IAML – INFR10069 (LEVEL 10): Assignment #1

s1850642

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 data given is of size 50 rows by 2 columns, each row defines how many hours a student has worked and the corresponding mark on the exam from 0 to 100. It consists of 64 bit floating data types. Furthermore, we can assume that there is a linear relation, the more hours of studying means better performance. The ranges are:
Minimum: x = 2.723 hours , y = 14.731 score
Maximum: x = 48.011 hours, y = 94.945 score

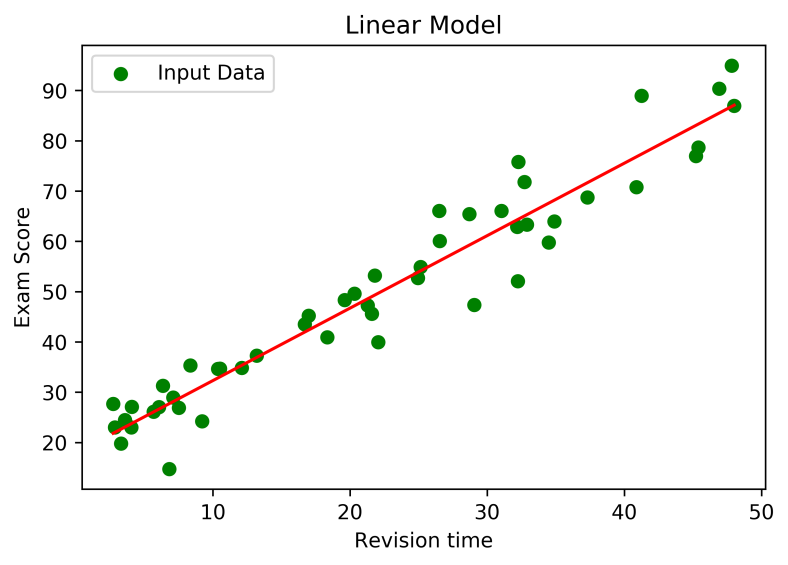
Question 1 continued on next page...

(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 equation that describes the fitted linear regression model is:
y = a*x + b
Substituting the values we obtained from our model, we get the following equation:
y = 1.44114091*x + 17.89768026
Where a is the regression coefficient and b is the intercept.

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 $\mathbf{w}$ *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.*

```
x_new = np.vstack((np.ones(len(x)), x)).T #pre-pend ones to get a 2-d array and get the transpose
beta = np.linalg.inv(x_new.T.dot(x_new)).dot(x_new.T).dot(y) #getting the model parameters of our linear model
y_predic = x_new.dot(beta)
```

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.

The Mean Squared Error expression is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

