IAML – INFR11182 (LEVEL 11): Assignment #1

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.

This dataset has 50 observations (rows) and 2 attributes (columns). The numerical characteristics of each attribute are shown as the table.

attribute	data type	min	max	mean	std
revision_time	numeric (float64)	2.723	48.011	22.220	13.986
exam_score	numeric (float64)	14.731	94.945	49.920	20.926

Question 1 continued on next page...

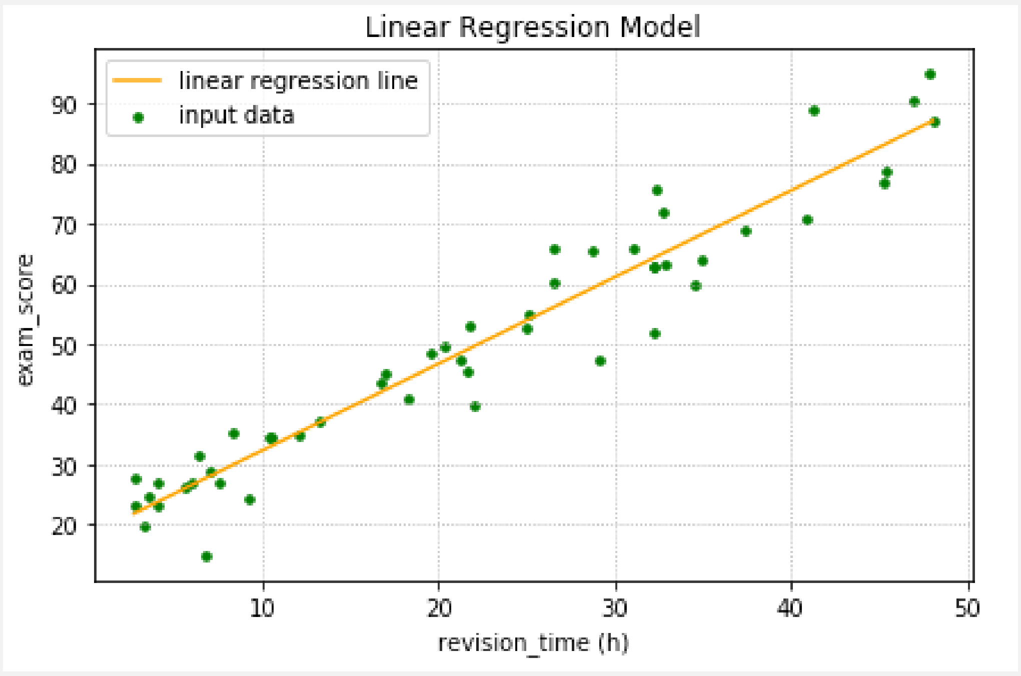
(b) (3 points) Fit a linear model to the data so that we can predict `exam_score` from `revision_time`. Report the estimated model parameters $\mathbf{w}$. Describe what the parameters represent for this 1D data. For this part, you should use the sklearn implementation of Linear Regression.
Hint: By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

$w = [17.898, 1.441]$
The first value ($w0 = 17.898$) represents the intercept of the fitted linear model.
The second value ($w1 = 1.441$) represents the slope of the fitted linear model.

Question 1 continued on next page…

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

The figure illustrates the fitted linear model and the input data for this dataset.



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.matrix((X.T).dot(X)).I).dot(X.T).dot(y_true)
```

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.*

Mean Squared Error (MSE):

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

Limitations: Sensitive to outliers. Outlier has a large squared error, it will make MSE larger.

Question 1 continued on next page...

