$SIT742$ Modern Data Science

Unit Assessment Handbook

Gang Li

Trimester 1, 2019

School of Information Technology
Deakin University, Australia

CONTENTS

CHAPTER
ZERO

ASSESSMENT OVERVIEW

Contents

0.1 General Information

2019 $SIT742$ assessment consists of 4 components as detailed below:

Table 0.1: $SIT742$ Assessment Plan

Components	Percentage	Submission Methods	Working Mode
Task 1	25%	Critical Analysis and Report	Individual
Task 2	40%	Project Work	Group
Task 3	05%	Online Quiz in *CloudDeakin*	Individual
Final Examination	30%	Exam	Individual

0.1.1 Where to Get Help?

Students are encouraged to discuss the unit related assessment in *CloudDeakin* unit [**home page**] → [**Discussions**] → [**Student Discussion**].

If you are trying to form a project group for Assessment Task 2, you can post messages at: [**home page**] → [**Discussions**] → [**Finding a project group?**].

Any enquiry about this unit can be posted at: [**home page**] → [**Discussions**] → [**Questions for the Unit Chair**].

0.1.2 Where to Submit?

All tasks should be submitted to *CloudDeakin* by following corresponding specific requirements. You can find the corresponding submission boxes for Task 1 and 2 by following the [**home page**]→ [**Assessment**] → [**Assignments**] link.

0.1.3 Important Dates

Please be aware of the following important dates:

Task 1 **Due Date** *CloudDeakin* Submission, by **23:59pm,** 06/04/2019 (**Week** 05 **Saturday**).

Task 2 **Group Sign-Up Due Date** Task 2 group sign-up on *CloudDeakin* by **23:59pm,** 13/04/2019 (**Week** 06 **Saturday**).

Task 2 **Due Date** *CloudDeakin* Submission, by **23:59pm,** 18/05/2019 (**Week** 10 **Saturday**).

Task 3 **Due Date** *CloudDeakin* Online Quiz, by Week 10.

0.1.4 Assignment Results

Task 1 and 2 will be marked with report, while task 3 will be automatically marked.

- The marking report is expected to be released to CloudDeakin within 7 working days of the due date;
- Within 3 working days after the result is released, any student who wishes to challenge their mark must contact or approach the unit chair during contact hours and bring with them a copy of their assignment and their mark breakdown. Cloud students can email me for this issue.

0.2 General Requirements

Student information and assessment related forms can be found from this URL: `http://www.deakin.edu.au/sebe/students/`. Students are required to familiarize with Faculty regulations regarding plagiarism.

1. Any text or code adapted from any source must be clearly labelled and referenced. You should clearly indicate the start and end of any such text/code.
2. **All SIT742 assignments must be submitted as required by their corresponding assessment specifications**. Assignments will not be accepted through any other manner without prior approval. Students should note that this means that email based submissions will ordinarily be rejected.

3. Penalties for late submissions are indicated in the `Unit Guide`. Close of submissions on the due date and each day thereafter for penalties will occur at $11:59$pm local time. Students outside of Victoria, Australia, should note that the local time zone is $UTC + 10$, and in Daylight Saving Dates, it will be $UTC + 11$.

4. Information regarding assignment extensions is provided in the `Unit Guide & Information` in CloudDeakin. Students must not assume an extension will be granted. Late penalties still apply in the case of a failed application for extension. Thus until an extension is granted students should submit any work completed before the assignment is due. Note that extensions cannot be granted for system outages or encumbrances.

ASSESSMENT TASK

ONE

DATA EXPLORATION: WINE RATING DATA

Contents

This task contributes 25% of your final *SIT*742 mark. It must be completed individually, and submitted to *CloudDeakin* by **23:59pm,** 06/04/2019 **(Week** 05 **Saturday)**.

