IAML – INFR10069 (LEVEL 10): Assignment #1

s1824891

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.

From what is displayed after loading the dataset, it can be seen that there are 50 entries ranging from 0 to 49 typed 'float64' and 2 attributes.

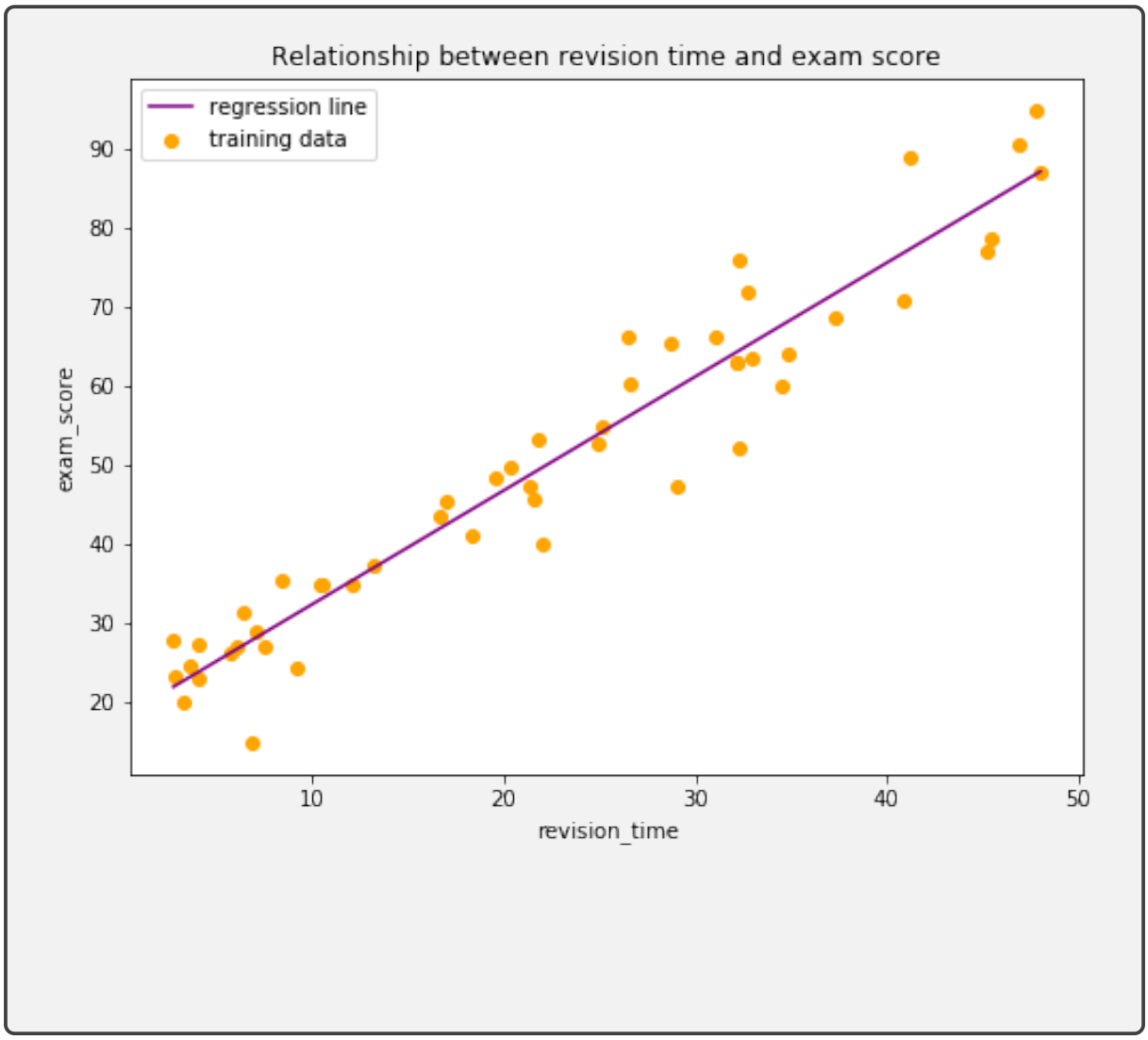
Question 1 continued on next page...

(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* $\boldsymbol{\phi}(x_i) = [1, x_i]$.