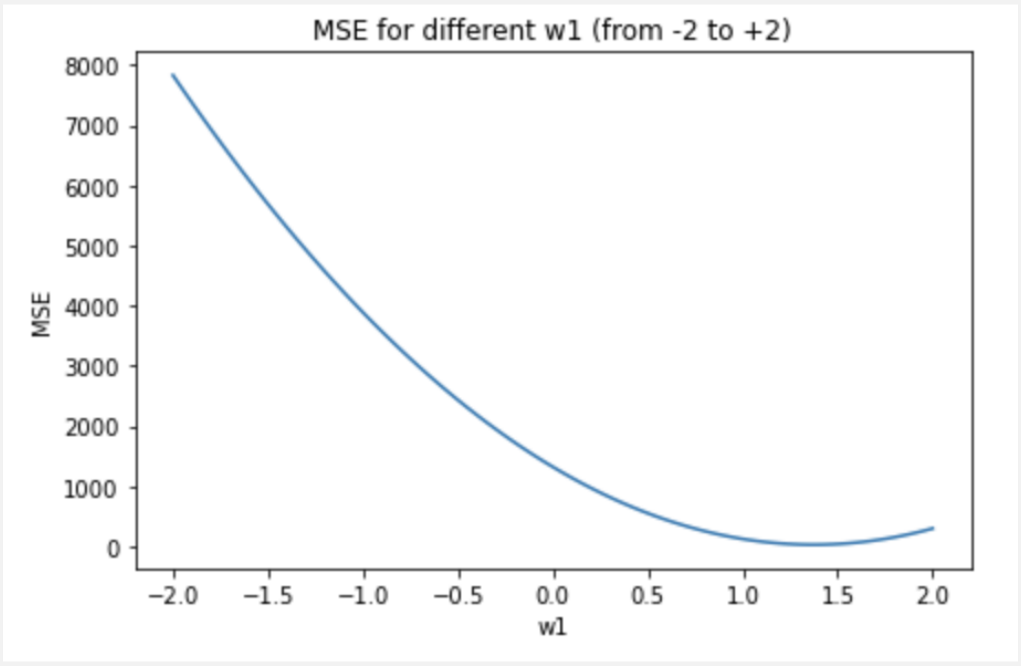
(f) (3 points) Our next step will be to evaluate the performance of the fitted models using Mean Squared Error (MSE). Report the MSE of the data in `regression_part1.csv` for your prediction of `exam_score`. You should report the MSE for the linear model fitted using sklearn and the model resulting from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

When using sklearn, MSE=30.9854726145413.
When using closed-form solution, MSE=30.98547261454129.
MSE values calculated by two methods are almost the same. The decimal precision of the result calculated using closed-form is a little bit higher.

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 figure indicates the Mean Squared Error (MSE) for w_1 changes from -2 to $+2$ ($w_0 = 20$).



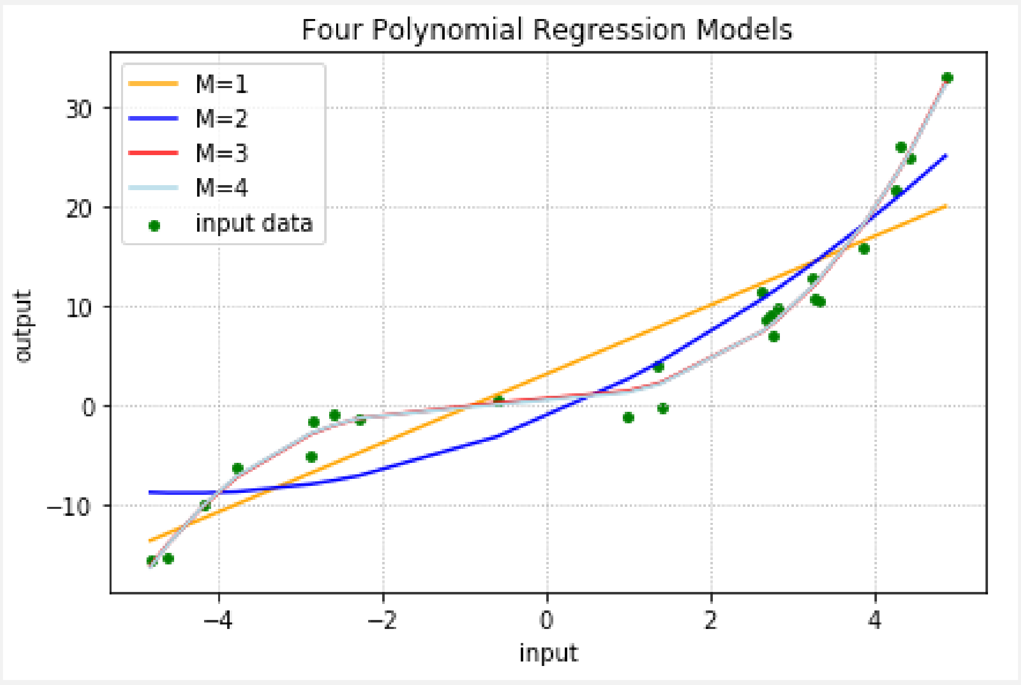
When w_1 varies from -2 to $+2$, MSE decreases and then increases. When $w_1 = 1.354$ ($w_0 = 20$), MSE gets the minimum value of 32.481.
