IAML – INFR10069 (LEVEL 10): Assignment #1

s1862671

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 has 50 rows and 2 columns. The name of two columns are *revision_time* and *exam_score*. The data range of *revision_time* is from 2.72(min) to 48.01(max). The data range of *exam_score* is from 14.73(min) to 94.95(max). The mean of *revision_time* is 22.22. The mean of *exam_score* is 49.92. The standard deviation of *revision_time* is 13.99. The standard deviation of *exam_score* is 20.93. The data type of both columns is float64. All float values are rounded to 2 decimal places.

Question 1 continued on next page...

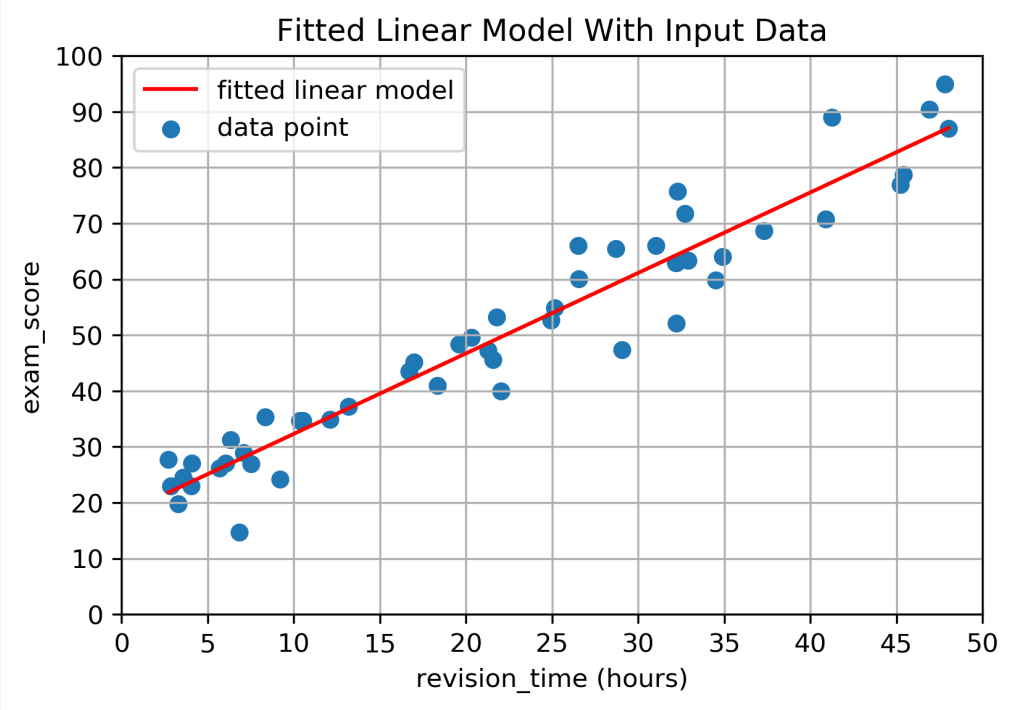
(b) (3 points) Fit a linear model to the data so that we can predict `exam_score` from `revision_time`. Report the estimated model parameters $\mathbf{w}$. Describe what the parameters represent for this 1D data. For this part, you should use the sklearn implementation of Linear Regression.
Hint: By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

w = [17.90,1.44] (rounded to 2 decimal places)
17.90 is the intercept of linear function for linear regression.
1.44 is the coefficient of linear function for linear regression.

Question 1 continued on next page...

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

This image shows fitted linear model with input data.