The obtained parameters are [17.89768026, 1.44114091], where the first element represents the intersection of the line with the y-axis which is the exam score, and the second element represents the slope of the regression line of revision time against exam score.

Question 1 continued on next page...

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



Question 1 continued on next page. . .

(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.*

```
w = np.dot(np.linalg.inv(X.T.dot(X)), X.T.dot(y))
```

Question 1 continued on next page...

(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 ŷ (`$\hat{y}$` in latex) in place of the model prediction.

$MSE = \frac{1}{n}\Sigma_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$
Limitation: MSE is sensitive to outliers.

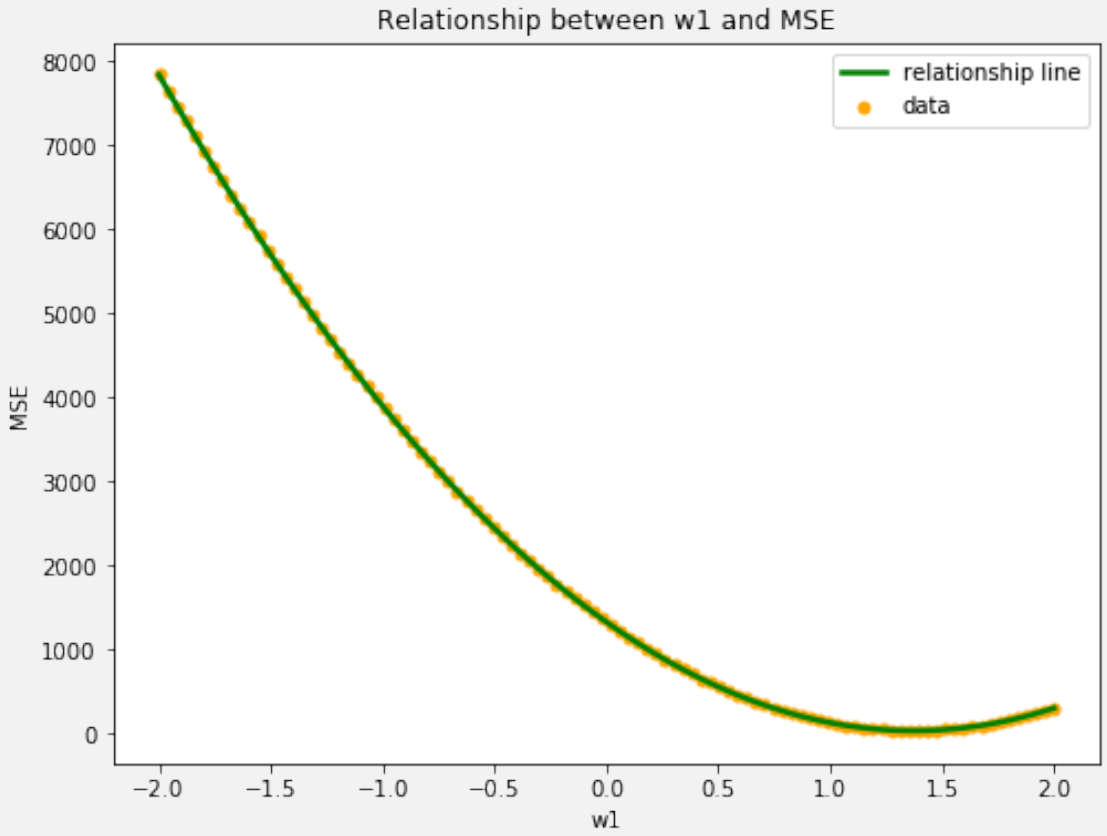
Question 1 continued on next page...

(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.

MSE prediction using scikit-learn is 30.98547261454129 and MSE prediction using closed-form is 30.985472614541287. There is no much difference between these two values because the values of w in both models are identical. The MSE computed by the closed-form solution has one more decimal, whereas the MSE computed by sklearn is rounded up to 14 decimal places. Thus, from my point of view, I reckon the model of the closed-form is more precise in this part, as the result computed by it has an extra digit.

Question 1 continued on next page...

(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)`.

