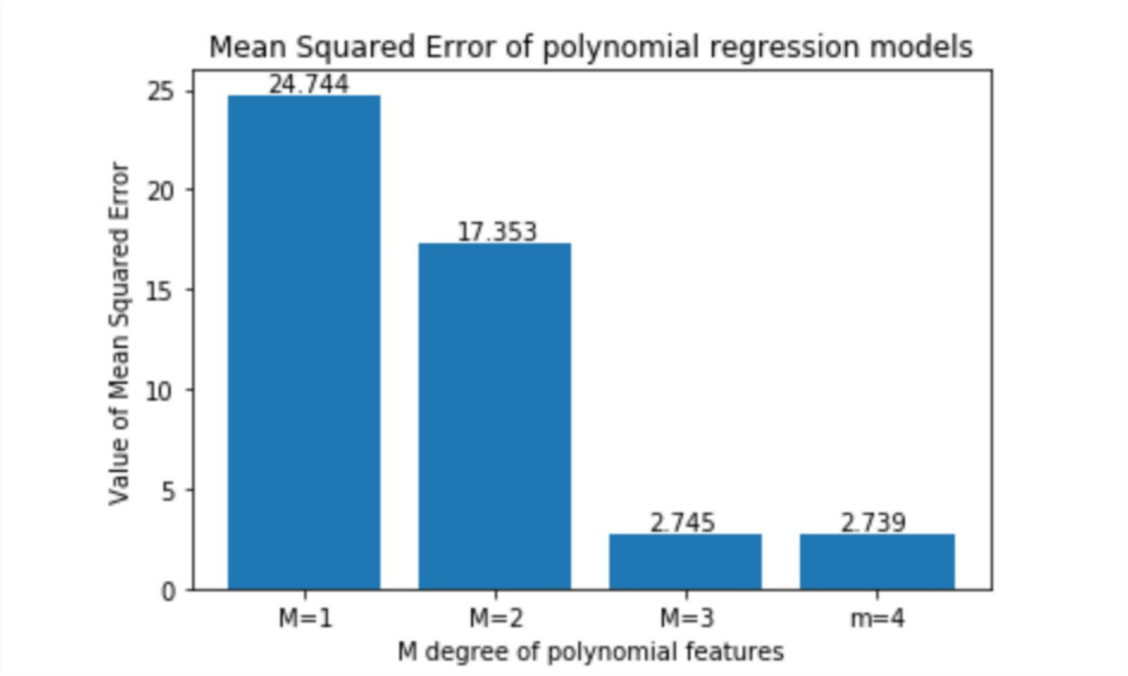
- 5 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 0.5 pts The input data should be plotted as points as there is no connection among them

- 3 pts You did not plot some of the lines for the models

(b) (3 points) Create a bar plot where you display the Mean Squared Error of each of the four different polynomial regression models from the previous question.



Note: Values above each the bar plot have rounded to 3 decimal places.

Question 2 continued on next page...

2.2 Bar plot 3 / 3

✓ **- 0 pts Correct**

- 2 pts Your bar plot looks very different from the expected answer e.g. M = 3 and M = 4 should have very similar values for MSE, but both are less than M = 1 or M = 2

- 1 pts Your estimated MSE values are incorrect e.g. M = 1 should be around 24.7

- 1 pts Your plot is not clear e.g. missing axes

- 3 pts You did not answer the question

- 1 pts Answer too long/answer box resized

(c) (4 points) Comment on the fit and Mean Squared Error values of the $M = 3$ and $M = 4$ polynomial regression models. Do they result in the same or different performance? Based on these results, which model would you choose?

MSE(When $M = 3$) = 2.745(rounded to 3 decimal places)
MSE(When $M = 4$) = 2.739(rounded to 3 decimal places)
Two MSE are basically the same value. Hence they have same performance. MSE will decrease when the number of degree of polynomial features increase. The difference between MSE of $M = 3$ and $M = 4$ is much samller than the difference between MSE of $M = 2$ and $M = 3$. The MSE is nearly no decreasing between M is 3 and 4. This means the weights on x^4 is small. $M = 3$ will be better(There is no need to add another feature (x^4), also this might cause overfitting while adding too many useless degree of polynomial features).
Hence I will choose model whose $M = 3$.

