IAML – INFR10069 (LEVEL 10): Assignment #1

s1810054

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.

A total of 50 samples is taken.Each sample has the data of revision time and exam score. All the data are represented by floating-point numbers, with revision time ranging from 0 to 50 and exam score ranging from 0 to 100.

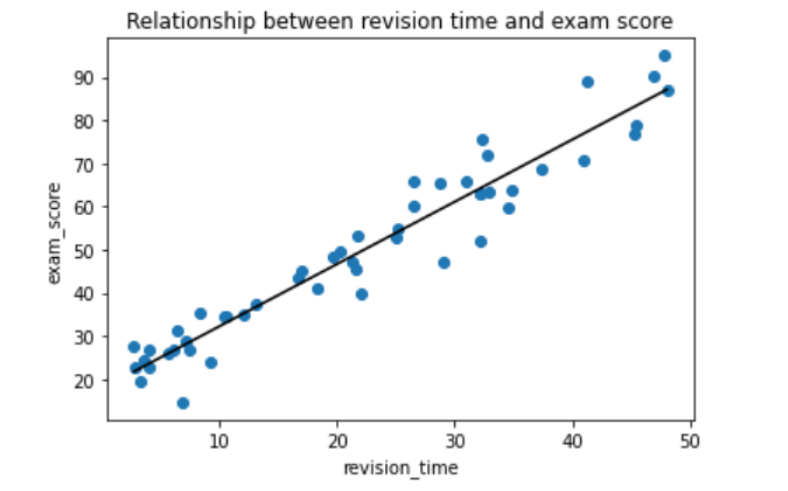
Question 1 continued on next page...

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

Estimated model parameter $\mathbf{x} = (17.89768026, 1.44114091)$.
The number 17.89768 represents the predicted exam score when zero time is spent for revision.
The number 1.44114 represents the expected increase in exam score for every one hour spent for revision.

