|  |  |
| --- | --- |
| **University of Aberdeen**  **School of Natural and Computing Sciences**  **Department of Computing Science**  **MSc in Artificial Intelligence**  **2019 – 2020** | |
| **Assessment Item 1 of 2 Briefing Document – Individually Assessed (no teamwork)** | |
| **Title: CS551G - Data Mining and Visualisation** | Note: This assessment accounts for 50% of your total mark of the course. |
| **Learning Outcomes**  On successful completion of this component a student will have demonstrated competence in the following areas:   * Using a non-trivial dataset, plan, execute and evaluate significant experimental investigations using multiple data mining and machine learning strategies | |
| **Information for Plagiarism:** The source code and your report may be submitted for plagiarism check (e.g., Turnitin). Please refer to the slides available at MyAberdeen for more information about avoiding plagiarism before you start working on the assessment. Please also read the following information provided by the university: <https://www.abdn.ac.uk/sls/online-resources/avoiding-plagiarism/> | |
| **Report Guidance & Requirements**  Your report must conform to the below structure and include the required content as outlined in each section. Each subtask has its own marks allocated. You must supply a written report, along with the corresponding code, containing all distinct sections/subtasks that provide a full critical and reflective account of the processes undertaken.  This assessment focuses on two very important problems that data mining experts might face in real-life situations. The first one is grouping/clustering non-annotated, i.e. data that have no corresponding labels (unsupervised learning); the second one refers to classifying time-series sequences to different classes. Time-series data usually correspond to sensor measurements and they might be captured at various time intervals and/or including trends and noise.  A detailed description of each of the **two tasks** can be found below. Please use Python for all programming tasks. You may use python-based frameworks, such as **Tensorflow and Keras**, in particular for task 2.  **Both datasets needed to fulfil the requirements of this assessment can be found in MyAberdeen.**  **Task 1: Unsupervised Learning with K-means and EM for Dog Breed Data Clustering and EDA (25/50)**  **Subtasks**:   1. Using your own words, the lecture material and any other relevant sources, please explain what is the (simple) K-Means clustering. Your description should include the following points (**3 marks**):  * Objective function * Centroids * Euclidean distance * Assignment step * Update step * K-means vs EM  1. Please provide a short description of the dataset provided, along with how you imported the data, providing snippets of code and/or detailed description **(2 marks).** 2. Employ exploratory data analysis (EDA) techniques to gain an initial understanding of the data. Please provide appropriate visualisation results and initial insights gained from EDA (**3 marks**). 3. Use K-means clustering algorithm to cluster the data provided, using two k values, i.e. **3** and **4**. The data include four different features, pertaining to various dog characteristics that correspond to a few dog breeds. Describe in detail how you deployed k-means and adjusted its parameters, going into detail on what each parameter does as well. Do the same as above but with EM this time as well. You may use open source code and libraries as long as you acknowledge them (**10 marks**). 4. Use one internal cluster validation approach to evaluate the performance of k-means and EM on the given dataset. Please present and discuss your results. How many breeds are included in this dataset, 3 or 4. Create two separate plots, a) showing the clusters created, and b) a line plot, where *x* axis is the iteration step and *y* axis is the objective function (**7 marks**)?   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **Bonus – Optional**: Should you decide to develop k-means without using any packages or pre-built libraries, there will be a bonus of **5 marks.** The maximum overall mark for this assessment still remains at 50/50; however, attempting the bonus exercise will a) make you practise more on developing algorithms on your own and b) enhance your chances of getting a higher mark overall. You can use pandas, numpy and matplotlib. You may follow the following structure:   |  | | --- | | def compute\_euclidean\_distance(vec\_1, vec\_2):  # your code comes here  return distance  def initialise\_centroids(dataset, k=3,4):  # your code comes here  return centroids  def kmeans(dataset, k=3,4):  # your code comes here  return centroids, cluster\_assigned |  * The function compute\_euclidean\_distance () calculates the distance of two vectors, e.g., Euclidean distance * The function initialise\_centroids () randomly initializes the centroids * The function kmeans () clusters the data into k groups   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **Task 2: Time Series Classification of User Movement Data (25/50)**  This dataset contains temporal data from a Wireless Sensor Network deployed in real-world office environments. The task is intended as a real-life benchmark in the area of Ambient Assisted Living. The binary classification task consists in predicting the pattern of user movements in real-world office environments from time-series generated by a Wireless Sensor Network.  **Subtasks**:   1. Please provide a comprehensive description of the dataset provided, such as how many features there are, number of classes, examples per class, etc. (**2 marks**). 2. Develop a long short-term memory (LSTM) recurrent neural network model (to refresh your memory see fig.1) to classify the time series included in the dataset provided. The number of layers, units, etc. can be chosen by yourself, but it would be prudent to be a result of experimentation with different values. A comprehensive step to achieving this is how to import and preprocess the data. Please bear in mind that the time series might have different lengths. Use the 80th quartile across all available sequences to decide the sequence length to be used. Longer sequences than the sequence length decided previously can be cropped, whereas shorter ones can be zero padded. You may develop this in keras (**10 marks**). 3. Employ a 10-fold cross validation to evaluate the performance of the developed model (**5 marks**). 4. Implement at least three different models, i.e. adjusting parameters of the LSTM, learning rate, etc. and repeat step 3. Present your results. (**5 marks**). 5. A Long Short-Term Memory (LSTM) unitConclusion: Comment and elaborate on your findings. (**3 marks**).     Figure 1: Image by Klaus Greff and colleagues as published in LSTM: A Search Space Odyssey. | |
| **Useful Information**   * Please describe and justify each step that is needed to reproduce your results by using code-snippets, screenshots and plots. When using screenshots or plots generated in Python please make sure they are clearly readable. * As the datasets provided correspond to real-life problems, the performance expected might not be as high as you might think. Therefore, as long as your implementations and justifications are correct the performance achieved will not have any effect on your marks whatsoever. * If you use open source code, you must point out where it was obtained from (even if the sources are online tutorials or blogs) and detail any modifications you have made to it in your tasks. You should mention this in both your code and report. *Failure to do so will result in zero marks being awarded on related (sub)tasks.*   **Marking Criteria**   * Quality of the report, including structure, clarity, and brevity. * Reproducibility. How easy is it for another MSc AI student to repeat your work based on your report and code? * Quality of your experiments, including design and result presentation (use of figures and tables for better reporting). * Configured to complete the task and the parameter tuning process (if needed). * In-depth analysis of the results generated, including critical evaluation, insights into data, and significant conclusions. * Quality of the source code, including the documentation of the code. | |
| **Submission Instructions**  You should submit a PDF version of your report along with your code via MyAberdeen by 23:59 on Sunday 26th April 2020. The name of the PDF file should have the form “CS551G\_Assessment1\_< your Surname>\_<your first name>\_<Your Student ID>”. For instance, “CS551G\_Assessment1\_Smith\_John\_4568985.pdf”, where 4568985 is your student ID.  You should submit your code and any associated files along with your report. If you have additional files that you wish to include then these should also be included in your submission.  If you have more than two files to submit, please compress all your files into one “zip” file (other format of compression files will not be accepted). Please try to make your submission files less than 10MB as you may have issues when uploading large files to MyAberdeen.  Any questions pertaining to any aspects of this assessment, please address them to the course coordinator Georgios Leontidis, [georgios.leontidis@abdn.ac.uk](mailto:georgios.leontidis@abdn.ac.uk). | |