IAML – INFR10069 (LEVEL 10): Assignment #1

s1817972

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 data is (50,2) that means there are 50 rows and 2 columns.
The mean of exam_score is 20.93 and revison_time is 13.99.
The range of revison_time is from 2.72(minimum) to 48.01(maximum),the range of exam_score is from 14.73(min) to 94.95(maximum). The datatype of data is float64.

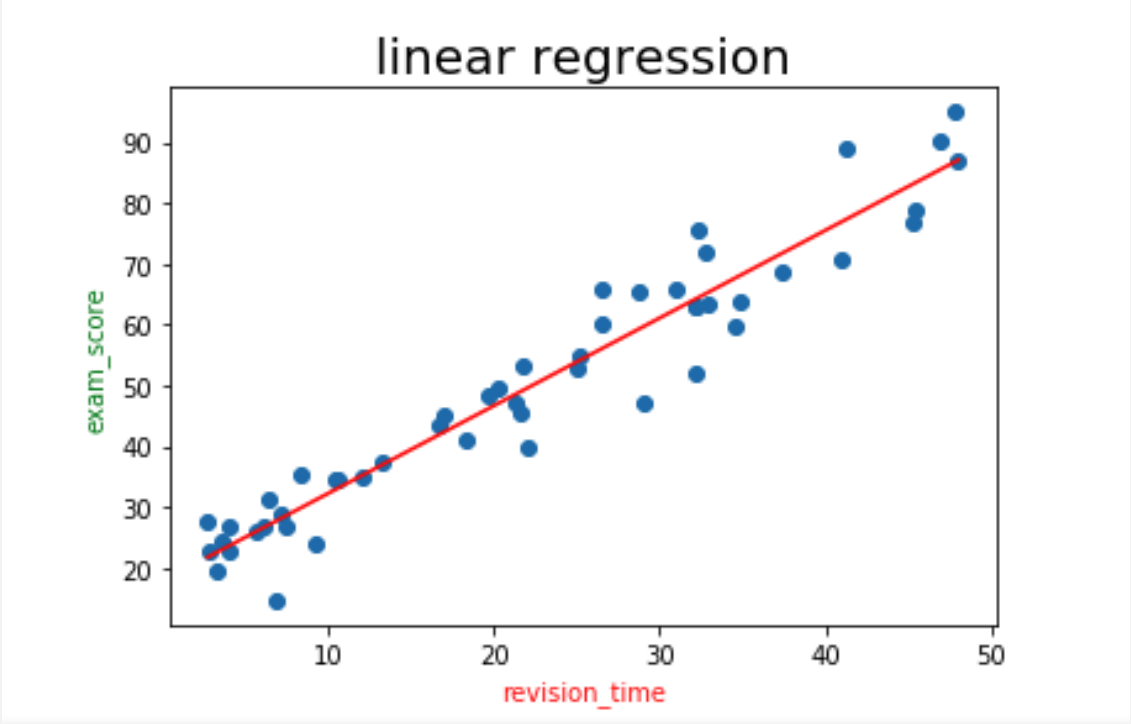
Question 1 continued on next page...

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

my estimated model parameter W is [17.89768026 , 1.44114091] in this liner regression yi = (xi)W, W should be a row vector. W[0] is bias(intercept of linear progression) and W[1] is the coefficient of the liner regression.

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.*

```
regression_part1.insert(0,'pre_pend',1.00)
data_train1=np.array(regression_part1[['pre_pend','revision_time']])
.reshape(regression_part1[['pre_pend','revision_time']].shape[0],2);
data_test=regression_part1['exam_score']
w=inv(data_train1.T.dot(data_train1)).dot(data_train1.T).dot(data_test)
```

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

$$MSE = (1/n) * \sum_{i=1}^{i=n} (y - \hat{y})^2$$

MSE is prone to outliers as it uses the same concept of using mean in computing each error value.Mean will change a lot by some significant outliers, then MSE will also change a lot.