Question 2 continued on next page...

2.3 Analysis 4 / 4

✓ **- 0 pts Correct**

- 1 pts You failed to mention that M = 3 and M = 4 give very similar predictions.

- 2 pts You said that the M = 4 model is better. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

- 1 pts You did not mention that the weight vector entry corresponding to x 4 i is very small.

- 4 pts You did not answer the question

- 1 pts Answer too long/answer box resized

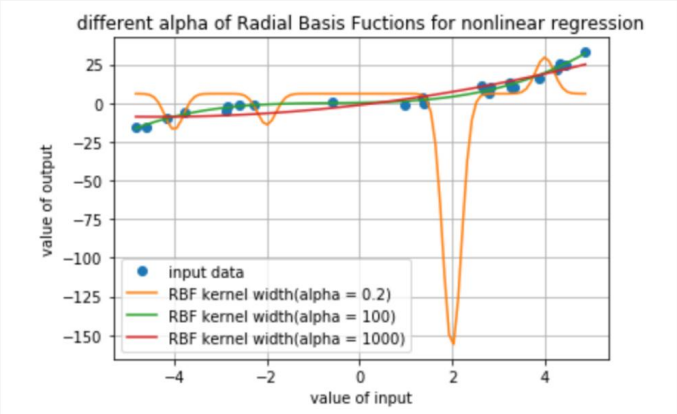
- 1 pts you did not explain correctly why M = 3 model is better. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

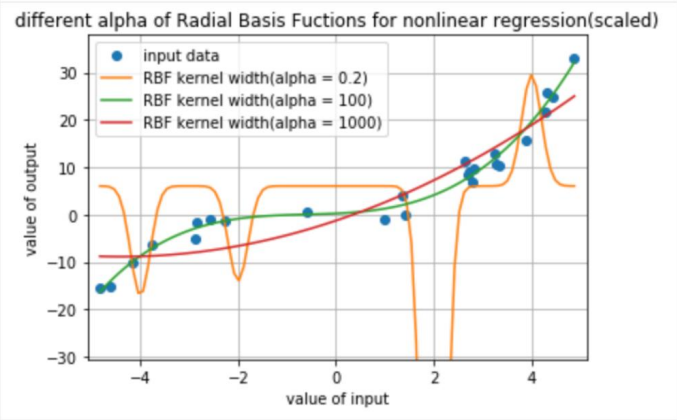
- 2 pts you did not answer which model to choose. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

- 1 pts you did not explain correctly why M=3 is better. You don't have sufficient evidence to use overfitting as an argument. To comment on overfitting a model should be evaluated on heldout/unseen data. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

Here you should not mention of overfitting. To comment on overfitting a model should be evaluated on held-out/unseen data. The lower parameter M = 3 is a good trade off between complexity and performance i.e. it requires less parameters.

(d) (6 points) Instead of using polynomial basis functions, in this final part we will use another type of basis function - radial basis functions (RBF). Specifically, we will define $\boldsymbol{\phi}(x_i) = [1, rbf(x_i; c_1, \alpha), rbf(x_i; c_2, \alpha), rbf(x_i; c_3, \alpha), rbf(x_i; c_4, \alpha)]$, where $rbf(x; c, \alpha) = \exp(-0.5(x - c)^2/\alpha^2)$ is an RBF kernel with center c and width α. Note that in this example, we are using the same width α for each RBF, but different centers for each.
Let $c_1 = -4.0$, $c_2 = -2.0$, $c_3 = 2.0$, and $c_4 = 4.0$ and plot the resulting nonlinear predictions using the `regression_part2.csv` dataset for $\alpha \in \{0.2, 100, 1000\}$. You can plot all three results on the same figure. Comment on the impact of larger or smaller values of α.