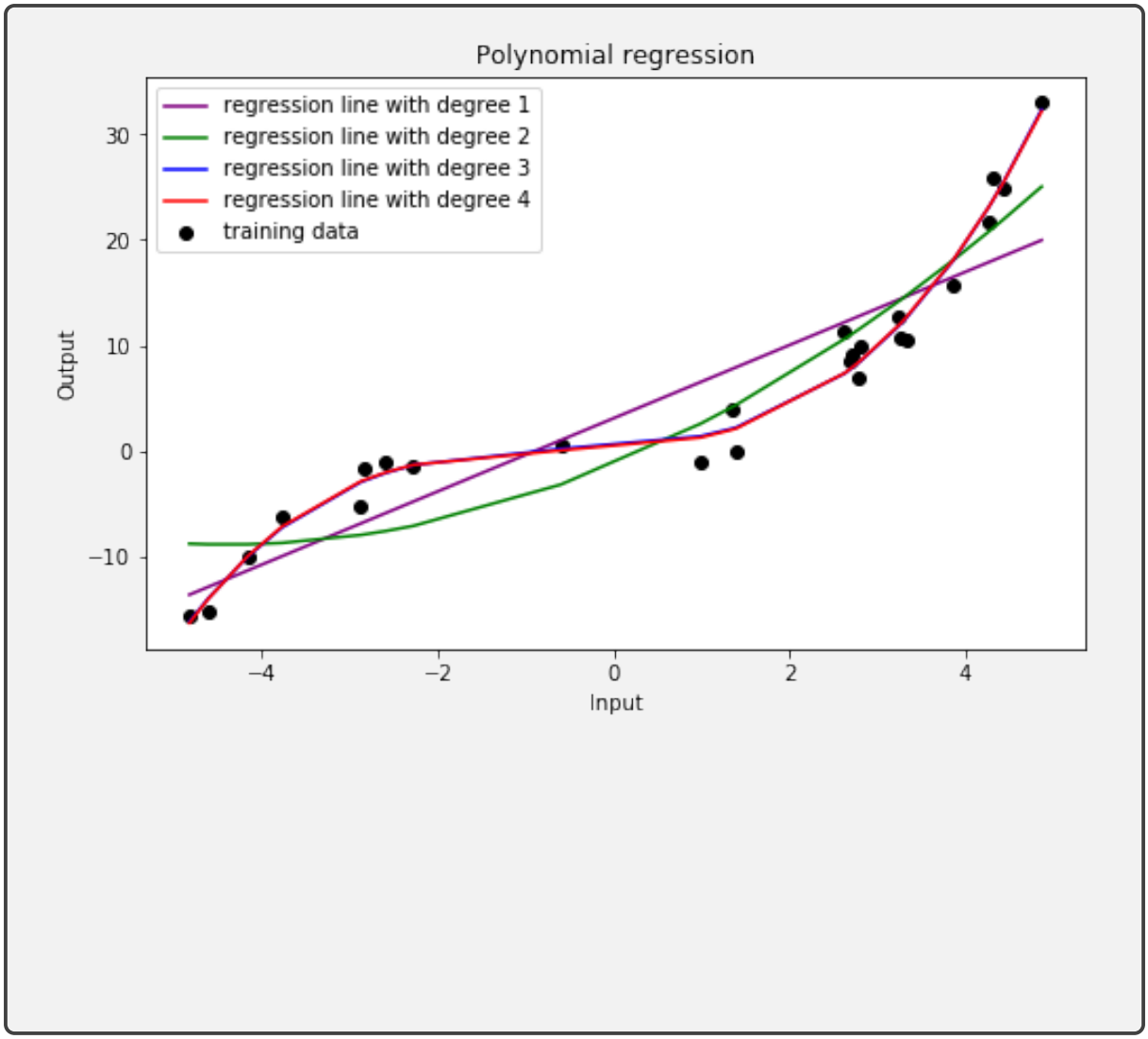


The generated plot shows a reduction of MSE at first as w1 increases until the line passes through the point where the minimum MSE occurs and after that, it starts slightly increasing. The minimum MSE is calculated to be 32.48096161535148 with its relevant index on the x-axis to be 1.354 rounded to 3 decimal places. So, I would say 1.354 is reasonable for being the minimum of the plot.