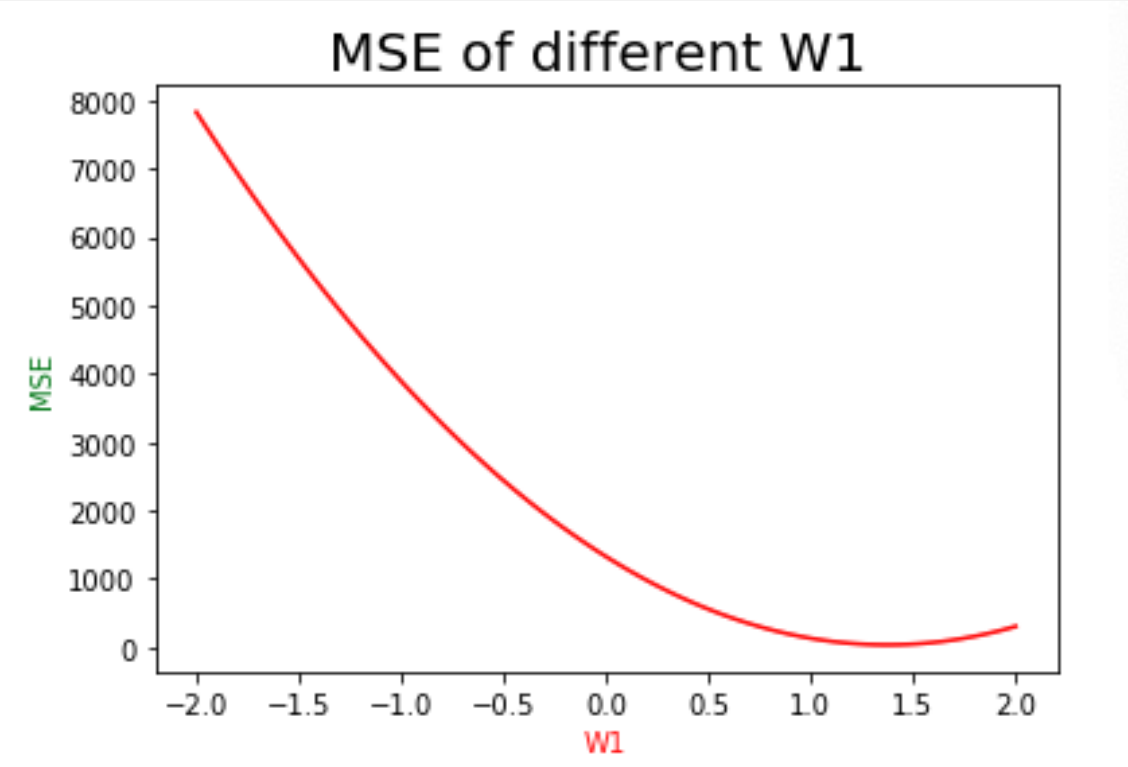
Question 1 continued on next page...

(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 for linear model fitted using sklearn = 30.985472614541290426;
MSE for linear model from closed-form solution = 30.985472614541301084;
In general, the MSE of two method is almost same. However after 13th significant figures, they are different.

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)`.

