Tasks 2020

Machine Learning and Statistics

Due: last commit on or before January 1st, 2021

These are the instructions for the Tasks assessment for Machine Learning and Statistics in 2020. The assessment is worth 50% of the marks for the module. Please read the *Using git for assessments* [3] document on the Moodle page which applies here. As always, you must also follow the code of student conduct and the policy on plagiarism [2].

Instructions

Four tasks will be listed here at different times during the semester. You should complete all tasks in a single jupyter notebook. This, along with relevant files like a README, should be in a single git repository synced with a hosting provider like GitHub [1]. That URL should then be submitted using the link on the Moodle page.

1. **October 5th, 2020:** Write a Python function called `sqrt2` that calculates and prints to the screen the square root of 2 to 100 decimal places. Your code should not depend on any module from the standard library[1] or otherwise. You should research the task first and include references and a description of your algorithm.

2. **November 2nd, 2020:** The Chi-squared test for independence is a statistical hypothesis test like a t-test. It is used to analyse whether two categorical variables are independent. The Wikipedia article gives the table below as an example [4], stating the Chi-squared value based on it is approximately 24.6. Use `scipy.stats` to verify this value and calculate the associated p value. You should include a short note with references justifying your analysis in a markdown cell.

	A	B	C	D	Total
White collar	90	60	104	95	349
Blue collar	30	50	51	20	151
No collar	30	40	45	35	150
Total	150	150	200	150	650

[1] By the standard library, we mean the modules and packages that come as standard with Python. Anything built-in that can be used without an `import` statement can be used.

3. **November 16th, 2020:** The standard deviation of an array of numbers `x` is calculated using `numpy` as `np.sqrt(np.sum((x - np.mean(x))**2)/len(x))`. However, Microsoft Excel has two different versions of the standard deviation calculation, `STDEV.P` and `STDEV.S`. The `STDEV.P` function performs the above calculation but in the `STDEV.S` calculation the division is by `len(x)-1` rather than `len(x)`. Research these Excel functions, writing a note in a Markdown cell about the difference between them. Then use `numpy` to perform a simulation demonstrating that the `STDEV.S` calculation is a better estimate for the standard deviation of a population when performed on a sample. Note that part of this task is to figure out the terminology in the previous sentence.

4. **November 30th, 2020: NB – when I first posted this task, I accidentally wrote "k-means" where I meant to write "kNN" for k Nearest Neighbours. Because of this, I will allow either algorithm to be used and have extended the deadline by two weeks.** Use `scikit-learn` to apply k Nearest Neighbours clustering to Fisher's famous Iris data set. You will easily obtain a copy of the data set online. Explain in a Markdown cell how your code works and how accurate it might be, and then explain how your model could be used to make predictions of species of iris.

Marking scheme

The following marking scheme will be used to mark your submission out of 100%, which will then be scaled to 50%. The examiner's overall impression of your submission may influence marks in each individual component. It is important that your submission provides direct evidence of each of the items listed in each category. For instance, your commit history should demonstrate and provide evidence that you had a pragmatic attitude to completing the assessment. Likewise, your submission should have references in it to demonstrate that you considered the literature and the work of others.

25%	**Research**	Evidence of research performed on topic; submission based on referenced literature, particularly academic literature; evidence of understanding of the documentation for any software or libraries used.
25%	**Development**	Environment can be set up as described; code works without tweaking and as described; code is efficient, clean, and clear; evidence of consideration of standards and conventions appropriate to code of this kind.
25%	**Consistency**	Evidence of planning and project management; pragmatic attitude to work as evidenced by well-considered commit history; commits are of a reasonable size; consideration of how commit history will be perceived by others.
25%	**Documentation**	Clear documentation of how to create an environment in which any code will run, how to prepare the code for running, how to run the code including setting any options or flags, and what to expect upon running the code. Concise descriptions of code in comments and README.

References

[1] GitHub Inc., "GitHub," https://github.com/.

[2] GMIT, "Quality Assurance Framework," https://www.gmit.ie/general/quality-assurance-framework.

[3] I. McLoughlin, "Using git for assessments," https://github.com/ianmcloughlin/using-git-for-assessments/.

[4] Wikipedia contributors, "Chi-squared test — Wikipedia, the free encyclopedia," 2020, [Online; accessed 1-November-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Chi-squared_test&oldid=983024096