Question 2 : (18 total points) Nonlinear Regression

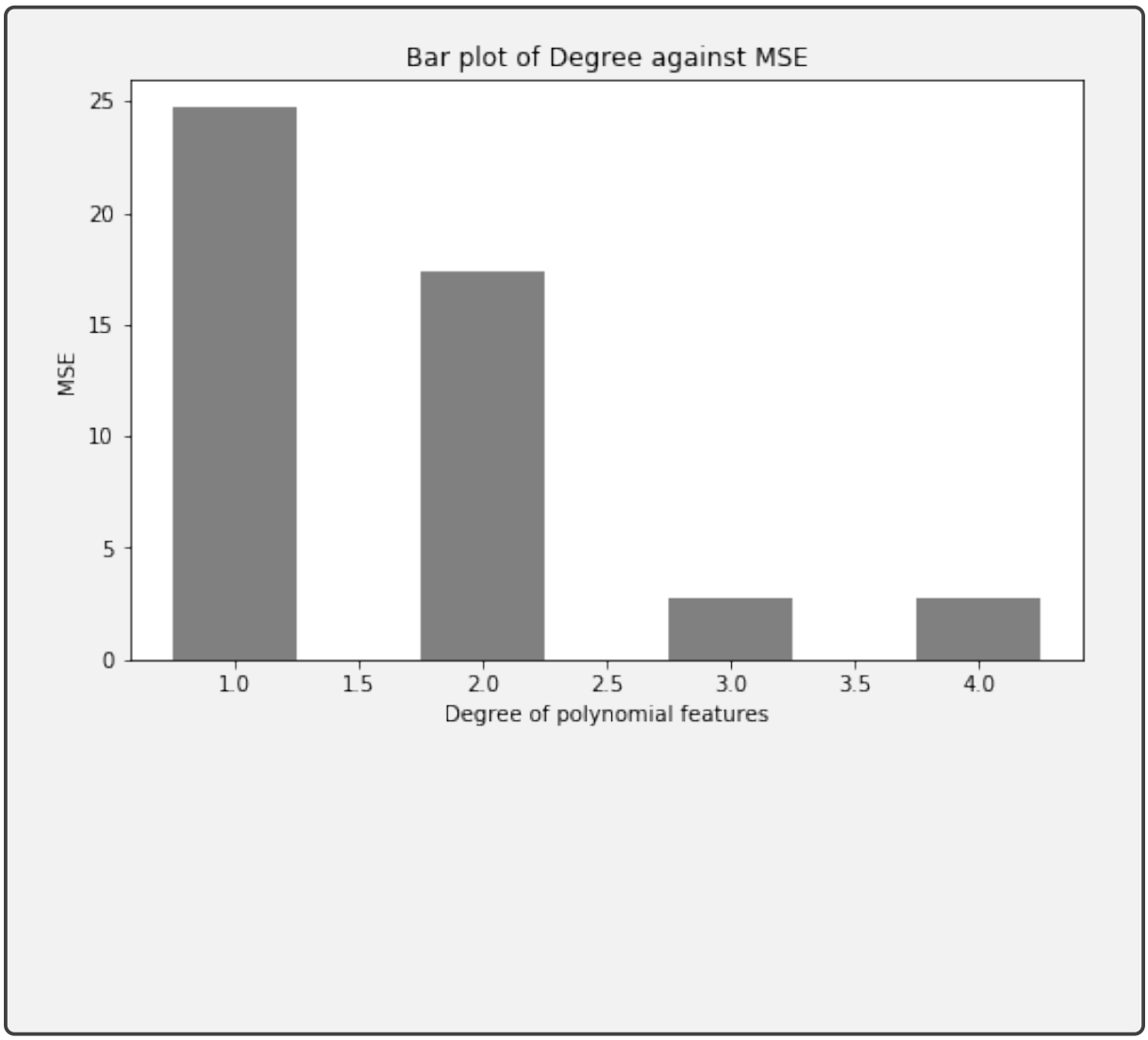
In this question we will tackle regression using basis functions.

(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`.



Question 2 continued on next page...

(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.