1.1 Background

Here is the hypothetical background:

> *Hotel TULIP* (a hypothetical organisation) is a five-star hotel that locates in Deakin University, Australia. It is a very special hotel with an equally special purpose: Not only does it embody all the creative energy and spirit of Deakin University, it's a "learning environment" on which the tourism and hospitality students are trained for future hoteliers.
>
> Recently, *Hotel TULIP* would like to review the menu and tempt any guest with delightful wines. The hotel's CIO, Dr *Bear Guts* (not *Bill Gates*!), believes that ratings provided by the *WineEnthusiast* (a multichannel marketer of a growing line of wine) are great resources to help their *Market Promotion Division* to identify potential excellent wine with affordable price. Hence, *Hotel TULIP* would like to outsource the web usage mining task to *Team-SIT742* (a hypothetical data analytics company) to analyze a wine rating dataset and discover taste and price patterns of different types of wine over the world.

The raw data was scraped from *WineEnthusiast* in `json` format. The original code for the scraper can be found at: `https://github.com/zackthoutt/wine-deep-learning`.

1.2 Task Description

We provide one `IPython` notebook `SIT742Task1.ipynb` at `https://github.com/tulip-lab/sit742/tree/master/Assessment/2019`, together with two data files at the `data` subfolder:

`wine.json` The `json` file contains the wine ratings and reviews from *WineEnthusiast*.

`stopwords.txt` This text file contains the most common English stop words.

You are required to develop a data exploration report using `IPython` notebook to complete the following two sub-tasks.

1.2.1 Numeric and Categorical Value Analysis [60% scores]

For a data scientist, after obtaining the dataset, the first most crucial task is to obtain a good understanding of the data he or she is dealing with. This includes: examining the data attributes (or equivalently, data fields), seeing what they look like, what is the data type for each field, and from this information, determining suitable numerical/visual descriptions.

The first task is to read the `json` file as a **Pandas DataFrame** and delete the rows which contain invalid values in the attributes of "`points`" and "`price`".

Then, you need to answer the following two questions in your `IPython` notebook based on this dataset:

(1) what are the 10 varieties of wine which receives the highest number of reviews?

(2) which varieties of wine having the average price less than 20, with the average points at least 90? Assuming there is no duplicate review in the data, i.e., each row represent a unique wine.

In addition, you need to group all reviews by different countries and generate a statistic table, and save as a `csv` file named "**statisticByState.csv**". The table must have four columns:

Country – listing the unique country name.

Variety – listing the varieties receiving the most reviews in that country.

AvgPoint – listing the average point (rounded to 2 decimal places) of wine in that country

AvgPrice – listing the average price (rounded to 2 decimal places) of wine in that country

Based on this table, which country/countries would you recommend *Hotel TULIP* to source wine from? Please state your reasons.

1.2.2 Text analysis [40% scores]

In this task, you are required to write Python code to extract keywords from the "`description`" column of the `json` data, used to redesign the wine menu for Hotel TULIP.

You need to generate two `txt` files:

`HighFreq.txt` This file contains the frequent unigrams that appear **in more than** 5000 **reviews** (one row in the dataframe is one review).

`Shirazkey.txt` This file contains the key unigrams with `tf-idf` score higher than 0.4. To reduce the runtime, first you need to extract the description from the variety of "`Shiraz`", and then calculate `tf-idf` score for the unigrams in these descriptions only.

In both `txt` files, all unigrams are sorted alphabetically and are saved line by line without duplicate. Before you calculate the unigram frequent or `tf-idf`, you need to remove the stop words in all description using the provided "`stopwords.txt`" or using the built-in function in Python.

1.3 What to Submit?

Please familiarise yourself with the *General Requirements* (see Section 0.2) on Assignments Submission. By the due date, you are required to submit the following files to the corresponding *Assignment* (Dropbox) in CloudDeakin:

`SIT742Task1.ipynb` Your `IPython` notebook solution source file for the data exploration of the wine rating data. You can fill your name and deakin ID information at the relevant place in the first markdown cell. Please follow the `PEP 8` guidelines (Section 3.1) for source code style.