It is very sensitive to outliers

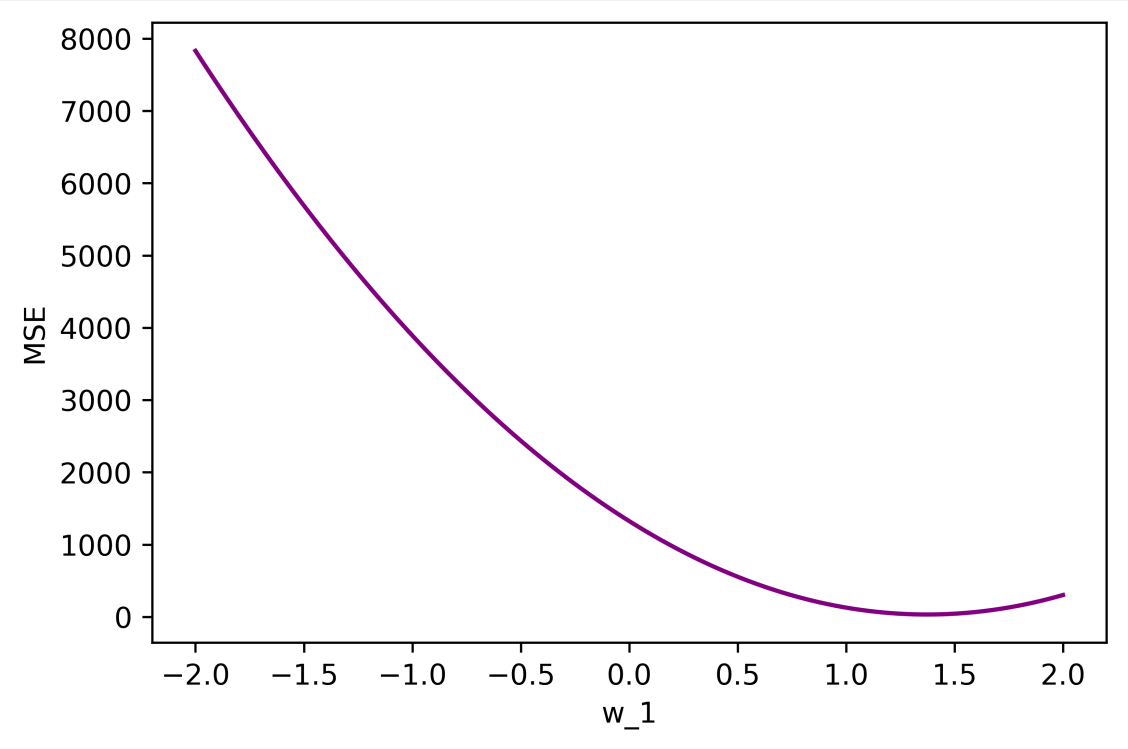
Question 1 continued on next page...

(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.

The Mean Squared Error for the closed-form solution is: 30.98547261454129
The Mean Squared Error for the sklearn solution is: 30.985472614541305
There is no noticeable difference between the values.

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)`.