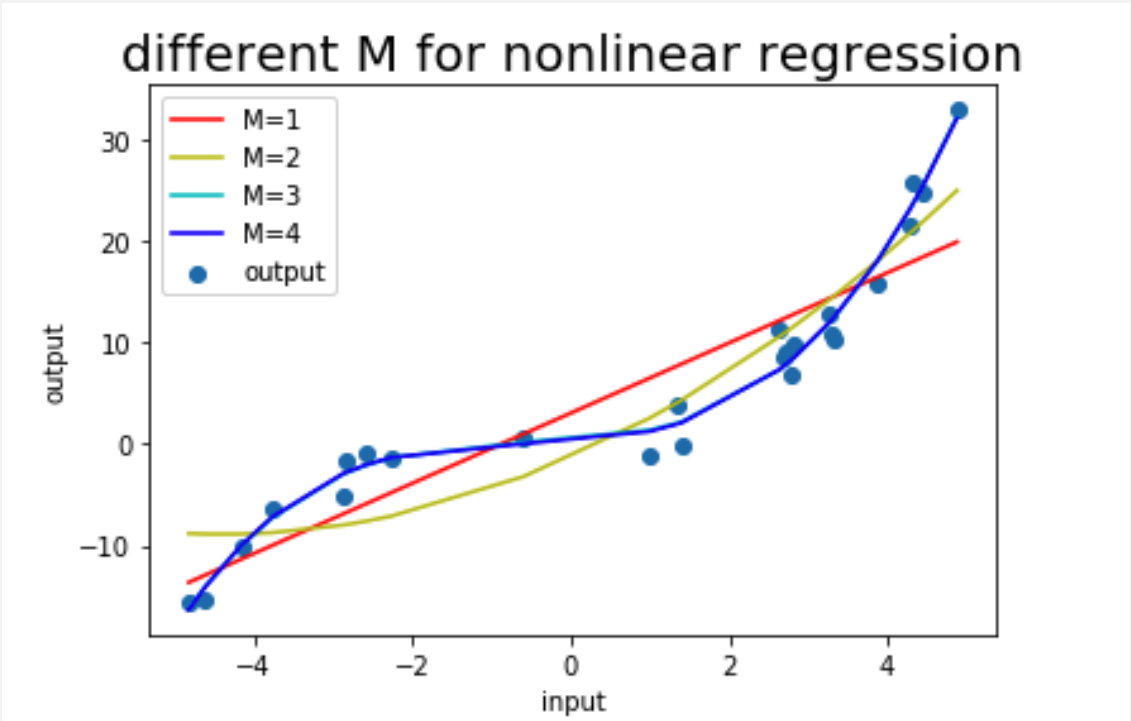


This is a curve line. when w1 equal to 1.35, the MSE is minimum,which is equal to 32.48. After minimum point, the MSE will increase with w1 increase. So this value is expected ,as it it global minimum.

Question 2 : (18 total points) Nonlinear Regression

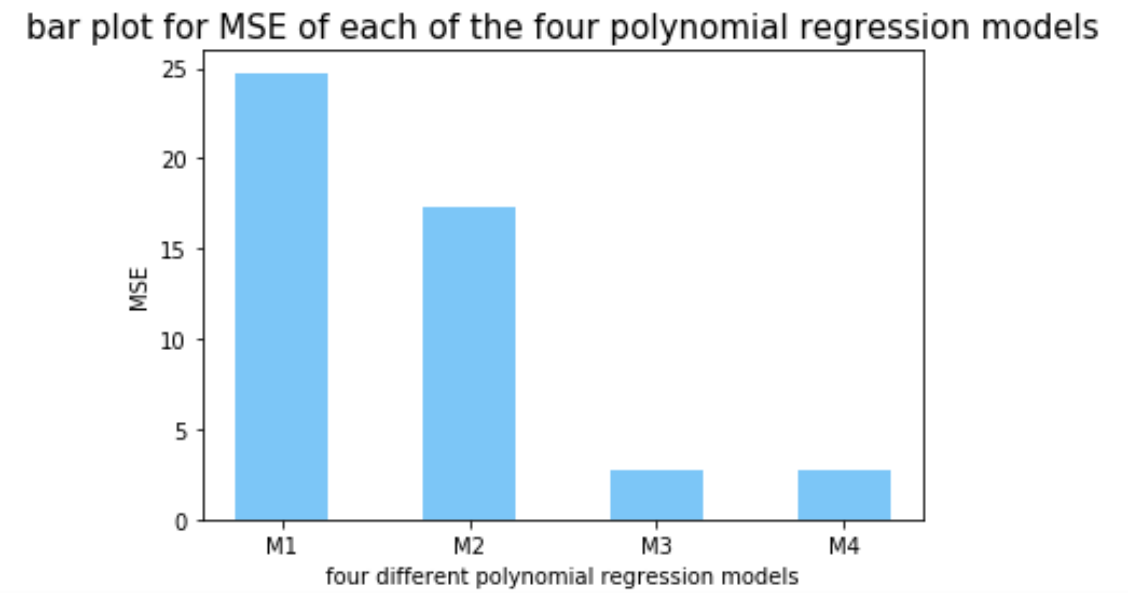
In this question we will tackle regression using basis functions.