`statisticByState.csv`

`Shirazkey.txt`

`HighFreq.txt`

No Special Consideration will be granted for this project. Students who have difficulty meeting the deadline because of illness, etc. must apply for an assignment extension no later than the noon on the day prior to the deadline.

ASSESSMENT TASK

TWO

DATA ANALYTICS: BANK MARKETING

Contents

This task contributes 40% of your final *SIT*742 mark. It can be done in group of 3 members and submitted to *CloudDeakin* by **23:59pm,** 18/05/2019 **(Week** 10 **Saturday)**.

2.1 Background

In this assignment, you will analyse an open dataset about a marketing campaign of a Portuguese bank in order to design strategies for improving future marketing campaigns. The object of this campaign is to pursuit customers to subscribe the term deposit. The marketing campaigns were based on phone calls. The dataset contains the call information with the following attributes in Table 2.1.

More information about this dataset can be found at: `https://archive.ics.uci.edu/ml/datasets/bank+marketing`.

2.2 Task Description

We provide one `IPython` notebook `SIT742Task2.ipynb` at `https://github.com/tulip-lab/sit742/tree/master/Assessment/2019`, together with a `csv` file **bank.csv** at the `data` subfolder. You are required to analyse this dataset using `IPython` notebook with **Spark** packages including **spark.sql** and **pyspark.ml** that you have learnt from SIT742.

Table 2.1: Attribute information of the dataset

Attribute	Meaning
`age`	age of the customer
`job`	type of job
`marital`	marital status
`education`	education level
`default`	has credit in default?
`balance`	the balance of the customer
`housing`	has housing loan?
`loan`	has personal loan?
`contact`	contact communication type
`day`	last contact day of the week
`month`	last contact month of year
`duration`	last contact duration, in seconds
`campaign`	number of contacts performed
`pdays`	number of days that passed by after a previous campaign
`previous`	number of contacts performed before this campaign
`poutcome`	outcome of the previous marketing campaign
deposit	has the client subscribed a term deposit?

2.2.1 `IPython` Notebook [60% scores]

To systematically investigate this dataset, your `IPython` notebook should follow the basic 6 procedures as:

(1) Import the csv file, "`bank.csv`", as a **Spark dataframe** and name it as `df`, then check and explore its individual attribute.

(2) Select important attributes from `df` as a new dataframe `df2` for further investigate. You are required to select 13 important attributes from df: `` `age' ``, `` `job' ``, `` `marital' ``, `` `education' ``, `` `default' ``, `` `balance' ``, `` `housing' ``, `` `loan' ``, `` `campaign' ``, `` `pdays' ``, `` `previous' ``, `` `poutcome' `` and `'deposit'`.

(3) Remove all invalid rows in the dataframe `df2` using **spark.sql**. Supposing that a row is invalid if at least one of its attributes contains `` `unknown' ``. For the attribute `` `poutcome' ``, the valid values are `` `failure' `` and `` `success' ``.

(4) Convert all categorical attributes to numerical attributes in `df2` using *One hot encoding*, then apply *Min-Max normalisation* on each attribute.

(5) Perform unsupervised learning on `df2` including *k-means* and *PCA*. For *k-means*, you can use the whole `df2` as both training and testing data, and evaluate the clustering result using *Accuracy*. For *PCA*, you can generate a scatter plot using the first two components to investigate the data distribution.

(6) Perform supervised learning on `df2` including *Logistic Regression*, *Decision Tree* and *Naive Bayes*. For the three classification methods, you can use 70% of `df2` as the training data and the remaining 30% as the testing data, and evaluate their prediction performance using *Accuracy*.

2.2.2 Case Study Report [40% scores]

Based on your `IPython` notebook results, you are required to write a case study report with 500 − 1000 words, which should include the following information:

(1) The data attribute distribution

(2) The methods/algorithms you used for data wrangling and processing

(3) The performance of both unsupervised and supervised learning on the data

(4) The important features which affect the objective ('`yes`' in '`deposit`') [Hint: you can refer the coefficients generated from the *Logistic Regression*]

(5) Discuss the possible reasons for obtaining these analysis results and how to improve them

(6) Describe the group activities, such as the task distribution for group members and what you have learnt during this project.