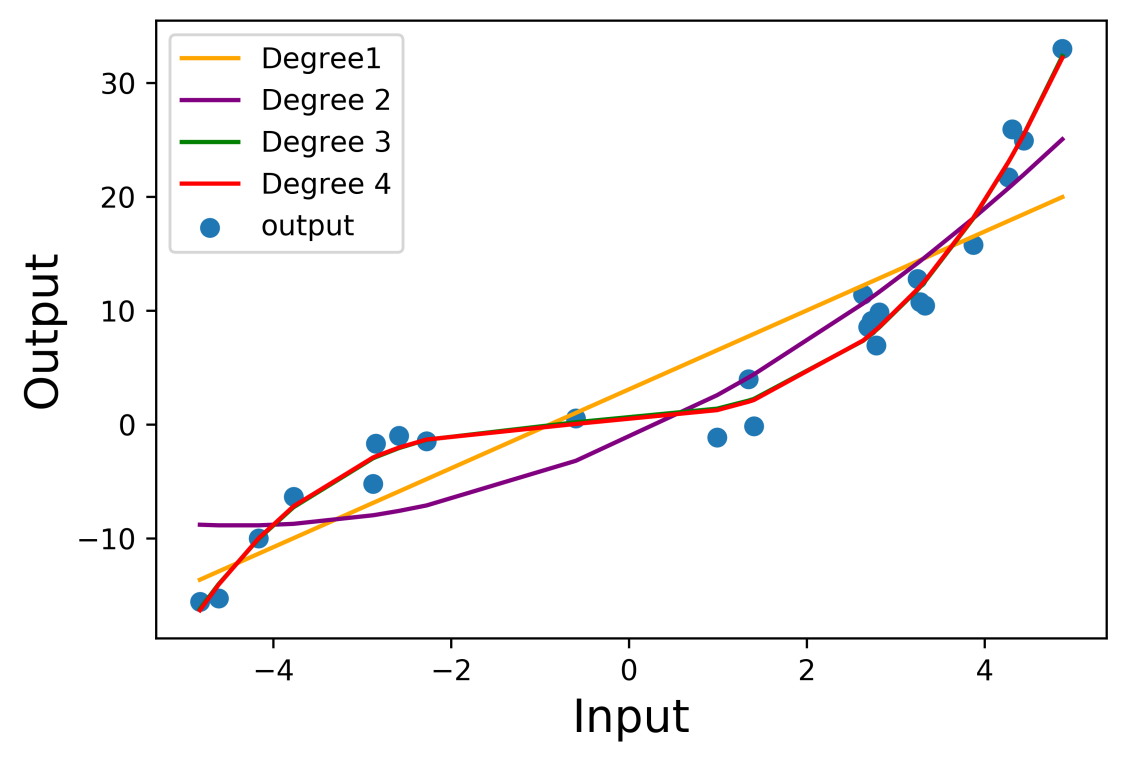


In the plot, the mean squared error decreases until the minimum value of w1 and then it starts increasing.
The Minimum value for w1 is '1.3535', it can be expected that the value will be similar to the one predicted from the model in the question above, also it is smaller to compensate for the change in the w0, being a bit bigger.

Question 2 : (18 total points) Nonlinear Regression

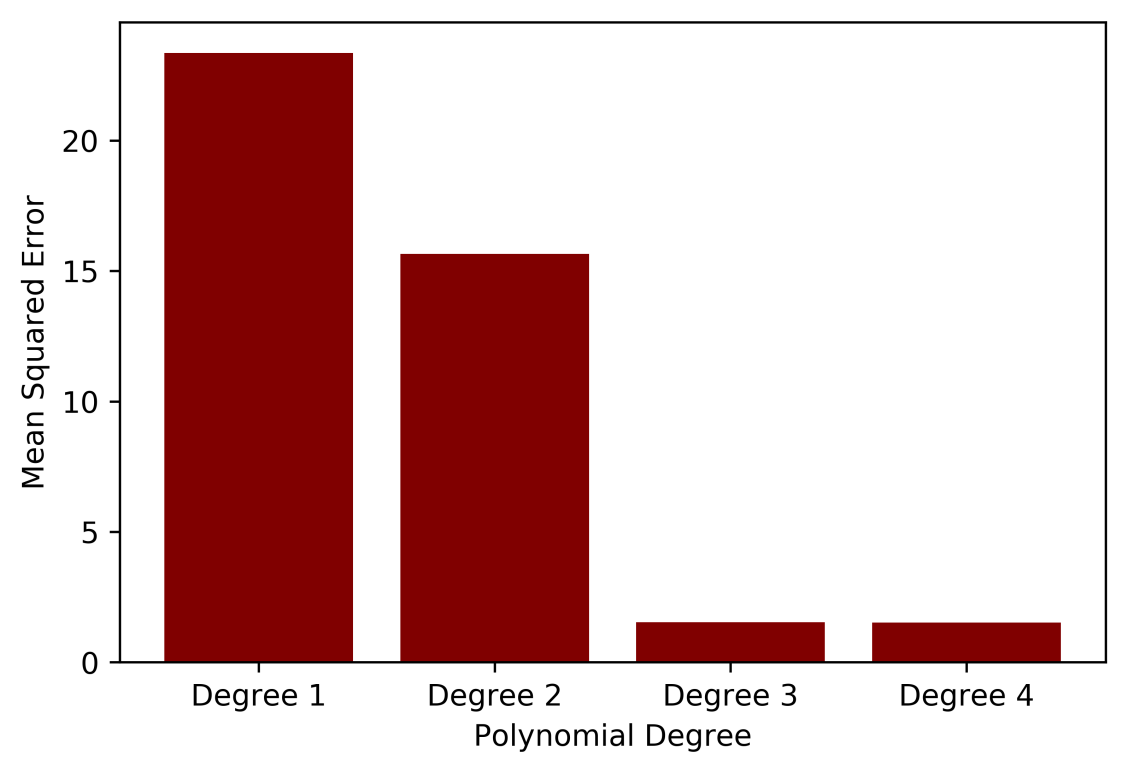
In this question we will tackle regression using basis functions.