This value is to be expected. According to the figure in question (c), when the intercept (w_0) increases, decreasing the slope (w_1) will better fit the model. So w_1 becomes a little smaller than the previous value (1.441) is to 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 $\phi(x_i) = [1, x_i, x_i^2, x_i^3]$. 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`.

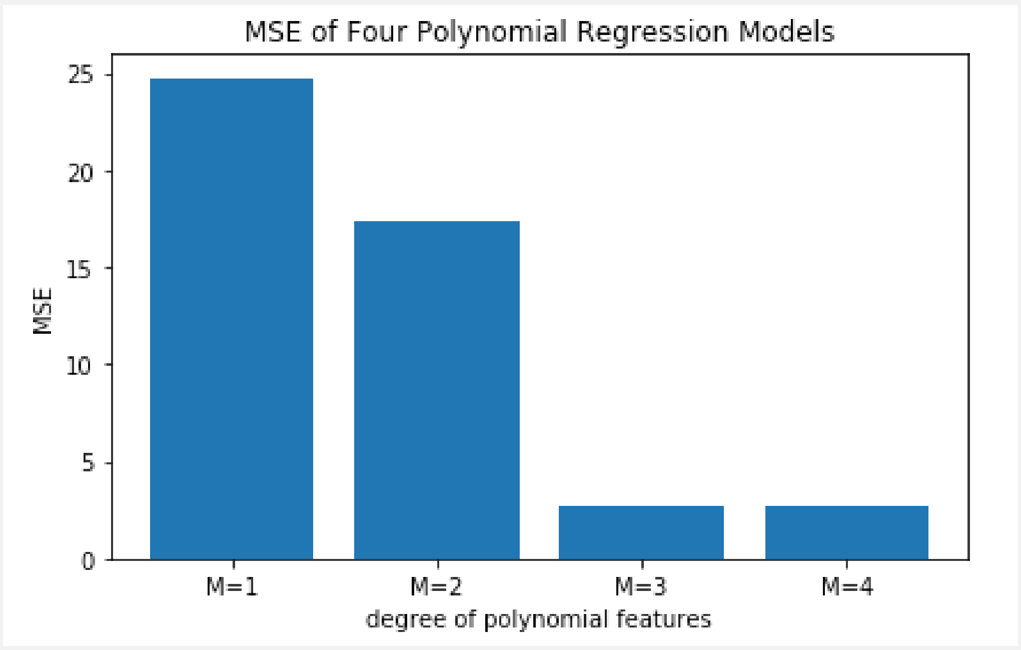
The figure shows four polynomial regression models ($M = 1$ to 4) and the input data.



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.

The figure indicates the Mean Squared Error (MSE) of four polynomial regression models ($M = 1$ to 4).



Question 2 continued on next page...

