IAML – INFR10069 (LEVEL 10): Assignment #1

s1803764

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.

Data properties of the 'regression_part1.csv' dataset:

Dataset size: 50 records
Dataset shape: (50, 2)
Data types: all attribute values of type float64

ATTRIBUTES:

revision_time (in hours)

Possible data range: $0 \leq \mathbf{x}_i \leq \alpha$, where α represents the available hours of study
Dataset data range: $2.723 \leq \mathbf{x}_i \leq 48.011$
Mean: $\mu_x = 22.220019999999998$
Standard deviation: $\sigma_x = 13.986112431936743$

exam_score (in %)

Possible data range: $0 \leq \mathbf{y}_i \leq 100$
Dataset data range: $14.731 \leq \mathbf{y}_i \leq 94.945$
Mean: $\mu_y = 49.91986$
Standard deviation: $\sigma_y = 20.92559441626157$

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

Fitting a linear model to predict exam_score from revision_time

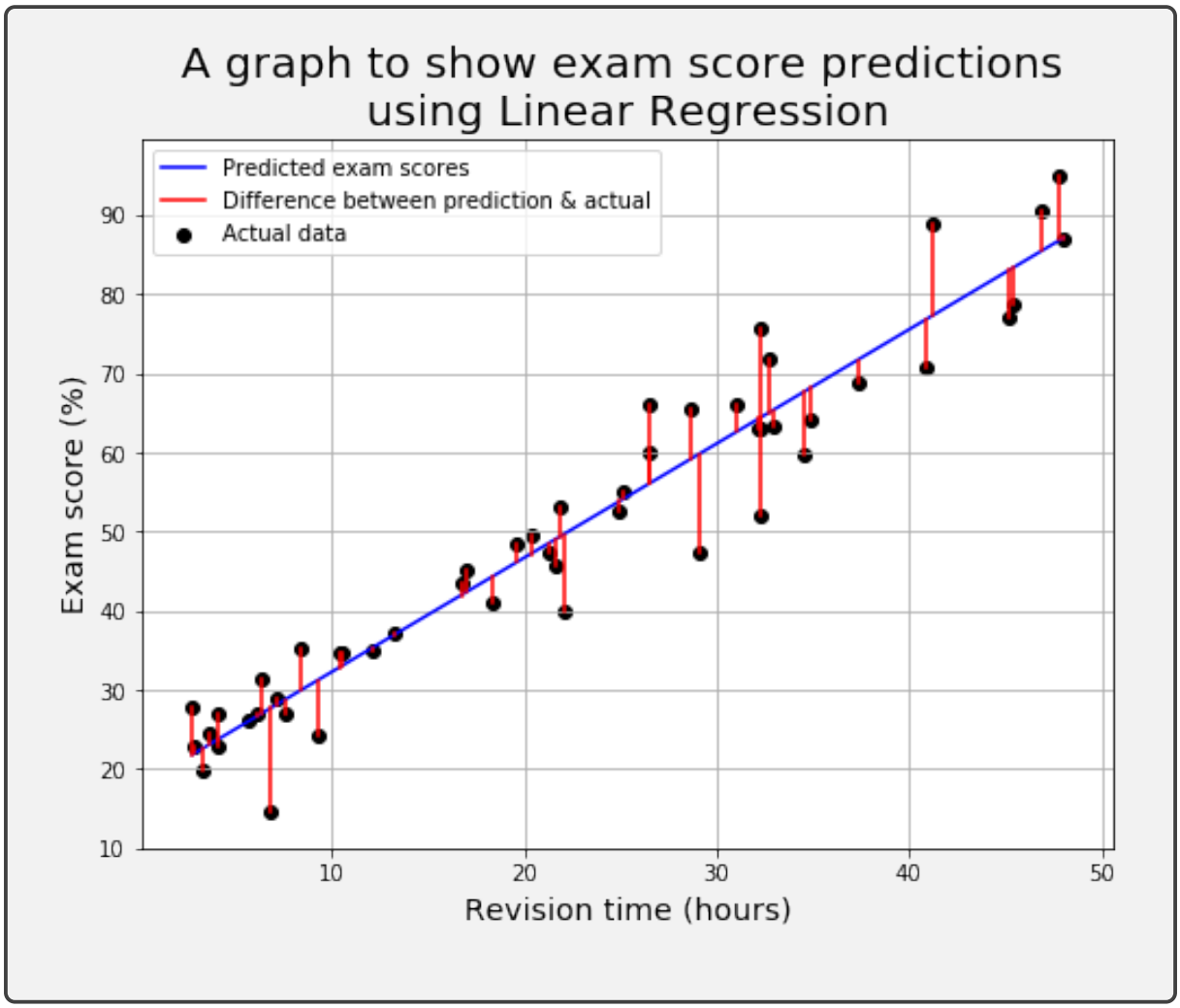
Linear model:
Coefficient = 1.44114091
Intercept = 17.89768025835017

Therefore:
$\boldsymbol{w} \approx [17.898, 1.441]$

Representation of $\boldsymbol{w}$:
$\boldsymbol{w_0}$ represents the y-intercept of the model and thus denotes the minimum estimated possible exam score.
$\boldsymbol{w_1}$ represents the gradient of the model and thus quantifies the direct proportionality between the student's hours of study and their exam score.

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.*

Given our analytical solution for calculating $\hat{\mathbf{w}}$:

$$\hat{\mathbf{w}} = (\mathbf{\Phi^T} \cdot \mathbf{\Phi})^{-1} \cdot \mathbf{\Phi^T} \cdot \mathbf{Y}$$

where $(\mathbf{\Phi^T} \cdot \mathbf{\Phi})^{-1} \cdot \mathbf{\Phi^T}$ is the pseudo-inverse of $\mathbf{\Phi}$

We can easily transfer this into code:

```
pseudoInversePhi = np.matmul(np.matrix(np.matmul(phi.transpose(), phi)).I, phi.transpose())

w_hat = np.matmul(pseudoInversePhi, Y)

print(w_hat)
```

```
[[17.89768026]
 [ 1.44114091]]
```

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.

Limitations of the Mean Squared Error (MSE) metric

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

One of the disadvantages of the MSE metric is that it is very sensitive to outliers. This is evident in the equation as shown above, where for each value of x_i the error value (difference in the predicted and actual value) of y_i is squared and the average error value is calculated. This squaring of error values ultimately magnifies the large errors (caused by outliers) and minimizes the small errors. Inevitably meaning a single outlier in our dataset could skew our analysis greatly, due to the fact that we could get the same MSE for a predictor with one large error as a predictor with lots of small errors.

The best way to solve such a problem would be through either identifying outliers in our dataset through visualization or using different error metrics which are more robust against outliers (eg. Median Absolute Deviation).

Question 1 continued on next page. . .