(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.



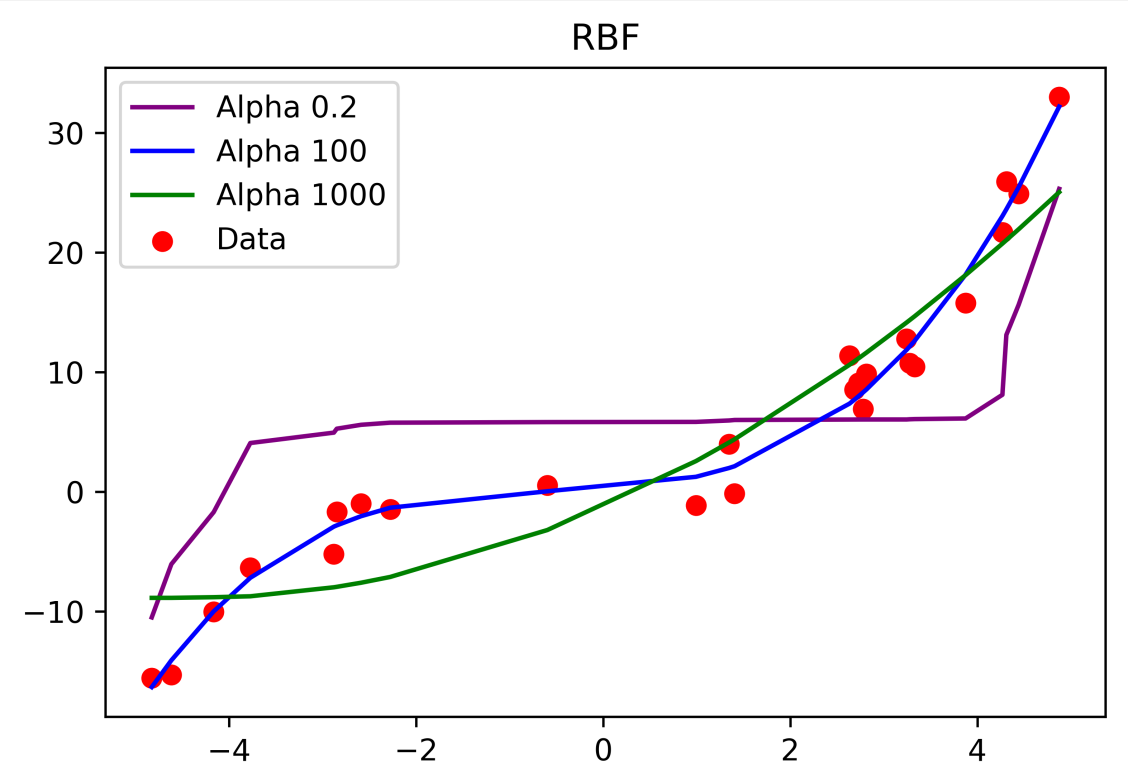
Question 2 continued on next page...

(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?

M3 = 1.5380270560714744
M4 = 1.5246859886011572
There is a small difference between both, a polynomial regression model with degree 4 seems to be a bit more accurate than the one with degree 3. Hence, I would choose the polynomial with M = 4.

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 is clear that with low values of α the nonlinear prediction performs poorly not fitting the date properly, while with high values the fitted prediction misrepresent the data.