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.*

```
x=np.array([[1,part1['revision_time'][i]] for i in range (50)])
y=np.array(part1['exam_score'])
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.

$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$
One limitation of MSE is that it is very sensitive to outliers.

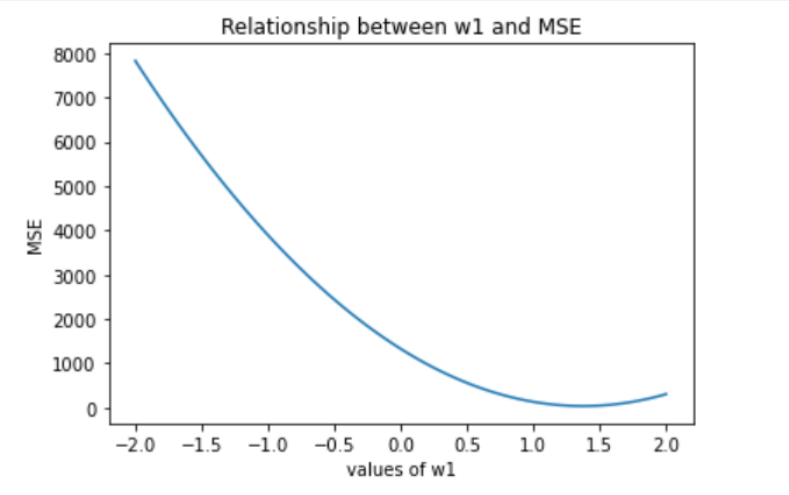
Question 1 continued on next page...

(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.

$MSE_{sklearn} = 30.98547$
$MSE_{closed-form} = 30.98547$
The MSE for two models are the same.

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)`.

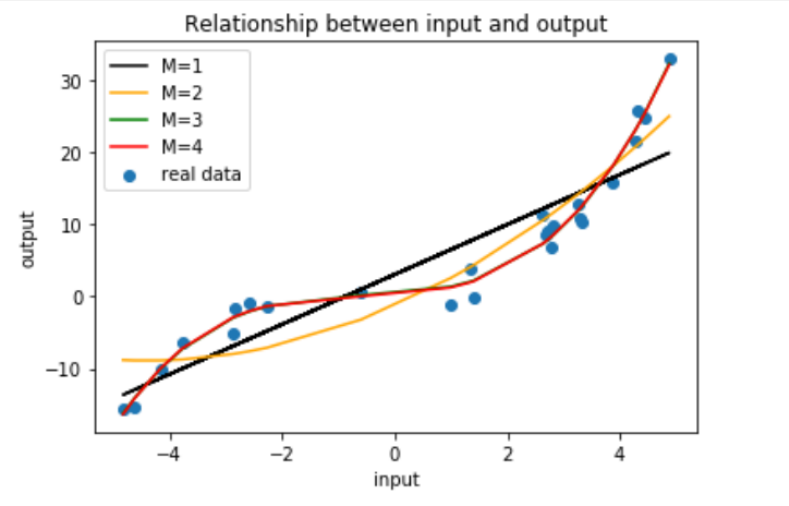


As can be seen from the graph,at first MSE decreases as w_1 increases. Then MSE reaches its minimum and increases afterwards.
The value of w1 for which MSE is minimum is 1.353535 and the minimum value of MSE is 32.48096.
This value is expected because we choose $w_0 = 20$, which is already rather far away from the value calculated 17.89768.Hence,I would expect the MSE to be larger than the one we obtained earlier no matter how good our choice of w_1 is.

Question 2 : (18 total points) Nonlinear Regression

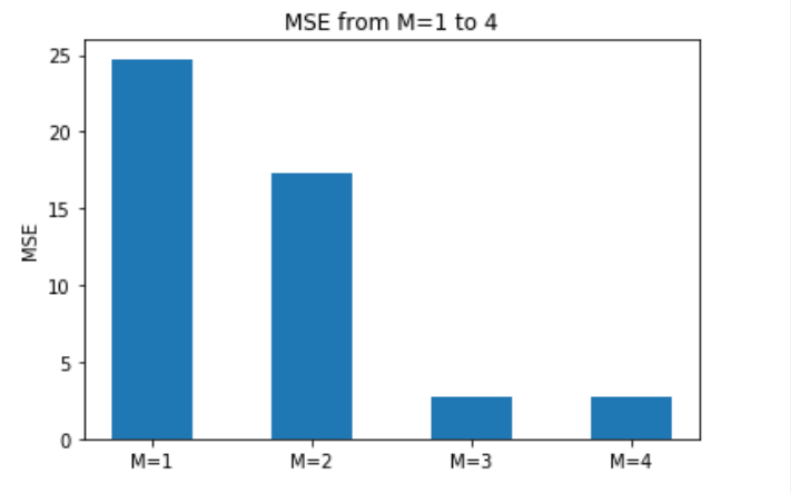
In this question we will tackle regression using basis functions.

(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.



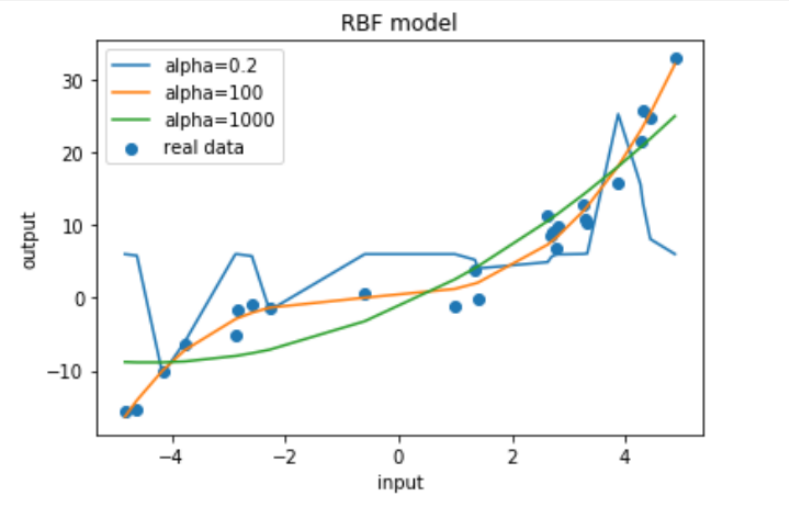
Question 2 continued on next page...

(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 fits for both M=3 and M=4 are very close to the actual values,with MSE for M=4 being slightly smaller than MSE for M=3. However, I would choose the model with M=3 because it is much less demanding in terms of computation even though it has a slightly larger error.

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 optimum fit is when $\alpha = 100$, then the quality of fit decreases for both smaller and larger value of α.

Question 3 : (26 total points) Decision Trees

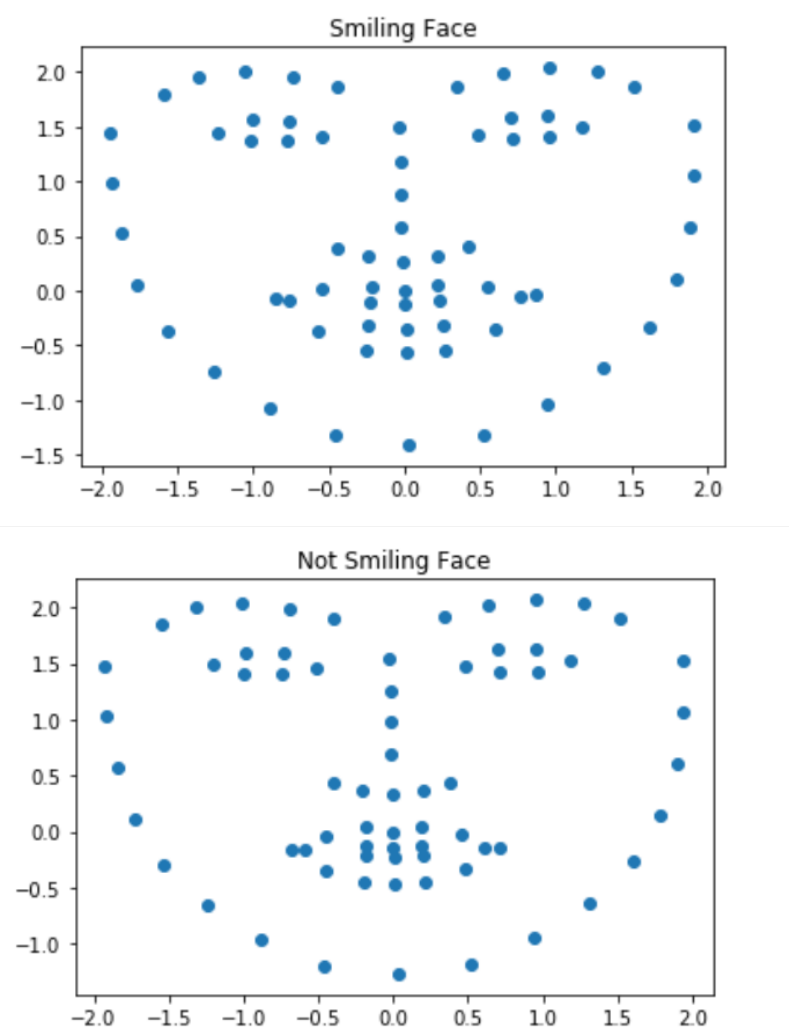
In this question we will train a classifier to predict if a person is smiling or not.

(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.

Both training and test data have 136 attributes and class 'smiling' that we want to predict. Those 136 attributes are paired up as x and y-coordinates, which gives 68 coordinates in total. The values of x and y-coordinates range roughly from -4 to 4.

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.



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?

Gini index is used by default. One advantage of Gini compared to entropy is that it is less demanding in terms of computation as it does not involve logarithms.