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.*

"bias" is a column with all values 1.
"dataset1" is the dataframe read by "regression_part1.csv".

```
X = dataset1[["bias","revision_time"]].values
y = dataset1["exam_score"].values
w = np.linalg.inv(X.T.dot(X)).dot(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 $\hat{y}$ (`$\hat{y}$` in latex) in place of the model prediction.

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

Limitation: MSE is prone to outliers. MSE calculates mean of sum of squared errors. Mean is prone to outliers and square intenses the effect of outliers, so the MSE is also prone 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.

Results are presented in the table below. (MSE rounded to 2 decimal places)

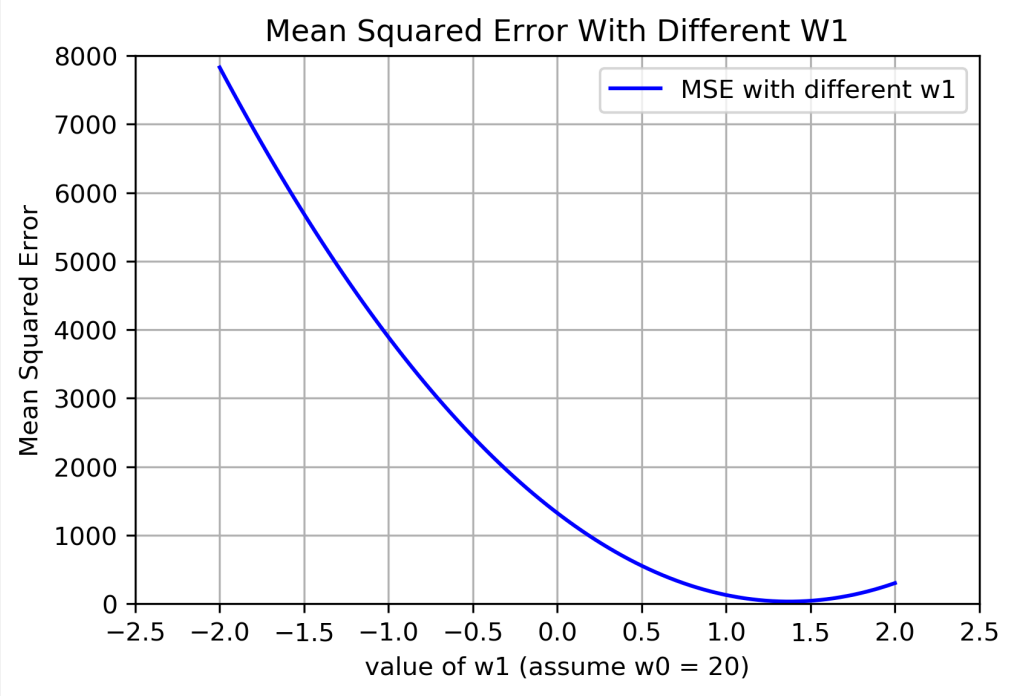
Model Name	Mean Squared Error
sklearn linear model	30.99
closed form solution	30.99

The MSE of two models are roughly the same except there is a technical problem for numpy operation to get a slightly different answer after many decimal places.

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 image shows Mean Squared Error with different w1.



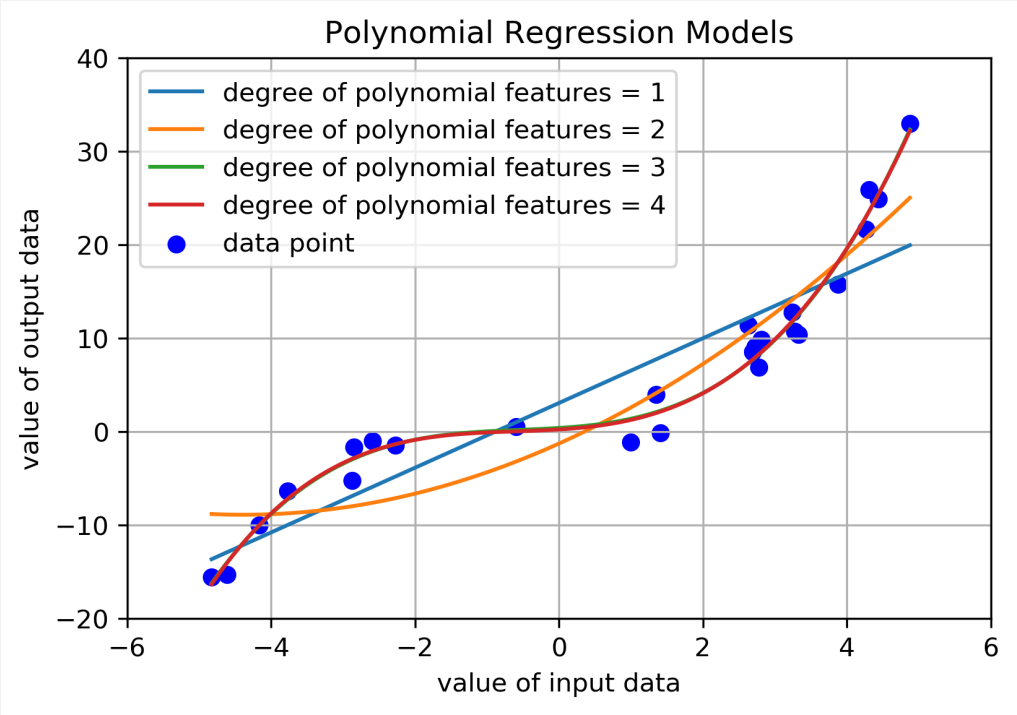
When value of w1 increases from -2.0 to 2.0, the value of Mean Squared Error firstly goes down and then goes up. The minimum MSE is around 32.48 which occurs when w1 is around 1.35. This value should be expected. The reason is that the model is a 2D linear function, the function MSE against w1 and w0 is concave, so there is only one minimum which is globally optimal, therefore 1.35 should be expected.

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`.

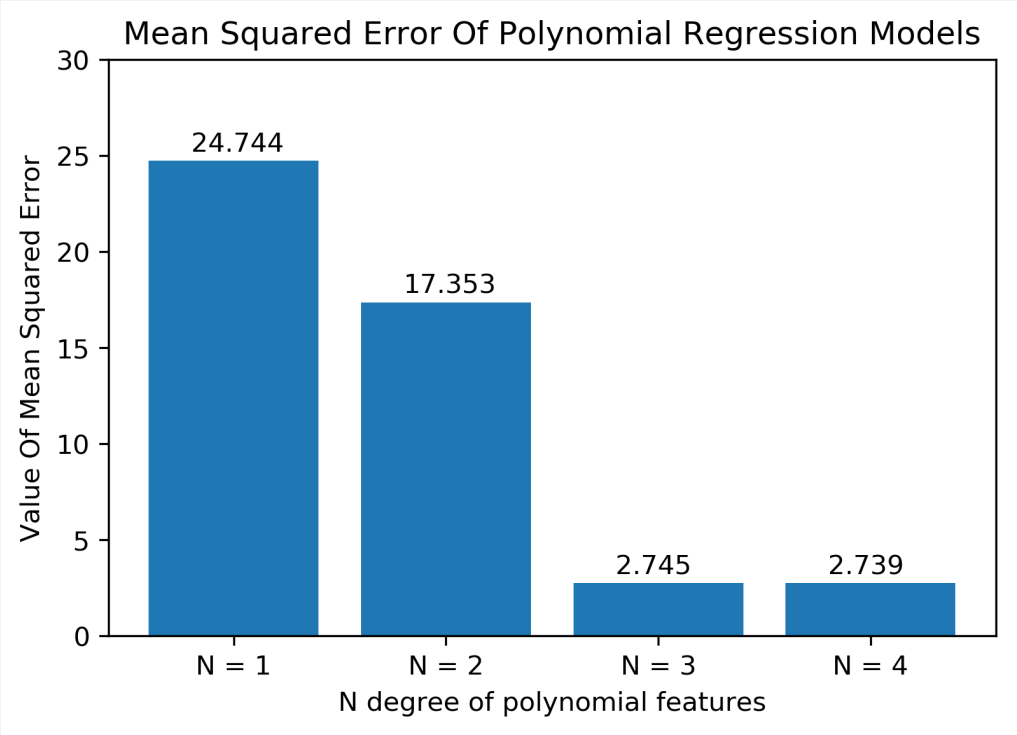
This image shows polynomial regression models with different degrees of polynomial features. Note that the third and fourth line on the plot overlap with each others.



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.

This image shows a bar plot for Mean Squared Error of each of the four different polynomial regression models. (MSE rounded to 3 decimal places)



Question 2 continued on next page...

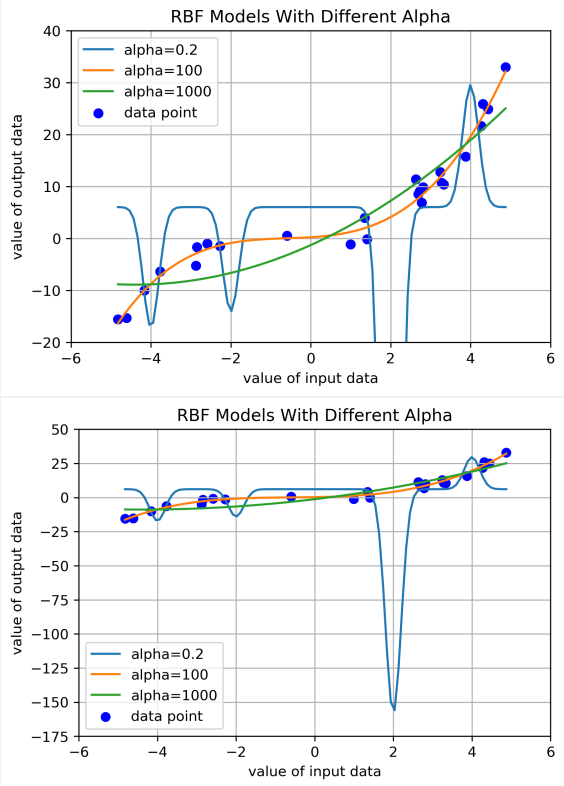