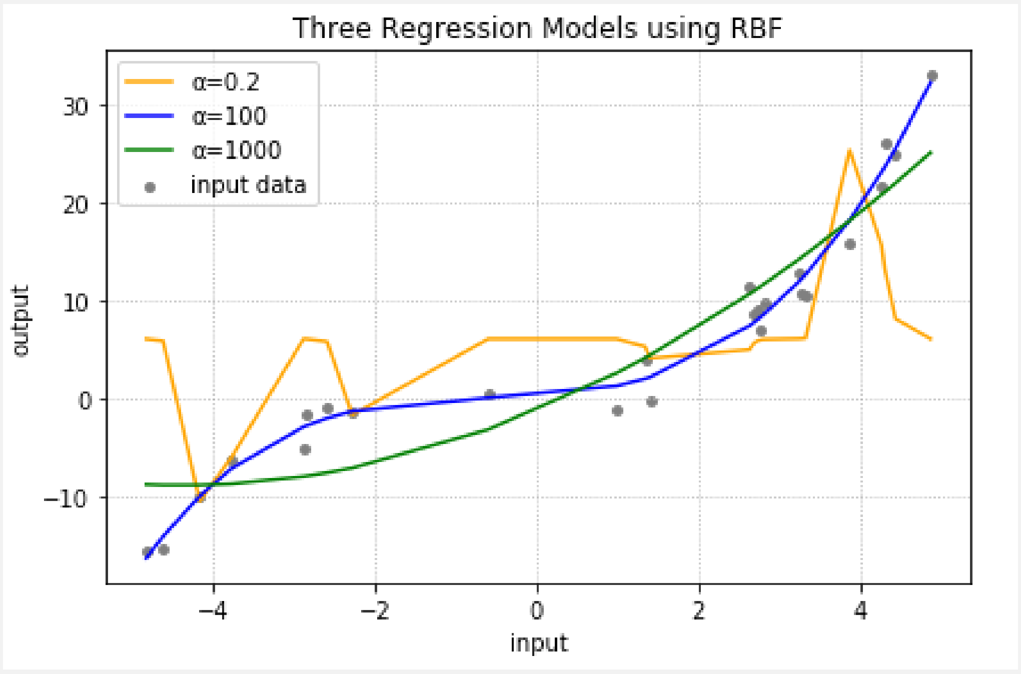
(c) (4 points) Comment on the fit and Mean Squared Error values of the $M = 3$ and $M = 4$ polynomial regression models. Do they result in the same or different performance? Based on these results, which model would you choose?

When $M = 3$, $MSE = 2.745$.
When $M = 4$, $MSE = 2.739$.
At the same time, these two models almost overlap according to the figure in question (a).
They have almost the same performance. Model 4 ($M = 4$) is a little better.
So I would choose the model with $M = 3$. In the case of similar performance, model 3 has low complexity and high efficiency. Model 4 may be overfitting.

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 figure illustrates three regression models using RBF ($\alpha \in \{0.2, 100, 1000\}$).



When α is smaller, the image is steep and the amplitude is large. The model tends to be overfitting.
When α is larger, the image is relatively smooth. The model tends to be underfitting.

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.

The main properties of both the training and test splits are summarised as the table below.

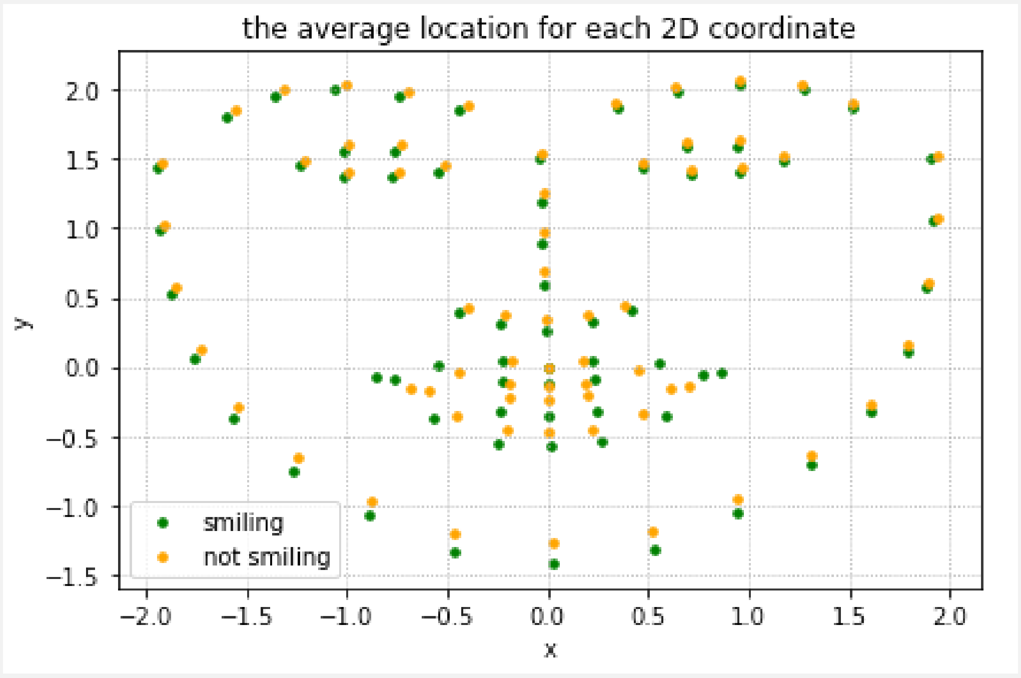
dataset	instance	attribute (exclude the label)	smiling	not smiling
training	4800	136	2335	2465
test	1200	136	592	608

The value in the table represents the number of each category.