$\alpha = 100$ is optimal, while $\alpha = 100$ the graph is more fitting than α is too small($\alpha = 0.2$) or too large($\alpha = 1000$). Larger or smaller values of α will both cause the descendent of the testing ability of the model, this could also be seen as underfitting.

2.4 RBF 4 / 6

- **0 pts** Correct
- **3 pts** Your plot looks very different from the expected answer
- **2 pts** You only plotted your model predictions where the input data was. You should have used more input points to better visualize the predictions i.e. to make the plot more continuous

✓ **- 1 pts** **You failed to mention what would happen if the width parameter is too large i.e. more datapoints further away from the basis center are included resulting in underfitttting (or overly smooth predictions)**

✓ **- 1 pts** **You failed to mention what would happen if the width parameter is too small i.e. model predictions will be a constant for points that are not close to the kernel center**

- **1 pts** Your plot is not very clear e.g. you did not plot the input data, it is missing a legend, axis labels or the plot is not complete
- **6 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized
- **3 pts** You failed to provide the figure
- **2 pts** Some of the plotted graphs are different from the expected answer

Question 3 : (26 total points) Decision Trees

In this question we will train a classifier to predict if a person is smiling or not.

(a) (4 points) Load the data, taking care to separate the target binary class label we want to predict, `smiling`, from the input attributes. Summarise the main properties of both the training and test splits.

There are 4800 rows in training data and 1200 rows in test data. There are 137 columns(different features) in both data-set. Both have 136 columns for training data which type are float and 1 column for label which type is int(it is binary (0 or 1)).
Training splits: The size of faces_train(`smiling = 0`) is (2465, 137) and the size of faces_train(`smiling = 1`) is (2335, 137).
Test splits: The size of faces_test(`smiling = 0`) is (608, 137) and the size of faces_test(`smiling = 1`) is (592, 137).

Question 3 continued on next page...

3.1 Dataset analysis 4 / 4

✓ **- 0 pts Correct**

- 1 pts You reported the wrong number of attributes. It should have been 136

- 0.5 pts You did not report the correct train set size (4800)

- 0.5 pts You did not report the correct test set size (1200)

- 0.5 pts You did not report the correct number of positive labels in the train set (2335)

- 0.5 pts You did not report the correct number of negative labels in the train set (2465)

- 0.5 pts You did not report the correct number of positive labels in the test set (592)

- 0.5 pts You did not report the correct number of negative labels in the test set (608)