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.

Small maximum depth could result in underfitting.Big maximum depth could result in overfitting as it captures too many details of the training data, which may not be relevant to new data.Another problem for big maximum depth is that it requires too much computation.

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.*

max_depth	Train Accuracy	Test Accuracy
2	0.79479	0.78167
8	0.93354	0.84083
20	1.0	0.81583

The model with $max\_depth = 8$ is the best because it suffers from neither underfitting nor overfitting. Its testing accuracy is the highest, which means it is more suitable for future unseen data.

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`.

Features	Gini importance
$x50$	0.3304
$y48$	0.08996
$y29$	0.08831

The result does not make sense because $x50$ is the horizontal position of the middle of the upper lip, which should not change much no matter a person smiles or not.

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.

One limitation is that the data are numerical. They are continuously distributed, which means there will be a lot of possible values that input data can take i.e. many values the node can split on.This will increase the time complexity of fitting the decision trees.

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?

classification algorithm	Classification Accuracy
alg_1	0.616
alg_2	0.55
alg_3	0.321
alg_4	0.329

The model alg_1 is the best according to this metric.One limitation is that we lose a lot of information when using the threshold value to convert to binary predictions. For example, 0.01 and 0.49 are both 0 with threshold being 0.5 even though they are much more different.

Question 4 continued on next page...

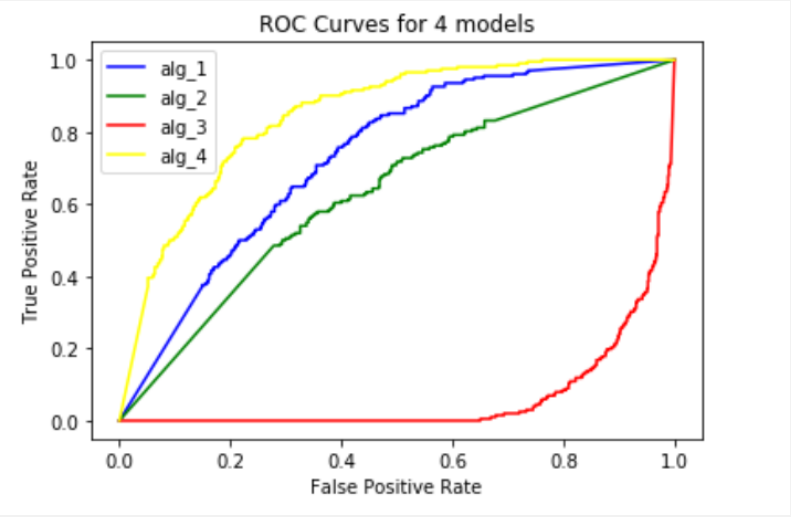
(b) (4 points) Instead of using classification accuracy, report the Area Under the ROC Curve (AUC) for each model. Does the model with the best AUC also have the best accuracy? If not, why not?
Hint: You can use the roc_auc_score function from sklearn.

classification algorithm	AUC
alg_1	0.73209
alg_2	0.63163
alg_3	0.06395
alg_4	0.84739

The model alg_1 has the highest classification accuracy yet the model alg_4 has the highest AUC.This is because only 0.5 is used as the threshold for calculating accuracy while AUC takes the average of the accuracies with all possible thresholds. Hence, a model with higher accuracy may not have the higher 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.



As can be seen from the graph, the curve for alg_3 is below the curve for random performance. The accuracy of alg_3 can be improved by adjusting the sample weights.