(c) (4 points) Comment on the fit and Mean Squared Error values of the $M = 3$ and $M = 4$ polynomial regression models. Do they result in the same or different performance? Based on these results, which model would you choose?

The model with M=3 has Mean Squared Error 2.745. The model with M=4 has Mean Squared Error 2.739. Although the model with M=4 has a lower Mean Squared Error than the model with M=3, the difference between Mean Squared Error of the model with M=3 and model with M=4 is much smaller than the difference between Mean Squared Error of the model with M=2 and model with M=3. This means that there is little improvement to add the x^4 to train a model and the model with M=4 may be overfitting. Therefore I will choose the model with M=3 which is less prone to be overfitting. (MSE rounded to 3 decimal places)

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.

This image shows RBF Models with different alpha values.



When alpha becomes smaller, the RBF model will be more and more underfitting. When alpha becomes larger (when it is bigger than a specific value), the RBF model also can not fit the data well. The RBF model can only fit data well when alpha is in the middle range of some values like 100.

Question 3 : (26 total points) Decision Trees

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

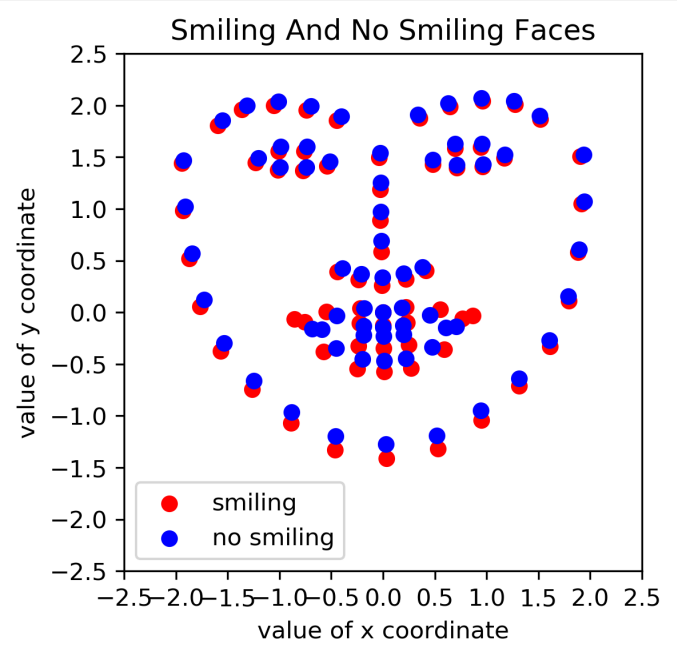
(a) (4 points) Load the data, taking care to separate the target binary class label we want to predict, `smiling`, from the input attributes. Summarise the main properties of both the training and test splits.

Data in the training split has 4800 rows and 137 columns. Data in the test split has 1200 rows and 137 columns. They both have 136 columns for features and 1 column for labels. They both have data type of float64 for features and data type of int64 for labels. There are 2465 rows in training split with label smiling=0 and there are 2335 rows with label smiling=1. There are 608 rows in the test split with label smiling=0 and 592 rows with label smiling=1.

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.

This image shows smiling and no smiling face images.