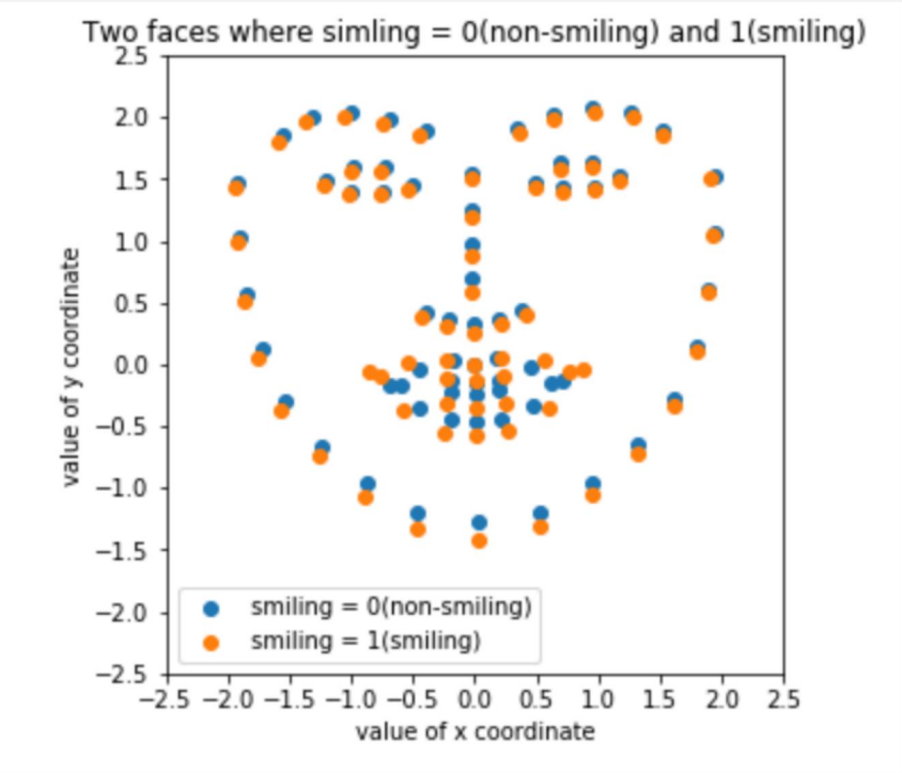
- 4 pts You did not answer the question

- 1 pts Answer too long/answer box resized

- 1 pts You only reported the ratio of smiling faces to total faces, not the total number of smiling and not smiling faces.

- 1 pts You only reported the ratio of smiling faces to not smiling faces, not the total number of smiling and not smiling faces.

(b) (4 points) Even though the input attributes are high dimensional, they actually consist of a set of 2D coordinates representing points on the faces of each person in the dataset. Create a scatter plot of the average location for each 2D coordinate. One for (i) smiling and (ii) one not smiling faces. For instance, in the case of smiling faces, you would average each of the rows where `smiling = 1`. You can plot both on the same figure, but use different colors for each of the two cases. Comment on any difference you notice between the two sets of points.
Hint: Your plot should contain two faces.



The corner of the mouth of graph(`smiling = 1`) is upward whereas the mouth corner of graph(`smiling = 0`) is non-smiling. Also the chin of the mouth of graph(`smiling = 1`) is more downward than graph(`smiling = 0`). The graph(`smiling = 1`) distance between the upper and lower lips is further than graph(`smiling = 0`).

Question 3 continued on next page...

3.2 Analysis 4 / 4

✓ **- 0 pts** **Correct**

- 2 pts You failed to mention any difference between the two sets of points e.g. the mouth is wider and the chin is lower on the smiling face

- 1 pts You did not mention the difference clearly

- 3 pts The plot does not contain two faces

- 1 pts You did not plot both faces on the same figure. This makes it difficult to see the differences

- 1 pts You did not include a legend or way to indicate which points correspond to the train set and which correspond to the test

- 1 pts You plot is missing axes labels, you plotted lines instead of points, etc.

- 4 pts You did not answer the question

- 1 pts Answer too long/answer box resized

(c) (2 points) There are different measures that can be used in decision trees when evaluating the quality of a split. What measure of purity at a node does the DecisionTreeClassifier in sklearn use for classification by default? What is the advantage, if any, of using this measure compared to entropy?

Gini Impurity.
Since computing square is cheaper than logarithmic function we prefer Gini impurity over entropy.

Question 3 continued on next page...

3.3 Decision Trees 2 / 2

✓ **- 0 pts** **Correct**

- 1 pts You did not specify the correct measure used by sklearn by default i.e. gini

- 1 pts You did not specify an advantage of gini over entropy e.g. computing entropy requires more computation as you need to take logarithms

- 2 pts You did not answer the question

- 1 pts Answer too long/answer box resized

(d) (3 points) One of the hyper-parameters of a decision tree classifier is the maximum depth of the tree. What impact does smaller or larger values of this parameter have? Give one potential problem for small values and two for large values.

Large depth values might cause overfitting. The decision tree might overfit the training data without capturing useful patterns as we would like and will operate in a very low speed. Also it might fit outliers as well. Too much focus on the training set and learns complex relations which may not be valid in general for new data, this will cause testing error to increase. Small depth values might cause underfitting. It might be giving the decision tree too little flexibility to capture the patterns and interactions in the training data and cause the testing error to increase.