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 figure shows the average location for each 2D coordinate of smiling and not smiling faces.



The mouth area of a smiling face is larger than that of a non-smiling face. The corners of the mouth on the smiling face are higher. The position of the lower lip on the smiling face is lower. It looks like an open mouth with a smile.

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 to measure the purity at a node for classification by default.
The calculation of gini impurit doesn't need logarithmic operation. So it's more efficient. Gini impurit is more suitable for continuous attributes, while entropy is more suitable for discrete attributes. In most cases, the decisions made by the two methods are equivalent when the decision tree is split.

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.

A smaller value of the maximum depth of the tree may make the model underfitting. A larger value of the maximum depth of the tree may make the model overfitting. At the same time, it will increase the complexity and computational cost.

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 table shows the training and test accuracy of three decision tree classifiers with a maximum depth of 2, 8, and 20 respectively.

maximum depth	training accuracy(%)	test accuracy(%)
2	79.5	78.2
8	93.4	84.1
20	100.0	81.5

Model 1, model 2, and model 3 refer to models with a maximum depth of 2, 8, and 20 respectively.
I think the model with a maximum depth of 8 is the best. The training accuracy and test accuracy of model 1 are lower than another two. Model 3 has a higher training accuracy but is lower on the test accuracy compared to model 2. So model 3 is overfitting. Model 2 has higher accuracy on both training and test set.

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`.

From high to low, the top three most important attributes are 'x50', 'y48', and 'y29'.
I don't think it makes sense. The mean value of 'x50' is -0.221. From the figure in question (b), we can see that this section on two faces isn't much different. Instead, the corners of the mouth have changed more obviously. Besides, in this classification task, we need to consider the point which consists of two values. So I don't think this attribute is as important as it has been measured.

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.

Using 2D point locations would cause facial information to be lost. The deviation of the position of the object from the standard state may affect the prediction result.

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?

The table shows the classification accuracy for the four different models.

model	alg_1	alg_2	alg_3	alg_4
classification accuracy (%)	61.6	55.0	32.1	32.9

The model alg_1 is the best according to this metric.
A limitation is that only one threshold has been used to calculate the accuracy. We can get the accuracy under several different thresholds and then compare them.

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.

The table indicates the Area Under the ROC Curve (AUC) for four models.