Question 3 : (26 total points) Decision Trees

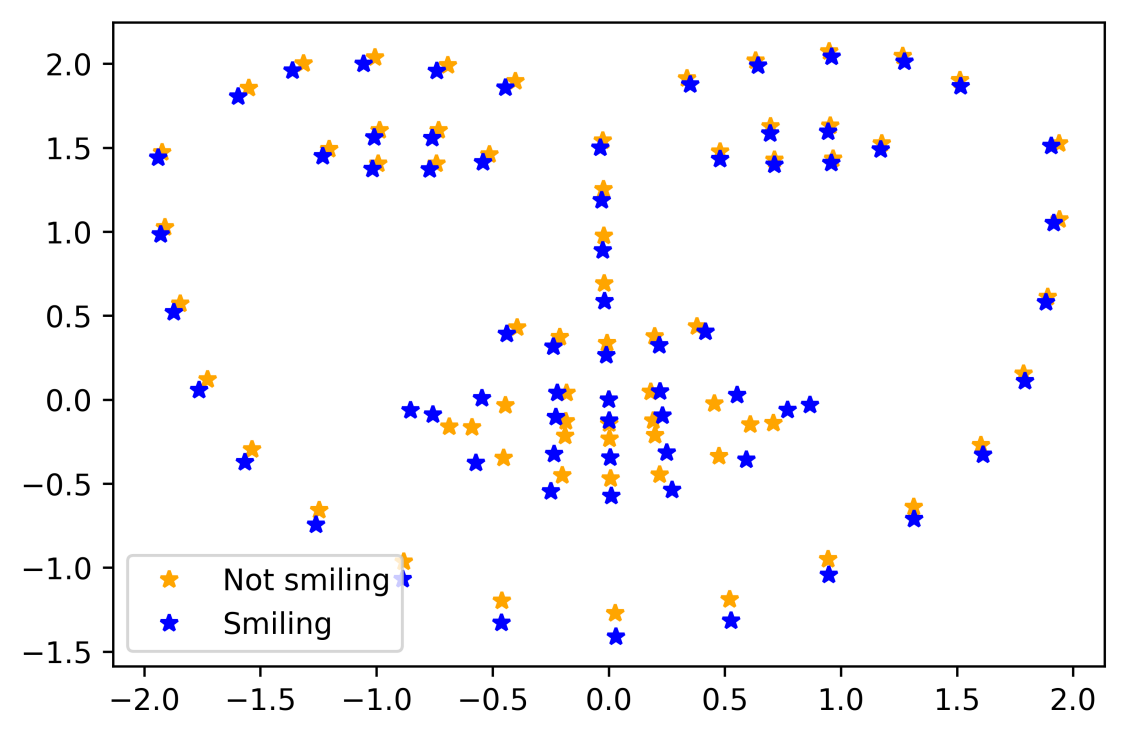
In this question we will train a classifier to predict if a person is smiling or not.

(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.

All instances are made of 68 pairs of x and y coordinates, and a last binary bit stating whether it is smiling (1) or not (0).
Training split has 4800 instances.
Test split has 1200 instances.

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.



When smiling the average points from the mouth are slightly wider on the x-axis, and the eyes seem to be slightly smaller on the y-axis

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?

It uses the Gini value.Which is calculated with the function:

$$Gini = 1 - \sum_{i=1}^{n} p_i^2$$

The main advantage, there is no logarithmic function involved, as opposed to Entropy, hence, it requires less computation.

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.

Usually the higher value of maximum depth of the tree causes over-fitting, and a lower value causes under-fitting

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.*

The values obtained are represented below

Depth	Train Accuracy	Test Accuracy	Maximum Depth
2	79.5%	78.2%	2
8	93.4%	84.1%	8
20	100%	81.6%	20

It can be observed that depth 20 has the highest accuracy in the training set but not in the testing set. Meanwhile, depth 8 has lower accuracy than depth 20 in the training set but higher in the testing set. Hence, the algorithm with depth 20 is over-fitting the data. The best model is the one with depth 8.

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`.

Attribute	Importance
x50	0.3304
x48	0.0899
y29	0.0883

In our case it does not make sense, we are comparing importance of each individual attribute. We would need pairs of attributes(x, y) for importance to be relevant.