The corners of mouth in average smiling face is upper than corners in average no smiling face. The size of mouth in average smiling face is also bigger that the size of mouth in average no smiling face. The mouth is where model can recognize whether the face image is smiling or not. Other places are mostly the same for two plots.

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 defult measure of purity in sklearn DecisionTreeClassifier is Gini (Gini impurity). One advantage of Gini is that it is not required to compute logarithmic functions which are computationally intensive but entropy does. In addition, Gini can also find closed form solution.

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.

When maximum depth of tree is larger, it will make tree more complex since it has more splits. When maximum depth of tree is smaller, it will make tree simpler since it has less splits. One potential problem for small maximum depth is that the model has more chance to be underfitting since the leaves of the tree may not be pure. Two potential problems for large maximum depth are that the model has more chance to be overfitting and be more sensitive to the outliers in the dataset (tree will fit outliers in it).

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.*

Results are presented in the table below. (rounded to 2 decimal places)

Max Depth	Training Accuracy	Test Accuracy
2	79.48%	78.17%
8	93.35%	84.08%
20	100.00%	81.58%

The second model is the best because it has the highest test accuracy and a reasonably high training accuracy among three models. In addition, the first model is underfitting because its both training and test accuracy are really low (high bias) and the third model is overfitting because the difference between training and test accuracy is really large (high variance). The second model fits the data best because it has a good balance between variance and bias.

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 names of the top three most important attributes are x50, y48, y29.