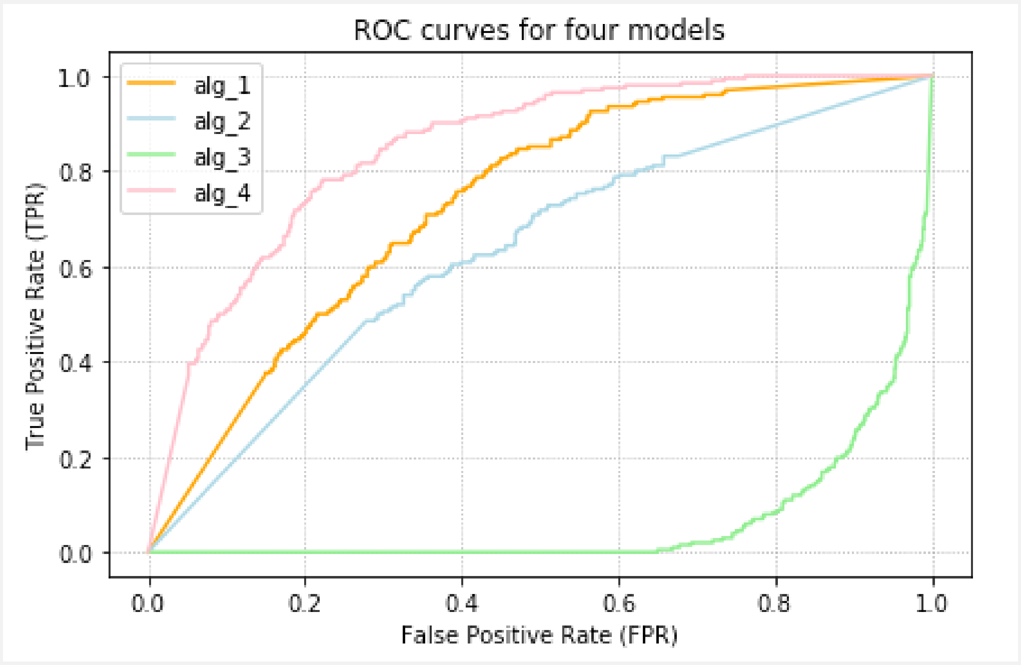
model	alg_1	alg_2	alg_3	alg_4
AUC	0.732	0.632	0.064	0.847

Model alg_4 has the best AUC. It doesn't have the best accuracy.
AUC measures the classification ability of a model, which corresponds to a series of accuracy. Accuracy measures the prediction precision of a model under a certain threshold. So there's no internal relationship between them.

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 figure illustrates ROC curves for four models.



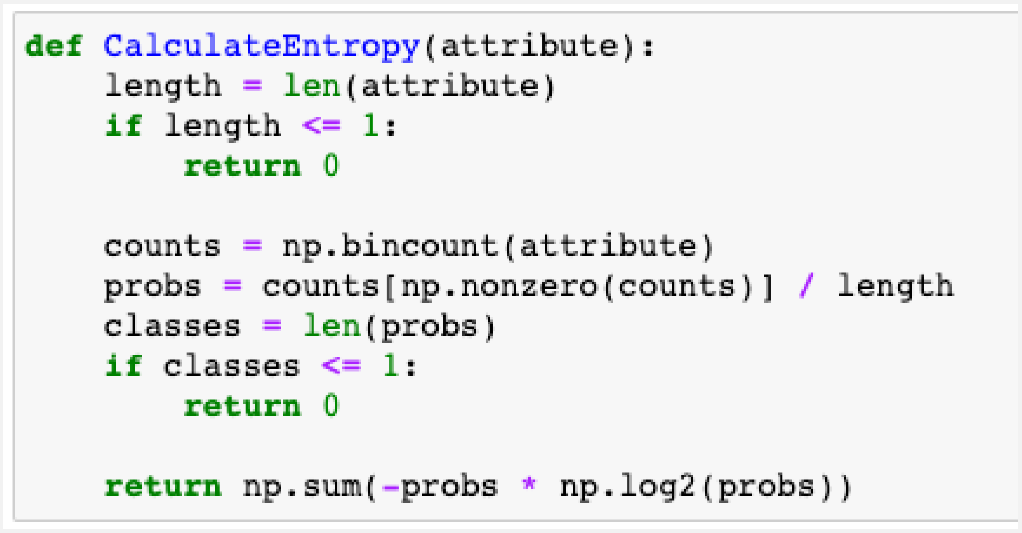
From the ROC curve for alg_3, we can see that the false positive rate is higher than the true positive rate under a certain threshold. It means the classification performance of alg_3 is poor.