Question 3 continued on next page...

3.4 DT Depth 2 / 3

- **0 pts** Correct
- **1 pts** You did not mention what happens when you use a maxiumum depth that is too small e.g. underfitting
- **2 pts** You did not give two examples of what happens when you use a maxiumum depth that is too large e.g. overfitting, large trees requiring larger storage, and can be slow to evaluate

✓ **- 1 pts** **You only gave one example of what happens when you use a maxiumum depth that is too large e.g. overfitting, large trees requiring larger storage, and can be slow to evaluate**

- **3 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized

(e) (6 points) Train three different decision tree classifiers with a maximum depth of 2, 8, and 20 respectively. Report the maximum depth, the training accuracy (in %), and the test accuracy (in %) for each of the three trees. Comment on which model is best and why it is best.
Hint: Set `random_state = 2001` *and use the* `predict()` *method of the DecisionTreeClassifier so that you do not need to set a threshold on the output predictions. You can set the maximum depth of the decision tree using the* `max_depth` *hyper-parameter.*

Maximum depth	Training Accuracy	Test Accuracy
2	79.5%	78.2%
8	93.4%	84.1%
20	100%	81.6%

(accuracy rounded in 1 decimal places)
Model whose max depth is 8 is best. model(depth = 2) is underfitting, the accuracy of training data is 79.5% and test data is 78.2%. Those values are neither high enough, which means this model is too simple. Model(depth = 20) is overfitting, the accuracy of training data is 100% and test data is 81.6%. The model focus too much on the training set and learns complex relations which may not be valid in general for test set(The accuracy fall from 100% to 81.6%, this is a significant dencent). Model(depth = 8) is fitting well, the accuracy of training data is 93.4% and test data is 84.1%. Those two values do not have too much difference(9.3%) which means this model neither underfitting nor overfitting.

Question 3 continued on next page...

3.5 Hyperparameter tuning 6 / 6

✓ **- 0 pts Correct**

- 2 pts Your reported accuracy numbers are significantly different from what is expected. Are you sure you used the correct random seed and the correct version of sklearn?

- 1 pts You did not report the train set accuracy

- 1 pts You did not report the test set accuracy

- 2 pts You did not correctly identify why max_depth = 8 is the better model i.e. you did not mention the overfitting that happens in the case of max_depth = 20 which clarifies why max_depth = 8 is best.

- 1 pts You reported results with too many digits after the decimal place. Less than three would have been sufficient.

- 6 pts You did not answer the question

- 1 pts Answer too long/answer box resized

(f) (5 points) Report the names of the top three most important attributes, in order of importance, according to the Gini importance from DecisionTreeClassifier. Does the one with the highest importance make sense in the context of this classification task?
Hint: Use the trained model with `max_depth = 8` *and again set* `random_state = 2001`.

Importance: 1st: $x50$; 2nd: $y48$; 3rd: $y29$.
The mean of $x50$ is -0.181 in `smiling = 0` split and -0.221. The information gain of $x50$ is the greatest among all features. The mean difference between two splits is quite large. The standard deviation of $x50$ is 0.033 in `smiling = 0` split and 0.032 in `smiling = 1` split. Both std are very small compared with other features. So $x50$ make sense.(Values above all rounded to 3 decimal places)

Question 3 continued on next page...

3.6 Attribute importance 3 / 5

- **0 pts** Correct
- **2 pts** Your reported attributes are different from the expected answer of \`x50' \`y48', and \`y29'
- **1 pts** The order of your attributes is incorrect. It should be \`x50' \`y48', and \`y29'.
- **2 pts** You reported the indices of the attributes and not their names i.e. 100, 97, and 59 instead of \`x50' \`y48', and \`y29'.