x50 makes sense. The reason is that x50 has the largest information gain in the first split of decision tree and has a significant contribution in construction of decision tree. The Gaussian distribution of x50 in smiling samples and no smiling samples are slightly different. They both have similar standard deviation but clearly different mean values. In addition, x50 also has the largest negative correlation coefficient with labels among the attributes.

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.

Yes, one limitation is that it is difficult for the machine learning model such as decision tree classifier to fit the orientation or rotation information provided by 2D point locations. The reason is that the decision tree can only fit the data by cutting the 2d space vertically or horizontally.

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?

Results are presented in the table below. (rounded to 1 decimal places)

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

The alg_1 model is the best. The limitation is that accuracy is a poor metric when ground truth labels of each class in the dataset are imbalanced (there are 202 class 1 and 798 class 0). The way to improve this method is to increase the threshold value in the metric or using sample weights to each class.

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.

Results are presented in the table below. (rounded to 2 decimal places)

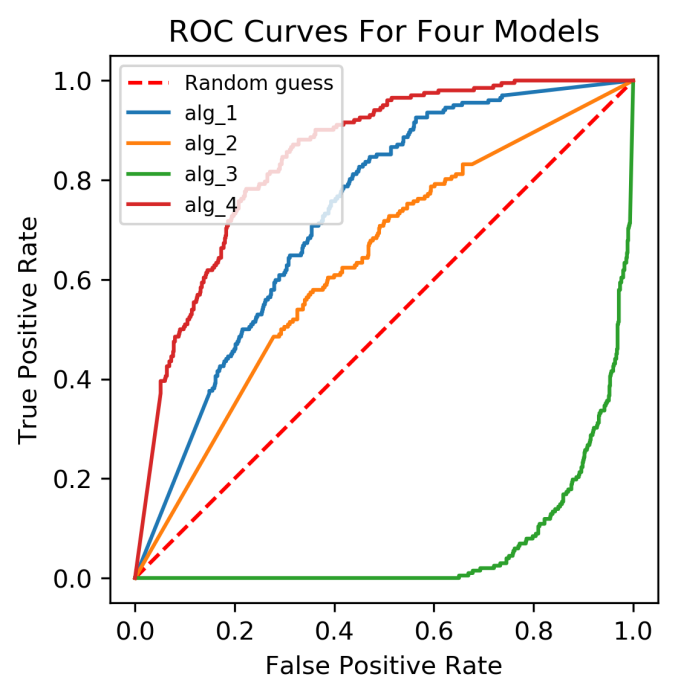
Model Name	AUC Score
alg_1	0.73
alg_2	0.63
alg_3	0.06
alg_4	0.85

The model with the best AUC score does not have the best accuracy. One reason is that the ground truth labels of each class in the dataset are imbalanced. Another reason is that accuracy is computed at the threshold value of 0.5 while AUC can be seen as an overall performance of all the accuracy for all threshold values.

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.

This image shows ROC curves for each of the four models.



alg_3 has a ROC curve that plots on the other side of Random guess line, which is much different from other models. It may use the opposite labels to train the model which let alg_3 predicts class 1 with low score and class 0 with high score. alg_3 can be improved by changing class 1 to class 0 and class 0 to class 1 as new results and then it will have the best performance among four models.