(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.



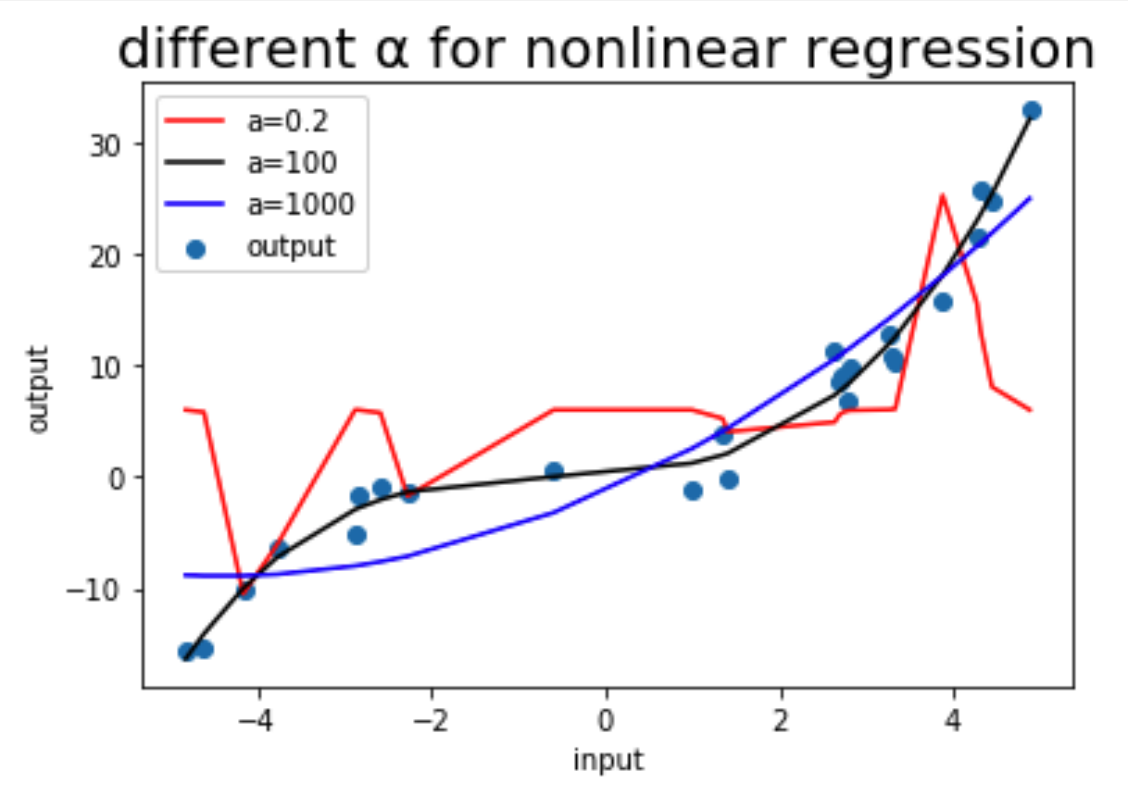
Question 2 continued on next page...

(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 line of regression model M3 is overlapped with the regression model of M4, when we look the fit plot graph.However, the mean squared error of m3 is greater than mean squared error of m4, m3 is fit enough for training data. so i think m3 is better.

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.



the larger value of alpha, this model will more underfitting.
the smaller value of alpha, this model will more overfitting.
Both overfit and underfit is not good.