✓ **- 2 pts** **You did not give a reason why the attributes were likely to be good choice in the context of the task. The most important attribute, \`x50', corresponds to the upper right lip. This is a part of the face that is likely to move a lot when someone smiles.**

- **5 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized
- **1 pts** You answered that the most important attribute does not make sense. \`x50' corresponds to the upper right lip and it is a part of the face that is likely to move a lot when someone smiles.
- **1 pts** You reported one incorrect attribute. The expected answer was \`x50' \`y48', and \`y29'.

(g) (2 points) Are there any limitations of the current choice of input attributes used i.e. 2D point locations? If so, name one.

Yes. The face might rotate and different faces might have different scales.
If the faces rotate or have different scales, it will be hard to find the main features.

3.7 Analysis 2 / 2

✓ **- 0 pts** **Correct**

- 2 pts You did not give a sensible limitation of the choice of input feature encodings e.g. they are based on absolute pixel locations, it might be better to use relative distances between points or they are subject to noise if detected poorly.

- 1 pts In your answer, you incorrectly said that the data only contains frontal views of faces, it actually contains side views as well

- 2 pts You did not answer the question

- 1 pts Answer too long/answer box resized

Question 4 : (14 total points) Evaluating Binary Classifiers

In this question we will perform performance evaluation of binary classifiers.

(a) (4 points) Report the classification accuracy (in %) for each of the four different models using the `gt` attribute as the ground truth class labels. Use a threshold of >= 0.5 to convert the continuous classifier outputs into binary predictions. Which model is the best according to this metric? What, if any, are the limitations of the above method for computing accuracy and how would you improve it without changing the metric used?

Model Name	alg_1	alg_2	alg_3	alg_4
Accuracy	61.6%	55.0%	32.1%	32.9%

(values rounded to 1 decimal places).
As the greatest classification accuracy among four models is 61.6%, hence alg_1 model is the best according to this metric. Limitations: If the class is imbalanced, then the result accuracy is pointless. Accuracy is related to the number of TP and TN , which will changed by setting different thresholds. Improvement: We need to set different values of threshold then compare the accuracy of each model.(In this task, as number of 0(798) in *gt* is much greater than number of 1(202) in *gt*, therefore we need to increase the threshold to make it greater than 0.5)

Question 4 continued on next page...

4.1 Classification accuracy 2 / 4

- **0 pts** Correct
- **2 pts** Your accuracy numbers are incorrect (or missing)
- **1 pts** You failed to state that alg_1 has the best performance
- **1 pts** You reported numbers in the range of 0 to 1. You should have used 0 to 100 as the questions asked for %

✓ **- 1 pts Your answer is too generic or you failed to mention a limitation of using a FIXED threshold i.e. the threshold might not be optimal for each of the different set of predictions**

✓ **- 1 pts You failed to give a better way of choosing the threshold for each model e.g. using a held out validation set**

- **0.5 pts** You included too few or many decimal places in your answer. Between one and two is more appropriate
- **4 pts** You did not answer the question
- **1 pts** Answer too long/answer box resized

(b) (4 points) Instead of using classification accuracy, report the Area Under the ROC Curve (AUC) for each model. Does the model with the best AUC also have the best accuracy? If not, why not?
Hint: You can use the roc_auc_score function from sklearn.

Model Name	alg_1	alg_2	alg_3	alg_4
AUC	0.73	0.63	0.06	0.85

(Values rounded to 2 decimal places)
No. The greatest value of AUC is 0.85 which is alg_4 model, but the greatest value of classification accuracy among four models is 61.6% which is alg_1 model.
Accuracy will be pointless if model is imbalanced, accuracy only correspond with the number of TP plus TN. This accuracy is computed with the fixed threshold which is 0.5. But AUC is a combination of all the accuracy for all threshold values. AUC consider all the situations which use different thresholds.

Question 4 continued on next page...