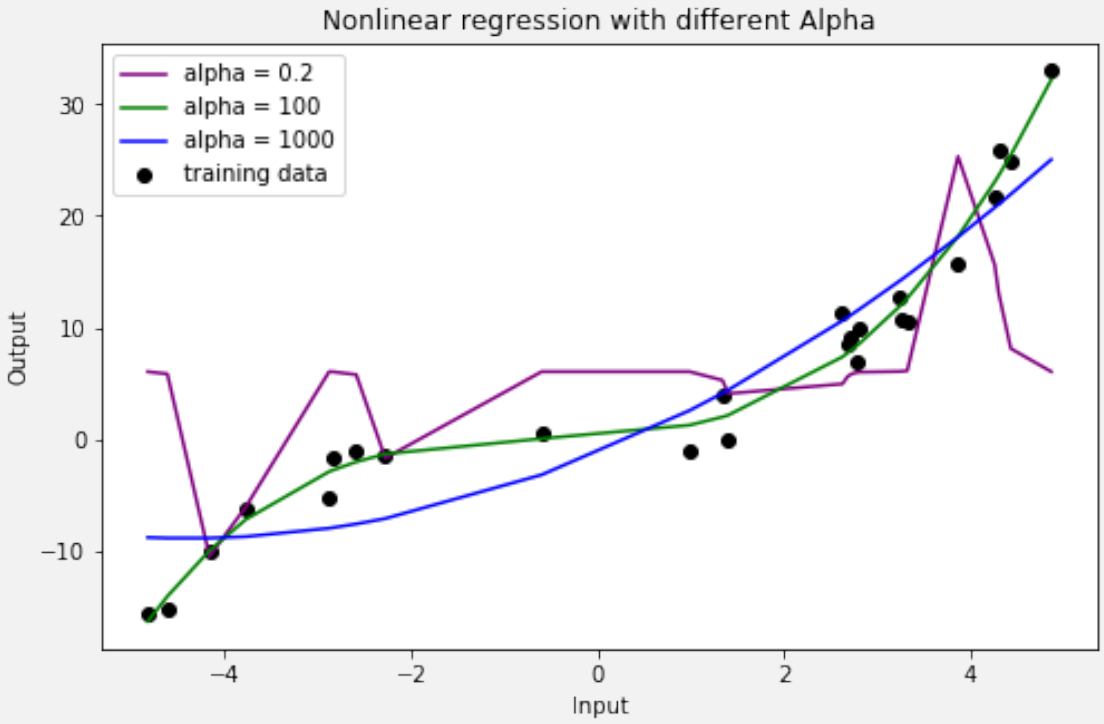
Question 2 continued on next page...

(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?

The MSE for degree equal to 3 is 2.7447567192524263 and the MSE value for degree equal to 4 is 2.7389111790755374. There is no much difference between these two values except for the decimals. Likewise, from the plotted polynomial regression graph, it can be seen that the blue line (regression line with degree = 3) and the red line (regression line with degree = 4) are approximately overlapped. Besides, the two bars referring to degree equal to 3 and degree equal to 4 respectively are almost on the same horizontal level on the bar plot. Which means the two polynomial regression models nearly result in the same performance. In my personal opinion, I would choose the model with the degree of polynomial features equal to 3 as for degree equal to 4, the greater degree the more likely for overfitting to occur.

Question 2 continued on next page...

(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.



It can be seen from the generated plot, the larger alpha value the more likely for the model to form a line of the best fit. However and theoretically, the smaller alpha value the more likely underfitting would occur, and the larger alpha value the more likely overfitting would occur.

Question 3 : (26 total points) Decision Trees

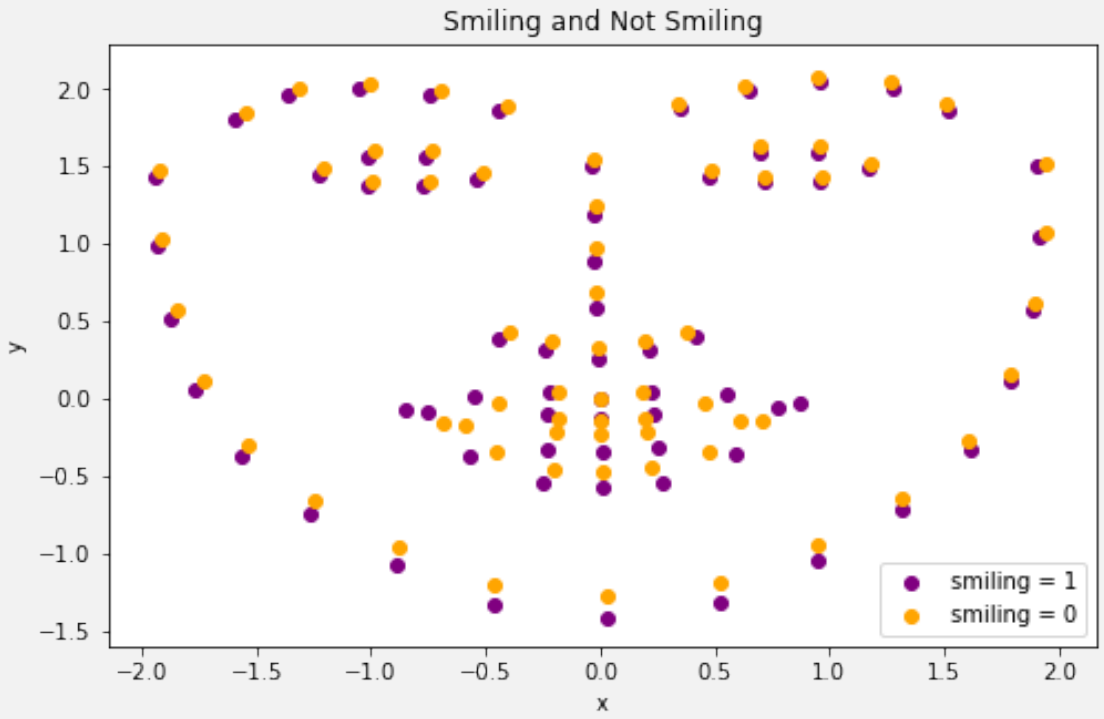
In this question we will train a classifier to predict if a person is smiling or not.