Question 3 : (26 total points) Decision Trees

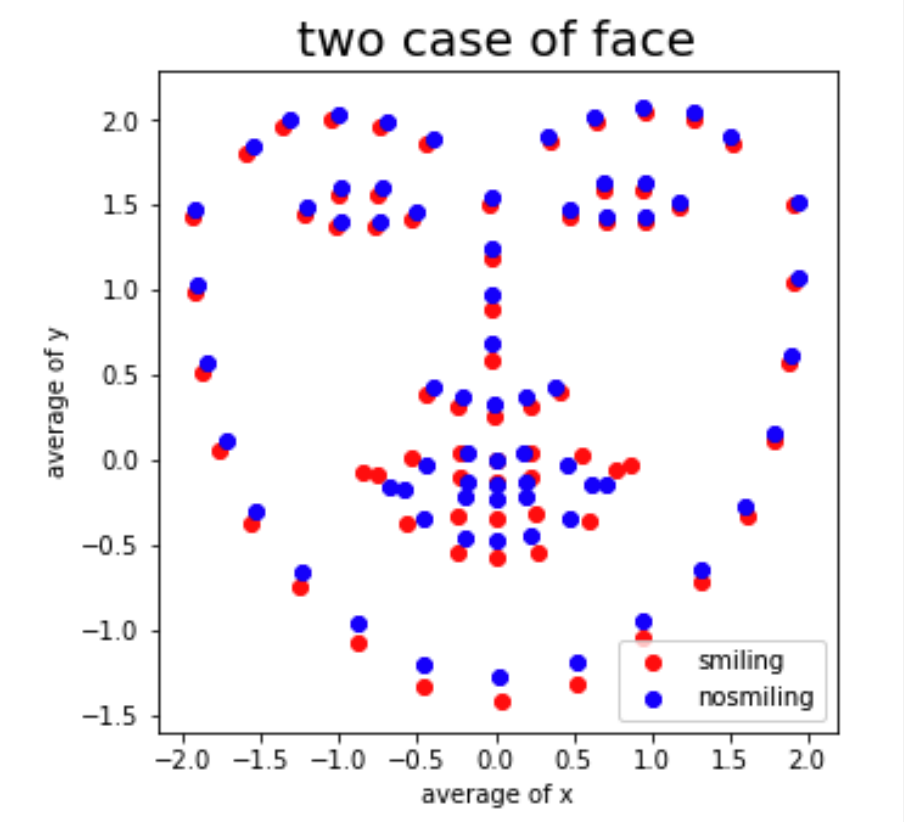
In this question we will train a classifier to predict if a person is smiling or not.

(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.

For training data, the data size is 4800 *137. First 136 column is training data with datatype(float64) and last column is label vector smiling with datatype(int)
For testing data, the data size is 1200*137,First 136 column is testing data with datatype(float64) and last column is label vector smiling with datatype(int)

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.



Compare two plot, we could find that the point of mouse will plot more widely when face smiling. If the face is no smiling, the point of corners of the mouth will distribute closely.

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 use "gini" as default classification. The advantage of "gini" is that "gini" could be less computationlly intensive compared to Entropy classification, as Gini does not compute logarothm.

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.

the larger value of maximum depth of the tree. There are more splits and it captures more information about the data and this is one of the root causes of overfitting in decision trees.Another problem is that outliers in the training data is picked up and learned as concepts by the model,these concepts do not apply to new data.Besides, the time complexity and space complexity will increase.
the smaller value of the parameter, it will cause underfitting. this model will be too simple to explain the variance.

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.*

trainning Accuracy for max depth=2: 79.5 %
test Accuracy for max depth=2: 78.2%
trainning Accuracy for max depth=8: 93.4%
test Accuracy for max depth=8: 84.1%
trainning Accuracy for max depth=20: 100.0%
test Accuracy for max depth=20: 81.6%
i think the model with max depth 8 is best.
For classification with max depth 2, it is underfitting, the accuracy of training data is 79.5 % and testing data is 78.2 %. Those value are not high enough to fit the model.
For classification with max depth 20, it is overfitting, the accuracy of training data is 100.0 % which means this model focus on training data too much, so the accuracy of test only be 81.6%. so the classifiers with max depth of 8 is fit well.

