****

**Faculty of Engineering, Environment and Computing**

##### 5011CEM Big Data Programming Project

**Assignment Brief**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module Title  Big Data Programming Project | Individual | | Cohort  1920JanMay | Module Code  5011CEM |
| Coursework Title (e.g. CWK1)  Coursework 1 of 1 | | | | Hand out date:  20/01/20 |
| Lecturer:  Richard Hyde: ad2105@coventry.ac.uk  Norlaily Yaacob: csx214@coventry.ac.uk  Mark Johnston: ad4039@coventry.ac.uk | | | | Due date and time:  Moodle: 24/04/20 at 17:00  Physical: n/a |
| Estimated Time (hrs):  Word Limit\*: 2,000 | | Coursework type:  Written Report | | % of Module Mark  34% |
| Submission arrangement online via CUMoodle  File types and method of recording: n/a  Mark and Feedback date: 13/05/20  Mark and Feedback method (e.g. in lecture, written via Gradebook): Moodle | | | | |

|  |
| --- |
| Module Learning Outcomes Assessed are the following Course Learning Outcomes:  B4: DATA SCIENCE: work with (potentially large) datasets; using appropriate storage technology; applying statistical analysis to draw meaningful conclusions; and using modern machine learning tools to discover hidden patterns.  B7: TRANSFERABLE SKILLS: apply a wide variety of degree level transferable skills including time management, team working, written and verbal presentation to both experts and non-experts, and critical reflection on own and others work. |
| You are required to submit a report on your project work. You will be provided with a basic template that indicates the minimum content, further content can be added as required.  The required sections include:   1. Specification (25%). This is a detailed specification of what your project is expected to achieve. These should be an enumerated list of SMART targets. Ensure that your specification is achievable in the time available. 2. Introduction (25%). This should give a brief overview of the whole project followed by an outline of the sub-project you have chosen to work on. Describe how your sub-project will contribute to the overall project. 3. Code (25%):    1. Code Description. This is not a detailed line-by-line description of your code. This should describe the overall methods your code employs to achieve your specification. You should specifically mention how you have employed ‘big data’ techniques in your code considering that you have been working with a small sub-set of code and the overall project is required to work with much larger data-sets.    2. Instructions on how to install and run your code. This should include details of any software, IDE, libraries etc that are required. A new user, unfamiliar with your coding environment should be able to install a suitable environment, run your code­ and view / find all outputs.    3. Upload an accompanying \*.zip file to Moodle containing all your code. 4. Summary / Conclusions (25%). Refer back to your specification, describe how each item has been achieved. If any parts of your specification have not been achieved, explain why and suggest how future work may solve these issues.   This assessment is graded out of 100, and contributes 34% of the module grade. |
| Notes:   1. You are expected to use the [Coventry University Harvard Referencing Style](https://curve.coventry.ac.uk/open/file/bdfb947c-9d43-48d3-8ec8-f511682e1dd1/1/The%20CU%20Guide%20to%20Referencing%20in%20Harvard%20Style.pdf). For support and advice on this students can contact [Centre for Academic Writing (CAW)](http://www.coventry.ac.uk/study-at-coventry/student-support/academic-support/centre-for-academic-writing/?theme=main). 2. Please notify your registry course support team and module leader for disability support. 3. Any student requiring an extension or deferral should follow the university process as outlined [here](https://share.coventry.ac.uk/students/Registry/Pages/Deferrals-and-Extension.aspx). 4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system. 5. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via your Module Leader. 6. Assignments that are more than 10% over the word limit will result in a deduction of 10% of the mark i.e. a mark of 60% will lead to a reduction of 6% to 54%. The word limit includes quotations, but excludes the bibliography, reference list and tables. 7. You are encouraged to check the originality of your work by using the draft Turnitin links on your Moodle Web. 8. Collusion between students (where sections of your work are similar to the work submitted by other students in this or previous module cohorts) is taken extremely seriously and will be reported to the academic conduct panel. This applies to both courseworks and exam answers. 9. A marked difference between your writing style, knowledge and skill level demonstrated in class discussion, any test conditions and that demonstrated in a coursework assignment may result in you having to undertake a Viva Voce in order to prove the coursework assignment is entirely your own work. 10. If you make use of the services of a proof reader in your work you must keep your original version and make it available as a demonstration of your written efforts. 11. You must not submit work for assessment that you have already submitted (partially or in full), either for your current course or for another qualification of this university, unless this is specifically provided for in your assignment brief or specific course or module information. Where earlier work by you is citable, ie. it has already been published/submitted, you must reference it clearly. Identical pieces of work submitted concurrently will also be considered to be self-plagiarism. |

**Mark allocation guidelines to students**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0-39 | 40-49 | 50-59 | 60-69 | 70+ | 80+ |
| Specification | Few items, poorly worded and vague | Vague targets, easily misinterpreted, not SMART | Generally good targets, although vague, open to misinterpretation and not fully SMART | Good targets, generally SMART, some vagueness or potential for misinterpretation | Generally excellent, concise targets. Generally SMART, some weaknesses | Excellent, concise targets. Fully SMART |
| Introduction | Little description of overall project or sub-project and how it contributes | Some description of overall project, little detail of sub-project or its contribution. | Fair description of overall project, sub-project described, but not very clear how it contributes overall. | Good description of overall project and sub-project, mention of how it contributes but weaknesses in vague in some areas | Good description of overall project and sub-project. Clear and concise explanation of contribution. | Excellent description of overall project, sub-project and clear explanation of contribution |
| Code | Poor code, poor annotations and variable names, poor use of functions or classes, unable to run or information missing.  Poor output. | Code runs with some additional info or knowledge, poor annotations and variable names, poor use of functions or classes.  Outputs unclear or not present. | Code runs, but may need some tweaking or additional knowledge, annotations present but may not be clear. Variable names mostly meaningful. Some use of function or classes.  Some outputs, but not always clear. | Code runs easily, all install info present but may be unclear. Annotations fair so it is possible to interpret the code operation. Variable names generally good. Functions or classes are well structured and clear.  Outputs clear and useful. | Code runs easily, all install info present to install and run. Annotations good so it is possible to interpret the code operation. Variable names good and consistent. Functions or classes are well structured and clear.  Outputs clear with some titles and labels. | Excellent code structure, clear variable names. Annotations make the code functionality clear and obvious. Excellent use of functions or classes. All outputs clearly titled and labelled. |
| Summary / Conclusion | Little mention of project spec, no analysis of success or failure | Some mention of project spec, basic mention of success or failure | Project spec addressed, but little indication of success or failure. No mention of solutions for failed items. | Project spec addressed, mentions success or failure but poor explanation of how or why. Failures not addressed. | All items in project spec addressed, Success or failure mentioned with some description of how & why. Failures addressed with some indication of how they could be addressed. | Each item in project spec specifically addressed with clear explanation of how and why each item has been a success or failed. All failures addressed with future work proposed to solve. |