School of Computer Science and Mathematics
Kingston University
CI6115: Programming III
Module Leader: Ahmed Shihab

Coursework II (Machine Learning with Python and Language Design)

Set by Jad Abbass and Ahmed Shihab

Weighting and allocations

This assessment constitutes 50% of the total marks of the module. Each question includes mark allocations.

Important notice: The course team reserves the right to invite any student to a viva to discuss their submitted work. The mark given at the viva will be the final mark for the submission. The viva will be an in-person session that lasts around 20 minutes. In the session, questions designed to elicit and validate authorship of the work will be asked. Failure to attend a viva if one is called will result in an F0 mark (non-submission) being awarded for the coursework.

Deadline: 20th April 2025.

"Machine Learning with Python"
Questions set by Drs. Abbas and Shihab
50% of the marks

Submission guidelines

- Create a folder called "Name-k1234567-CI6115-CW-MLWP", where Name is your surname and k1234567 is your knumber.
- Create sub-folders within the folder: "Q1", "Q2" and "Q3", each including the documents and Jupyter Notebook files for the appropriate question. Only one Jupyter notebook per question please; discussion, answers and notes should be provided inside the Jupyter Notebook as comments in Markdown.
- Zip up the "Name-k1234567-CI6115-CW-MLWP" top folder – use only standard zip format (not .7z or .rar).
- Upload the zipped file to the relevant Canvas assignment dropbox.
- Failure to keep to these guidelines risks loss of marks due to lack of attention to detail.

Question 1 [17 marks]

Consider the dataset entitled: "googleplaystore_new.csv" *available on Canvas*, and answer the following in ONE "Jupyter Notebook" File

A. Check for any missing values. If any, display all rows that contain null values, then delete them. Then, check for duplicates (similar contents for every feature). If any, delete the redundant rows. **[1 mark]**

B. Show the average rating per each 'Category' using the appropriate plot. **[1 mark]**

C. Create a new column called 'Size in bytes' (numeric) and convert the entries from 'Size' column (M means megabyte and k means kilobytes). **[1 mark]**

D. Create a new column called 'Numeric_installs' (numeric) and convert the entries from 'Installs' column (remove '+' and ','; for example, "5,000,000+" becomes "5000000" as an Integer). **[1 mark]**

E. Delete the following columns: "Type", "Price", "Genres", "Android Ver", "Size", "Installs" and "App" and Save the new CSV file as "Updated_googleplaystore.csv". **[1 mark]**

F. Using **all remaining features** ***(Category + Reviews + Content Rating + Size in Bytes + Installs Num)***, find the best regression model to predict "rating" (use the standard training/test partition without cross-validation). **[4 marks]**

G. Drop "Installs_Num" and "Size in Bytes" columns and repeat task F, what do you conclude? **[2 marks]**

H. Drop "Installs_Num", "Size in Bytes", "Category" and "Content Rating" columns (That is, using only "Reviews") and repeat task F, what do you conclude? **[2 marks]**

I. Back to task F, repeat it using a range of different Test Dataset sizes (ranging from 90% to 10% - deduct 10% at each iteration). Show the results on (a) plot(s), how would you interpret this? **[4 marks]**

Question 2 [13 marks]

Consider the dataset entitled: "imdb_top_1000_New.csv" (available on Canvas), and answer the following in ONE "Jupyter Notebook" File

A. Drop 'Series_Title', 'Certificate, 'recorded_date', and 'Runtime' columns. **[1 mark]**.

B. We want to predict the "IMDB_Rating" based on the remaining features, that is, Released_Year, Runtime2, No_of_Votes, Gross, and Genre (consider the 'Released_Year' feature a continuous value not a categorical feature). Find the regression model that best fits the model using the **standard train/test split** along with a **cross-validation approach (K=5)**. Based on the overall RMSE, do you think the model is reliable? **[7 marks]**

C. Repeat task B, however, to predict the "Gross". **[2 marks]**

D. Although predicting the 'Released_Year' is a classification problem, let's try to solve it through a regression problem. Repeat task B, however, to predict the "Released_Year'. **[3 marks]**

<u>Question 3</u> [20 marks]

Medication persistence is a critical issue in healthcare, and one key metric that measures it is *persistency* — the length of time patients continue taking their prescribed medications. Defined as "the duration of time from initiation to discontinuation of therapy," persistency is a crucial indicator of treatment success. By tracking persistency, pharmaceutical companies can gauge the effectiveness of their treatments and identify opportunities to improve patient outcomes.

Consider the dataset provided in this link:
https://www.kaggle.com/datasets/harbhajansingh21/persistent-vs-nonpersistent

The dataset uses patient information, provider attributes and clinical and treatment factors to predict the likelihood of patients continuing to use a particular medication over an extended period as prescribed by their healthcare provider.

Study the dataset carefully and familiarise yourself with each column. Consider the type of data in each column.

Now examine the solution provided in this link:
https://www.kaggle.com/code/harbhajansingh21/logistic-regression-vs-svm-hypothesis-testing

This solution carries out some exploratory data analysis, splits the data into training and test parts and trains a Logistic Regression model and an SVM model. Study the notebook carefully. Take time to run it yourself.

Train a KNN model and compare its performance to that of the Logistic Regression model. Hand in a Jupyter notebook containing the KNN model.

The notebook should show:

- Selection of an appropriate value for 'k' through methods such as the elbow method.
- Use of cross-validation.
- Evaluation of performance using relevant metrics such as accuracy, precision, recall, and F1-score.
- Comparison of the performance of the KNN model to that of the Logistic Regression model using the same evaluation metrics.
- Summary of findings via discussion of any differences in performance, and insights into why one model may outperform the other based on the characteristics of the dataset.
- Reflective evaluation of the problem from data to models, including various factors that a prospective ML engineer would bring up when looking at the data and models.
  [20 marks]

Include copious comments in your notebook. Find ways to show the originality and authorship of your work. Highlight the ways in which your notebook is not generic and is different to material that would be generated by AI.