(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.

From what is displayed after loading the training and test datasets, it's observed that the train split has a size of [4800 rows x 137 columns] and the test split has a size of [1200 rows x 137 columns], with the first 136 columns being the coordinates typed 'float64' that represent the location of points on the face and the last column being the binary class of type 'int64' that indicates smiling or not.

Question 3 continued on next page...

(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 points that form a smiling face are separated more widely in the middle around the mouth of the face, so this might be one reason why this face looks more likely a smiling face.

Question 3 continued on next page...

(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?

The DecisionTreeClassifier in sklearn uses "gini" as the default for classification. One advantage of using Gini compared to Entropy could be less computationally intensive, as Gini does not compute logarithm whereas Entropy does.

Question 3 continued on next page...

(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.

For the training data, if the maximum depth is increased, the accuracy will increase as well or at least not go down.
For the testing data, the decision tree model would overfit and occupy huge memory if the maximum depth is too high. On the contrary, the decision tree model would not be able to capture enough patterns and interactions if the maximum depth is too small.

Question 3 continued on next page...

(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.*

(Accuracy rounded to two decimal places):
Maximum depth = 2: Training Accuracy: 79.48%, Test Accuracy: 78.17%
Maximum depth = 8: Training Accuracy: 93.35%, Test Accuracy: 84.08%
Maximum depth = 20: Training Accuracy: 100.00%, Test Accuracy: 81.58%

From my point of view, I would reckon that the model with a maximum depth equal to 8 to be the best model. For the training accuracy of each of the three trees, it increases as the relative maximum depth increases. On the contrary, the test accuracy of each of the three trees firstly increases to the accuracy referred to the maximum depth equal to 8 and then decreases at the end. In general, the deeper for the tree to grow, the more complex the model will become because there will be more splits which may cause overfitting in decision trees and the reduction of accuracy. In this case, the test model is likely to overfit after the maximum depth equal to 8 and there is a reduction in accuracy. Thus, this model would be the best.

Question 3 continued on next page...

(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`.

The top three most importance features are [0.33040484928175073, 0.08995881431599788, 0.08831447209293988] with respective names [x50, y48, y29]. The conclusion of this question will be positive. As the name with the highest importance, x50, has the largest mean value and the smallest standard deviation across the class in comparison with other attributes. Thus, it has different performance in Gaussian distribution for class 0 and class 1 respectively.

Question 3 continued on next page...

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

Use of 2D point locations is not sufficient enough to classify a smiling face. The points are only distributed on one plane, so the side face, for example, can not be classified properly. Besides, the size and orientation of the object on a plot would affect the result of predictions. More specifically, the face observed by a person who stands 100 metres from the object will be unidentical from the face observed by a person who stands 10 metres from to object.

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?

Classfication accuracy for alg_1: 61.60%, Classfication accuracy for alg_2: 55.00%, Classfication accuracy for alg_3: 32.10%, Classfication accuracy for alg_4: 32.90%,

From the reported classification accuracy for each of the four models, the best model can be observed to be alg_1 with accuracy 61.60%.

