IAML – INFR11182 (LEVEL 11)
CLOSING DATE: MON, OCT 18, 2021 @ 16:00

N.B. *This document is best viewed on a screen as it contains a number of (highlighted) clickable hyperlinks.*
It is very important that you read and follow the instructions below to the letter. You will be deducted marks for not adhering to the advice below.

Good Scholarly Practice: Please remember the University requirement regarding all assessed work for credit. Details about this can be found at:

http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct

Specifically, this assignment should be your own individual work. Moreover, please note that Piazza is **NOT** a forum for discussing the solutions of the assignment. You may, in exceptional circumstances, ask **private** questions to the instructors if you deem that something may be incorrect, and if we feel that the issue is justified, we will send out an announcement.

General Instructions

- There are two versions of this assignment. One for INFR10069 (level 10) and the other for INFR11182 (level 11). The level 11 version has one additional part. **MAKE SURE you are doing the assignment that corresponds to the course you are registered on; you can check this on EUCLID.**
- You should use **Noteable** for implementing your solutions as this will standardise the output and also provide a consistent experience with the labs. Set up your environment as specified in the Labs. **It is VERY IMPORTANT that you use the exact same package versions as those specified in the requirements file from the labs!**
- If running `import sklearn; print(sklearn.__version__)` in your Jupyter Notebook does not print the package version `0.24.2`, then you are *not* using the correct environment.
- This assignment will not be assessed but you should still complete and submit a written report (compiled from a latex template which we provide) to learn how to submit coursework 2, which will be assessed.
- Read the instructions carefully, answering what is required and only that. Keep your answers brief and concise. Specifically, **for textual answers**, the size of the text-box in the latex template will give you an idea of the **maximum** length of your answer. You do **not need** to fill in the whole text-box but you will be penalised if you go over. This does not apply to figure-based answers.

- For answers involving figures, make sure to clearly label your plots and provide legends where necessary. You will be penalised if the visualisations are not clear.
- For answers involving numerical values, use correct units where appropriate and format floating point values to an appropriate number of decimal places. You will be penalised for using too many decimal places.

Submission Mechanics

Important: ***You must submit this assignment by Monday 18/10/2020 at 16:00. We do not accept late Submissions for this coursework, except in the case of mitigating circumstances. Please refer to the ITO Website for further details.***

- We will use the Gradescope submission system for uploading PDF assignments. Information describing how to upload your completed assignment will be made available on the IAML Learn page.
- You should clone or download the Assignment Repository from https://github.com/uoe-iaml/INFR11182-2021-CW1.

  This contains:

  1. The data you will need for the assignment under the `data` directory.
  2. Two `tex` files, `Assignment_1.tex` and `style.tex`. These provide the template for you to fill out the assignment questions. In particular, the template forces your answers to appear on separate pages and also controls the length of textual answers.
- You should **only** modify the `Assignment_1.tex` template by:

  1. Uncommenting and specifying your student number at the top of the document (compilation will automatically fail if you forget to do this). Remove the '%' and enter your student number e.g.

     ```
     \newcommand{\assignmentAuthorName}{s1234567}
     ```

  2. Filling in the answers in the provided `answerbox` environment.

  DO NOT modify anything else in the template and certainly **DO NOT** edit the style file.

Latex Tips

- To fill in text answers, you can modify the corresponding `answerbox`:

  ```
  \begin{answerbox}{5em}
  Your answer here
  \end{answerbox}
  ```

  with your answer (replacing 'Your answer here'):

```
\begin{answerbox}{5em}
Steam locomotives were first developed in the United Kingdom
during the early 19th century and used for railway transport
until the middle of the 20th century.
\end{answerbox}
```

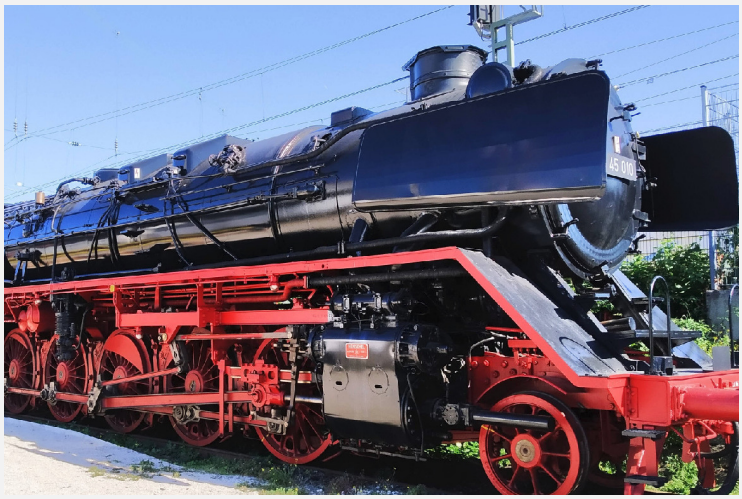
which, when compiled gives:

> Steam locomotives were first developed in the United Kingdom during the early 19th century and used for railway transport until the middle of the 20th century.