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.

Your Answer Here

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	Accuracy
1	61.6%
2	55.0%
3	32.2%
4	32.9%

According to this metric, Model 1 would be the best fit. The main problem with accuracy is that it is misleading if we have unbalanced classes and can lead to the wrong classifier.

Question 4 continued on next page...

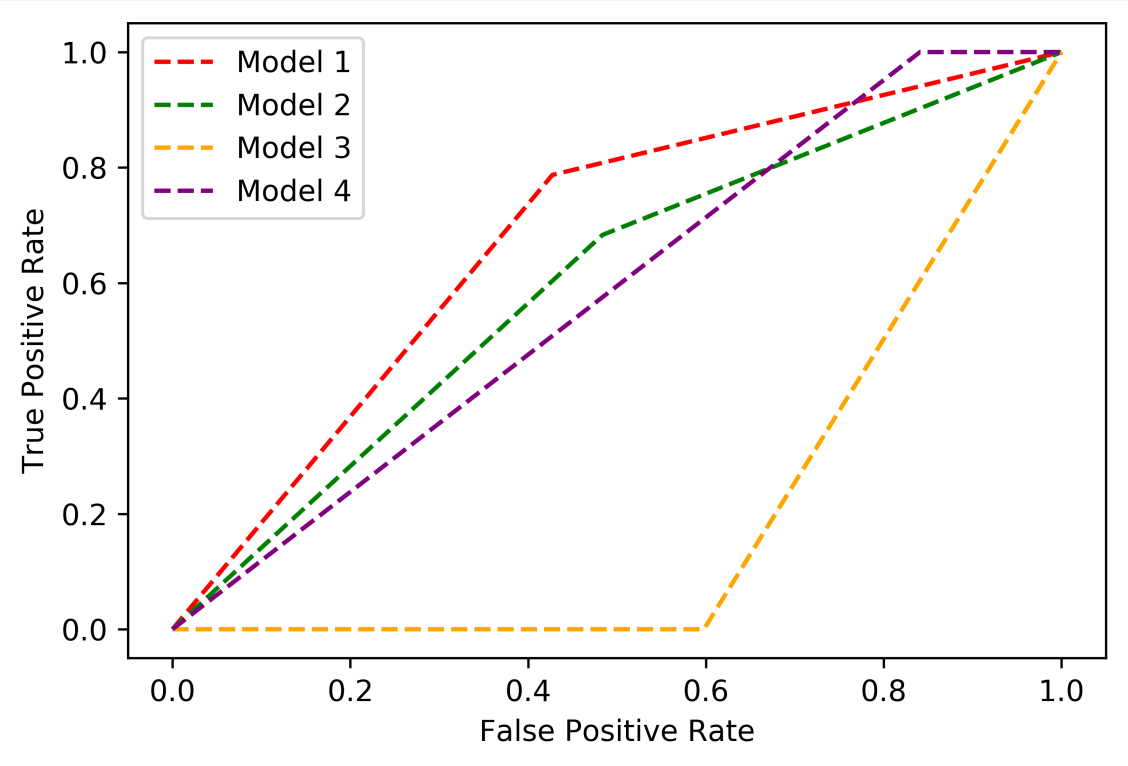
(b) (4 points) Instead of using classification accuracy, report the Area Under the ROC Curve (AUC) for each model. Does the model with the best AUC also have the best accuracy? If not, why not?
Hint: You can use the roc_auc_score function from sklearn.

Model	AUC
1	0.6799
2	0.5997
3	0.2011
4	0.5795

Yes, the Model 1 has the both best accuracy and 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.



The algorithm 3 performs poorly compared to the other models, recall that it had both the lowest accuracy and AUC values. A way to improve its AUC curve would be to minimize the false positives, even if that means an increase of false negatives. But since we can not re-train the model, increasing the threshold would work in this case.