However, in my opinion, the computed accuracy values for the four models do not mean anything since the class is not balanced. Specifically, the number of 0s is more than that of 1s in the ground truth class. As a result, I would suggest setting the threshold a bit higher than the current value and hence observing which model has the highest accuracy.

Besides, a disadvantage of computing accuracy by converting continuous outputs into binary predictions could be that this approach relies highly on the presentation of data. Due to the outputs being converted to discrete, this approach can only predict a categorical outcome.

Question 4 continued on next page...

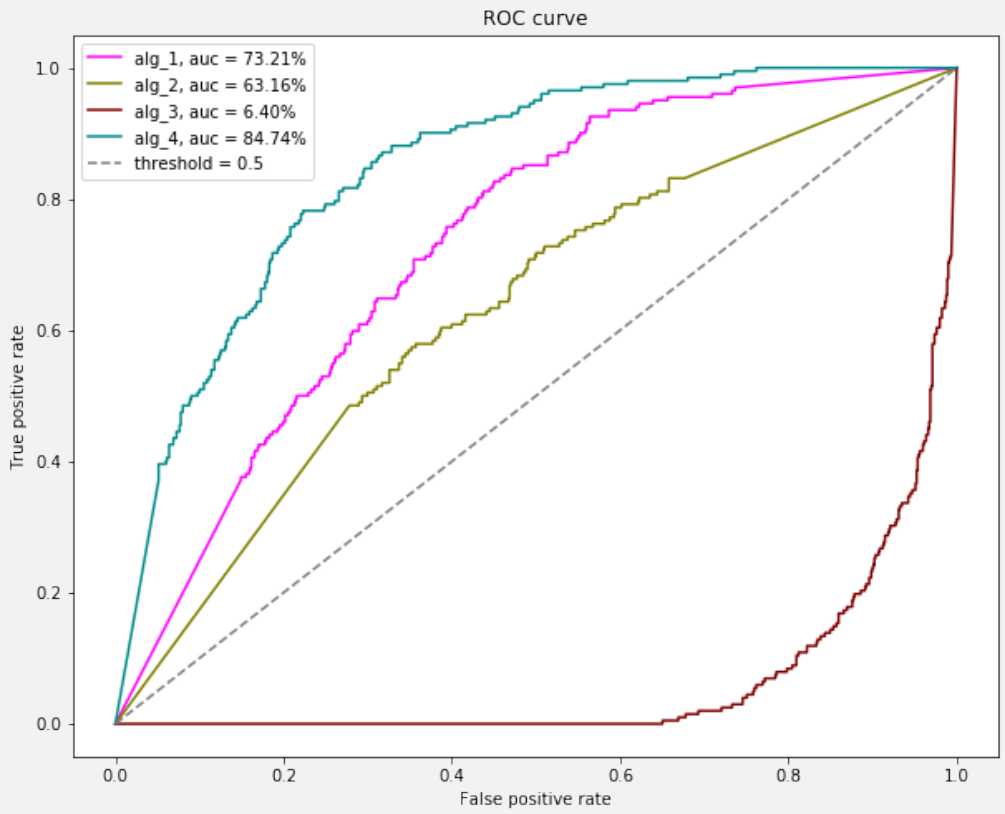
(b) (4 points) Instead of using classification accuracy, report the Area Under the ROC Curve (AUC) for each model. Does the model with the best AUC also have the best accuracy? If not, why not?
Hint: You can use the roc_auc_score function from sklearn.

AUC for alg_1: 73.21%, AUC for alg_2: 63.16%,
AUC for alg_3: 6.40%, AUC for alg_4: 84.74%

No, the model with the best AUC is alg_4 with AUC to be 84.74%. However, it does not have the best accuracy though. One reason could be that the dataset is unbalanced and overfitting. Besides, different ways of computing two outputs could be an alternative. In details, there is a threshold set to be 0.5 in the computation of the classification accuracy, while no threshold is involved in calculating AUC.

Question 4 continued on next page...

(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.



From the generated plot, the line that refers to alg_3 does not seem to perform as expected. The model has a low AUC of 6.4%, which means the model is approximately predicting the negative class as the positive class. One approach to overcome this issue would be flipping class 1 with class 0 for the predictions in alg_3. In addition, increasing the number of predictions in the class might be able to reduce bias, which then would probably enhance the performance.