The result of this curve being symmetric about the diagonal is reasonable. So we can reverse the label value in alg_3 to improve the performance.

Question 5 : (15 total points) Decision Tree Node Splitting

In this question we will explore how node splitting is performed in decision trees.

(a) (5 points) Compute the entropy of the training binary class labels in bits. Is this value to be expected? You should implement your own entropy function and report your code along with your answer.
Hint: Be sure that this works even when all the training labels belong to the same class. You do not need to include all your code (e.g. we do not need to see the data loading code), just the entropy computation function.

The figure shows the entropy function implemented by myself.



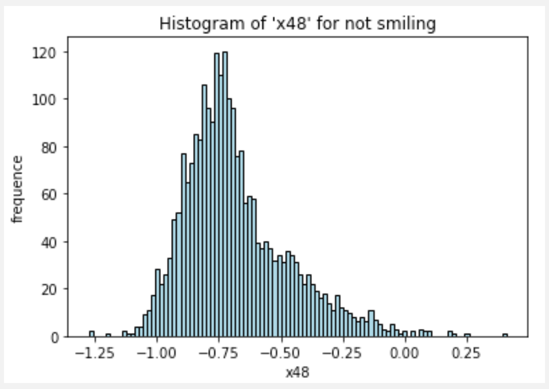
```python
def CalculateEntropy(attribute):
    length = len(attribute)
    if length <= 1:
        return 0

    counts = np.bincount(attribute)
    probs = counts[np.nonzero(counts)] / length
    classes = len(probs)
    if classes <= 1:
        return 0

    return np.sum(-probs * np.log2(probs))
```

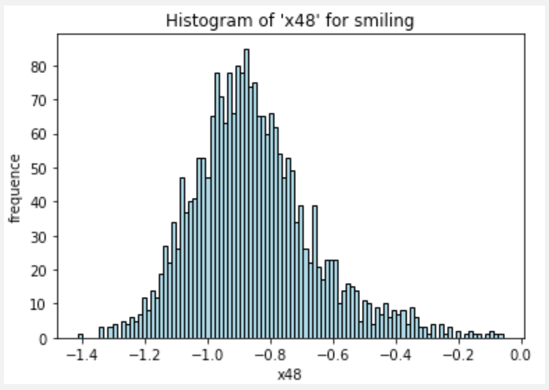
The entropy of the training binary class labels is 0.9995 bits.
This value is to be expected. Because the number of two categories in the label is about the same. Their probability is approximately equal to 1/2. So the calculated entropy is close to 1.

Question 5 continued on next page. . .

(b) (5 points) Plot a 1D histogram of the attribute labelled 'x48' from the training set for the instances where `smiling = 0`. Create a similar plot for the instances where `smiling = 1`. Comment on any differences or similarities between the two histograms for the attribute.
Hint: Set the number of bins in the histogram to 100.

Two figures illustrate 1D histogram of 'x48' for different instances.





Difference: Compared with the histogram on the right, the distribution of the histogram on the left is more concentrated and the maximum frequency is larger.
Similarity: The distribution trends of the two histograms are consistent, and the value ranges are much the same.

Question 5 continued on next page...

(c) (5 points) Compute the information gain resulting from splitting the attribute labelled 'x48' from the training data using splitting thresholds of -0.9 and -0.7. Here, all the entries that are greater to a threshold t (i.e. $> t$) are sent to the right child node and all the entries that are less than or equal to t (i.e. $<= t$) are sent to the left child node. Comment on which threshold results in a better split, and why it is better. You can make reference to your histograms from the previous answer.
Hint: You need to implement your own information gain function. You should report two separate numbers, one for each threshold. You do not need to include your code.

Two results of the information gain are shown in the table.

threshold	-0.9	-0.7
information gain	0.0952	0.0576

Threshold of -0.9 results in a better split.
According to two histograms, using a splitting threshold of -0.9 will get purer data.
So after splitting, the entropy is lower and the information gain becomes higher.