Submission format: a Jupyter notebook, with copious comments in Markdown.

"Language Design (DYOPL)"
Questions set by Dr Ahmed Shihab
50% of the marks

Important notice: All lecture materials provided for this part are proprietary to the University and have not been published elsewhere. As such, they are not publicly available on the internet. Sharing *fspow* material with AI tools is a serious breach of academic integrity. Please be aware that any attempt to copy and paste course materials into AI engines or share them with external tools will be taken very seriously and may result in consequences. It is essential that you respect the intellectual property rights of the University and the originality of the course materials. Students are expected to uphold the highest standards of academic integrity and refrain from sharing any course materials without explicit permission. AI tools may still be used to assist in learning, but such use must be carried out whilst respecting intellectual property rights.

Submission guidelines

- Create a folder called "Name-k1234567-CI6115-CW-DYOPL", where Name is your surname and k1234567 is your knumber.
- Create sub-folders within the folder: "Q1", "Q2", "Q3" and "Q4", each including the documents and source code files for the appropriate question.
- Zip up the "Name-k1234567-CI6115-CW-DYOPL", top folder – use only standard zip format (not .7z or .rar).
- Upload the zipped file to the relevant Canvas assignment dropbox.
- Failure to keep to these guidelines risks loss of marks due to lack of attention to detail.

Question 1 [10 marks]

Provide evidence to show that you followed the early workshops and managed to successfully set up an environment suitable for using ANTLR.

Create a Word document and make sure the document contains headings, external links, screenshots, code samples, and other media (if needed) to show:

- the various steps that you followed in setting up your DYOPL environment, **[4 marks]**
- how you tested the ANTLR installation, **[1 mark]**
- how you parsed a simple script of a toy language, **[1 mark]**
- how you used ANTLR to translate the simple script into Python, **[1 mark]**
- your thoughts and reflections on this experience. **[3 marks]**

Show evidence through screenshots to show that the actions expected from the script did occur.

Guideline length: aim for five pages; eight pages is too long.

Question 2 [12 marks]

In the lectures and workshops, we saw how to add to *fspow* capability the ability to do logical combinations of filters. Bring together all the efforts explained during teaching to let *fspow* support complex filter combinations. E.g., this snippet of code should work.

```
allFiles = FileCollection("directory")

complexFiles = Selector(name("*.mp4") intersect (not (size(">      300M") intersect
date(">1year"))))

allFiles.apply(complexFiles)
```

You will need to define all the appropriate grammar changes and write the appropriate instructions in Python to execute the scripts.

Marking criteria:

- correctness of syntax enhancements, [3 marks]
- evidence of testing the syntax enhancements, [2 marks]
- translation into Python, [3 marks]
- extensive testing of fspow scripts. [4 marks]

<u>Question 3</u> [12 marks]

Extend *fspow* to give it the ability to provide the "top 10" of a file-collection. Two options should be available: top 10 by size, top 10 by modification date. E.g., this snippet of code should work.

```
sel = Selector(top(10, Biggest))
topBiggest = fc.apply(sel)
sel = Selector(top(20, Oldest))
topOldest = fc.apply(sel)
```

The selector "top" should have two parameters:

- number: a literal number that represents the number of elements to return. E.g., 10.
- attribute: a keyword to represent an option.
    - `Biggest` would indicate top files by size, in decreasing order
    - `Oldest` would indicate top files by modification date, from the oldest to more recent ones
    - `Smallest` would indicate top files by increasing size, from the smallest to bigger ones
    - `Newest` would indicate top files by modification date, from the newest modification date to older ones.

You will need to define the appropriate syntax for this syntax extension and write the appropriate instructions in Python to execute the scripts.

Marking criteria:

- correctness of syntax enhancements, **[3 marks]**
- evidence of testing the syntax enhancements, **[2 marks]**
- translation into Python, **[4 marks]**
- extensive testing of *fspow* scripts. **[3 marks]**

Question 4 [16 marks]

Consider the mixed-styles notation of *fspow*. It is neither an object-orientated language nor is it a procedural language purely. The language designers used a mishmash of notational styles. In answering the questions below, make sure to show the originality and authorship of the answer. Think of ways for your answers to be ungeneric and unlike material generated by AI.

Answer the following questions:

Q6.1. What in your opinion are the potential features and benefits of *fspow* versus existing languages and/or solutions? **[3 marks]**

Q6.2. Suggest ways of tweaking the syntax of *fspow* to make it a more object-oriented language. You may suggest tweaks to the syntax, justify why they are more in line with OOP style and discuss how much this would affect the complexity of translating *fspow* code. Do not write the grammar, just show what the 'improved' *fspow* code would look like and discuss as requested above. **[6 marks]**

Q6.3. A computer languages expert suggested that the target language for *fspow* should be Java because it is cross-platform. Evaluate the suggestion and provide your opinion on whether Java would be a suitable target language for *fspow*, based on the new language's features and proposed use cases. **[3 marks]**

Q6.4. What are your recommendations for a development roadmap for this new language? **[4 marks]**

Assessment will judge:

– Level of reflection and thought shown in the answers: whether the responses are superficial and lack critical thinking or if they demonstrate a high level of analysis and contemplation.

– How comprehensive and wide-ranging the answers are: whether the responses are narrow and focused on a specific aspect or if they cover a broad scope of relevant topics.

– Quality of individual answers in terms of how well-researched they are: whether the responses if they are based on personal opinions and assumptions or if they are supported by strong and reliable sources.

END