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 importane attribute is x50 y48 and y29. The highest important attribute is x50, it is make sense, because the mean difference of x50 in class 0 and class 1 is largest and the std of x50 is smallest compare with all point's feature.Another reason is that x50,y48 and y29 is located at face mouse and face noise, which is important to show face smiling or not smiling. So x50 make sense.

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.

The limitation is that 2d point location is not good enough to classifier a face model,for example, the side face also will affect the face smiling or not. Another problem is that different pace have different scale.

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?

trainning Accuracy of alg1 : 61.6%
trainning Accuracy of alg2 : 55.0%
trainning Accuracy of alg3 : 32.1%
trainning Accuracy of alg4 : 32.9%
the first model with alg1 is the best one, as the accuracy of this model is highest. The limitation of this model is that we set threshold(0.5) to compute accuracy, in order to convert continuous outputs into binary predictions. Now the output is discrete, so method only predict a categorical outcome. For this problem, one way to solve is that we can change the threshold or we can normalise the data to calculate accuracy.

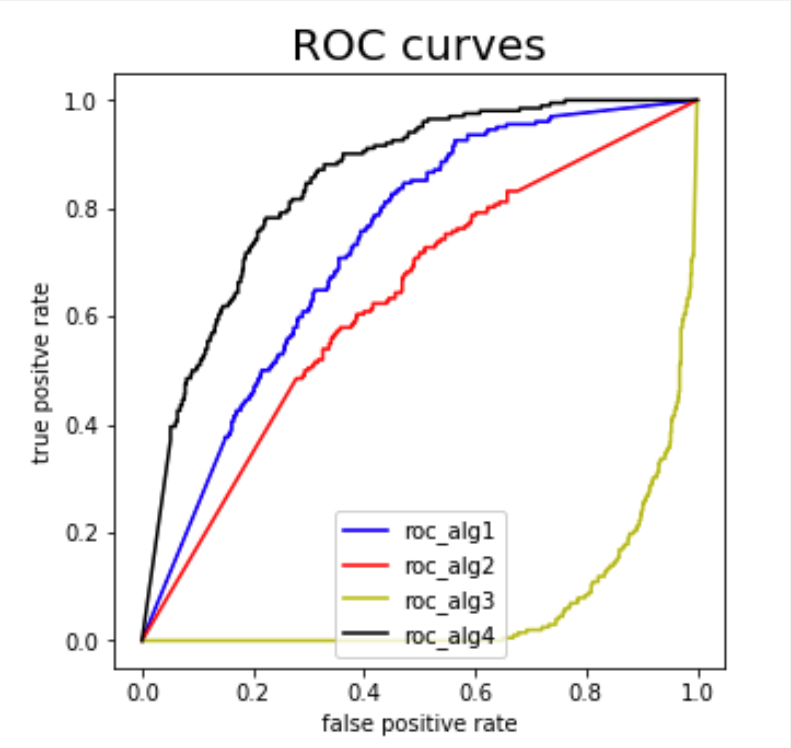
Question 4 continued on next page...

(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-alg1 : 0.732
auc-alg2 : 0.632
auc-alg3 : 0.064
auc-alg4 : 0.847
No, alg4 have the best AUC, but the accuracy of alg1 is largest. The reason could be that the dataset is unbalanced and overfitting. Besides, we set threshold = 0.5 when we calculate the accuracy also could lead to different performance.

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.



In this plot, we can see that the curve of alg3 does not seem to perform expected. The model of alg3 has a low AUC of 6.4,which means this model is predicting class 0 to class 1 and class1 to class0. For this issue, i think we should change each label of alg3 from 0 to 1 and from 1 to 0.