4.2 AUC 4 / 4

✓ **- 0 pts Correct**

- 2 pts Your AUC numbers are wrong

- 1 pts You did not correctly state that the model with the best accuracy does not have the best AUC score

- 2 pts You did not identify the reason why alg_4 has a poor accuracy i.e. it is because 0.5 is a poor choice of threshold of this particular model

- 0.5 pts You included too many decimal places in your answer. Four or less is more appropriate, or two or less if you report area in %

- 4 pts You did not answer the question

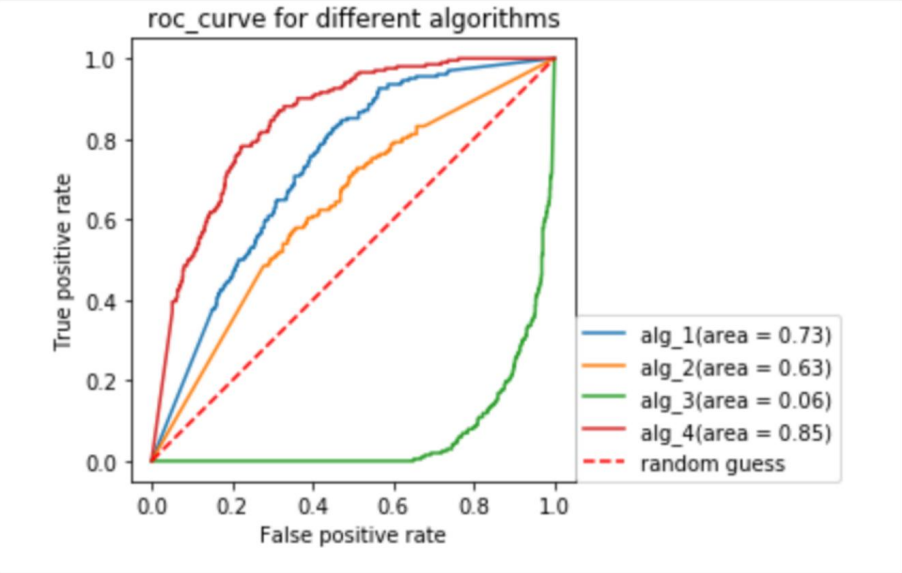
- 1 pts Answer too long/answer box resized

- 2 pts This is not about overfitting or imbalance, it is about classification thresholds and impact on TPR/FPR.

- 0.5 pts Algorithm 3 AUC is 6.4%, not 64%.

- 1 pts Your answer would have been fine if you had not gone on to incorrectly discuss class imbalance - this is about classification thresholds.

(c) (6 points) Plot ROC curves for each of the four models on the same plot. Comment on the ROC curve for `alg_3`? Is there anything that can be done to improve the performance of `alg_3` without having to retrain the model?
Hint: You can use the roc_curve function from sklearn.



The curve of alg_3 model is below the random guess curve, this means alg_3 model is not a suitable model. The labels of alg_3 model are reverse, which means that the model predict the high labels to 0 and small labels to 1. In order to fix this, we need to convert all the positive predictions to negative predictions and negative predictions to positive predictions. Hence after converting the continuous classifier outputs into binary predictions, I changed each label of alg_3 model from 1 to 0, and 0 to 1. Thus, the ROC of new alg_3 model will fold along with the line of random guess and the AUC of new alg_3 model became 0.94(1-0.06)(rounded to 2 decimal places), which become a suitable model now.

4.3 ROC plots 6 / 6

✓ **- 0 pts** **Correct**

- 2 pts Your ROC curves do not look like what is expected

- 2 pts Your ROC curves are not smooth lines i.e. you only created the plot for the thresholded predictions

- 1 pts You did not plot the ROC curves for all four models.

- 1 pts You did not plot the four curves on the same plot, making it more difficult to compare them

- 1 pts Your plot is not clear i.e. you failed to label the axis and to provide a legend

- 1 pts You failed to describe the performance of alg_3 i.e. it performs much worse than random guessing

- 2 pts You failed to identify that alg_3 can be improved by inverting its predictions i.e. a prediction of 0 would become a prediction of 1

- 1 pts Answer too long/answer box resized

- 6 pts You did not answer the question