Linear Models

Lab 1

Jay Paul Morgan

Introduction

Welcome to the first Machine Learning lab session! For this first session, we will be implementing and using linear models. In the following session we will be answering the questions below, of which will be marked using the marking criteria on the last page.

How to answer the questions

In these lab sessions we will implement what we've learn during the theory lectures. These implementation will need to be using the Python Programming Language. You're allowed to use 3rd party libraries such as pandas, matplotlib, numpy. But it is not allowed to use a library that implements exactly machine learning model we're learning about. For example, in this lab, we're learning about linear models, so you cannot use a library that provides functionality for computing a linear model. If you're uncertain about whether a library can be used, please do ask!

I am going to give you the maximum flexibility with how you want to program these linear models: you can create a Python script, or if you like, you can use jupyter notebooks. In both cases, You should preface your implementation with which question you're answering with a comment (if you're writing a python script), or markdown (if you want to use jupyter) like this:

```
# Q1. Download, parse, and load the data into a pandas dataframe,
# print the first 5 rows to ensure the data is formated correctly
import pandas as pd

def load_data(filepath: str) -> pd.DataFrame:
    # load_data logic
    return df
```

You are to work individually.

In summary:

- Work individually.
- Implementation should be using the Python programming language.
- You can use supplementary libraries such as pandas, matplotlib, numpy, but cannot use a library that provides a function call for the assignment.

Submission Procedure

To hand in your assignment, please zip all of your source code (**do not include the data**) into a zip-archive named using the following format:

```
<first-name>-<surname>-machine-learning-lab-1.zip
```

replacing `<first-name>` and `<surname>` with your name. Please send this assignment to my email address: `jay.morgan@univ-tln.fr` using the subject `Machine Learning Assignment 1` by the end of the lab session. I will accept late assignments, but will be deducted score accordingly.

Questions

Question 1

To start this lab session we want to setup our project. For this question, do the following:

- Create a new directory for this lab.
- Create a new conda environment.
- In this new conda environment, install `python`, `pandas`, `matplotlib`, and `numpy`.
- Export this conda environment to an `environment.yml` into the root of the directory.

Question 2

To implement a linear regression model, we will be using a toy dataset to ensure we've implemented the model correctly. To begin with this lab, we will want to download the boston dataset from https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html. Download and save the dataset exactly as it is, and save it as a CSV.

After we've downloaded the CSV file, we will want to parse and load the data into a pandas DataFrame.

Create a `load_boston_data(filepath: str) -> pd.DataFrame` function.

This may require iterating through each line in the file until you read the data, at this point you will need to parse the data. Finally, return the data as a dataframe.

Question 3

Visualise some scatter plots of the columns of your choice against the target house price column (i.e. the column of your choice will be on the x-axis, will house price will be the y-axis).

Decide what you think will be the singular best column to use for using a linear model to predict the house price.

What is this column?

Question 4

Create a function called `lm`, that takes an `x`, and `y`, and returns the random `m` and `b` variables in the linear equation:

$$y = mx + b$$

```
def lm(x, y) -> tuple[float, float]:
    # create random m, b
    return m, b
```

Question 5

Using these `m, b` variables, create a housing price prediction for each row of data.

Question 6

Create a function `mae` that calculate the mean absolute error of the true house price value and the predicted value. What is the error?

```
def mae(y, y_pred) -> float:
    # calculate the mean absolute error
    return error
```

Question 7

Visualise the linear model returned from `lm` on top of the scatter plot of the input and target data.

Question 8

Re-make the `lm` function. This time, when called with an `x, y` it returns the optimal `m` and `b`.

You are free to either implement least-squares regression, or the gradient descent method.

```
def lm(x, y) -> tuple[float, float]:
    # create the optimal values for m, b
    return m, b
```

Question 9

Re-plot this linear model against the scatter plot.

Question 10

Re-calculate the mean absolute error for these optimal `m, b` variables. What is the error now?

Marking Criteria

Criteria	**Marks**	**Not attempted** (0%)	**Attempted** (0-30%)	**Correct** (30-80%)	**Good solution** (80-100%)	**Score**
Question 1		-				
Question 2	10					
Question 3	10					
Question 4	5					
Question 5	10					
Question 6	10					
Question 7	10					
Question 8	20					

Criteria	**Marks**	**Not attempted** (0%)	**Attempted** (0-30%)	**Correct** (30-80%)	**Good solution** (80-100%)	**Score**
Question 9	5					
Question 10	5					
Code comments are helpful	2					
Variable names are descriptive	2					
Functions include docstrings	2					
Functions are generic	4					
					Total	