- To add an image, you can use:

```
\begin{answerbox}{18em}
This image shows a train.
\begin{center}
\includegraphics[width=0.7\textwidth]{stock_image.jpg}
\end{center}
\end{answerbox}
```

which will be compiled to:

> This image shows a train.
>
> 

Make sure that you specify the correct path to your image. For example. if your image was stored in a directory called `results`, you would change the relevant line to read:

```
\includegraphics[width=0.7\textwidth]{results/stock_image.jpg}
```

You can find more information about inserting images into latex documents here.

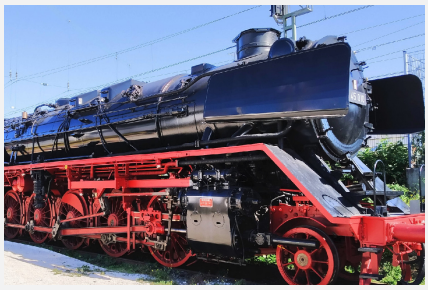
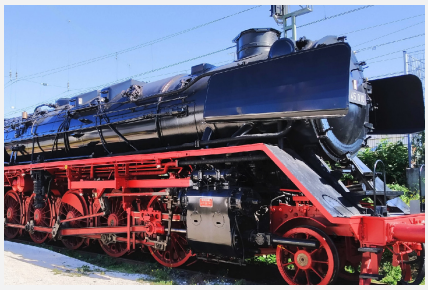
- You can also add two images side-by-side:

```
\begin{answerbox}{18em}
Below we see two trains.

\begin{center}
\begin{tabular}{ll}
\includegraphics[width=0.4\textwidth]{stock_image.jpg}
&
\includegraphics[width=0.4\textwidth]{stock_image.jpg}
\end{tabular}
\end{center}
\end{answerbox}
```

which will be compiled to:

Below we see two trains.

- To add an inline equation, you can use the '$' symbol to write:

```
\begin{answerbox}{3em}
I am using the following model, $y = \mathbf{x}^T\mathbf{w}$.
\end{answerbox}
```

which compiles to:

I am using the following model, $y = \mathbf{x}^T\mathbf{w}$.

- To add a table for numerical results you can use:

```
\begin{answerbox}{7em}
Results are presented in the table below.

\begin{center}
\begin{tabular}{ |c|c|c| }
 \hline
 Parameter Value & Train Accuracy & Test Accuracy \\ \hline
 1 & 10.1\% & 9.1\% \\
```

```
 2 & 12.5\% & 10.1\% \\
 \hline
\end{tabular}
\end{center}
\end{answerbox}
```

which compiles to:

Results are presented in the table below.

Parameter Value	Train Accuracy	Test Accuracy
1	10.1%	9.1%
2	12.5%	10.1%

You can find more information about tables in latex here.

- For a small number of questions we may ask you to report your code. You can include code as an image, but if you prefer you can use the following command:

```
\begin{answerbox}{5em}
\begin{verbatim}
import numpy as np
mean_time = 10.0
print('mean time', mean_time)
\end{verbatim}
\end{answerbox}
```

which, when compiled gives:

```
import numpy as np
mean_time = 10.0
print('mean time', mean_time)
```

- Once you have filled in all the answers, compile the latex document to generate the PDF that you will submit. You can use Overleaf, your favourite latex editor, or just run `pdflatex Assignment_1.tex` twice on a DICE machine to compile the PDF.

Question 1 : Linear Regression

We will fit linear regression models to the data in file `regression_part1.csv`.

We will investigate the relationship between the amount of time (in hours) each student in a class studied for an exam and their end of semester exam performance. We will model this relationship for each student using $y_i = \boldsymbol{\phi}(x_i)\mathbf{w}$, where $\boldsymbol{\phi}(x_i) = [1, x_i]$ is a row vector and $\mathbf{w}$ are the model parameters we will learn. Here, y_i is the exam score for student i and x_i is the amount of time they spent revising.
The dataset is contained in `regression_part1.csv`. You should load it into a Pandas DataFrame using `pandas.read_csv()`.

(a) Describe the main properties of the data, focusing on the size, data ranges, and data types.

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

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

(d) 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.

(e) 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.

(f) 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.

Question 1 continued on next page...

(g) 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)`.