More information about report writing can be found at: `https://www.deakin.edu.au/students/studying/study-support/academic-skills/report-writing`.

2.2.3 Important Dates

Please be aware of the following important dates:

Group Sign-Up The group needs to be finalized on *CloudDeakin* by **23:59pm,** 13/04/2019 **(Week** 06 **Saturday)**. If any issue or group correction is needed, please send SIT742 unit chair an email by **23:59pm,** 13/04/2019 **(Week** 06 **Saturday)**.

Final Submission The due date for this package submission is on **23:59pm,** 18/05/2019 **(Week** 10 **Saturday)**.

2.3 What to Submit?

Please familiarise yourself with the *General Requirements* (see Section 0.2) on Assignments Submission. By the due date, you are required to submit the following files to the corresponding *Assignment* (Dropbox) in CloudDeakin:

Group Sign-up done on *CloudDeakin* by **23:59pm,** 13/04/2019 **(Week** 06 **Saturday)**.

`SIT742Task2.ipynb` Your `IPython` notebook solution source file for the data exploration of the bank marketing data. You can fill your group information at the relevant place in the first markdown cell. Please follow the `PEP 8` guidelines (Section 3.1) for source code style.

`Report.pdf` A 500 – 1000 words report describing and discussing your analysis results.

No Special Consideration will be granted for this project. Students who have difficulty meeting the deadline because of illness, etc. must apply for an assignment extension no later than the noon on the day prior to the deadline.

APPENDIX

THREE

APPENDIX

Contents

3.1 Code Style: `Pep 8`

`Pep 8` is the de-facto code style guide for *Python* (`https://www.python.org/dev/peps/pep-0008/`). Skim the style guide to gain basic understanding of what is required. Conforming your *Python* code to `PEP 8` is generally a good idea and helps make the code more consistent when working on projects with other developers.

In your assessment task, if the source code or `IPython` notebook is to be included, you are required to format your code so that it meets at least the following major `PEP 8` guidelines:

Comment Please follow the following style for *Python* comments:

1. To explain the functionality of a group of statements, apply block comments before the statements. Indent the comments to the same level as the code.
2. Write documentation strings (i.e. `docstring`) for your function.

Code Lay-out Please follow the following style for *Python* code layout:

1. Blank lines: Surround top-level function and class definition with two blank lines. Use blank lines in functions, sparingly, to indicate logical sections.
2. Indentation: Use four white spaces instead of tab for indentation.

White spaces in expressions and statements Please follow the following style for *Python* while spaces:

1. Surround binary operators with a single space on either side.
2. If operators with different priorities are used, consider add whitespace around the operators with the lowest priority(ies). However, never use more than one space.

You should use:

```
i = i + 1
num += 1
x = x*2 - 1
```

rather than this:

```
i=i+1
num +=1
x = x * 2 - 1
```

String quotes Use either single-quoted or double-quoted strings. Pick one of them and stick to it for consistency. Only use the other one when a string contains single or double quote characters.

Naming Conventions Make sure the naming of your variable follow consistent style: e.g. `lowercase`, `lower_case_with_underscores`, or `mixedCase`.

3.2 Academic Skills

3.2.1 How to Find Papers?

For the assessment task in this unit, you can try to find some related references from highly respected journals and conferences. You can find papers from Scopus, IEEEXplore, ACM Portal, Elsevier ScienceDirect, and DBLP [1].

Search engines like Google, Scopus and CiteSeer are widely being used to find papers, though they do not have a warrant for paper quality. Beside above search method, there are some other tricks which can help you to find the most suitable paper:

People/Group Oriented You could first identify important people/groups in that sub-domain by MS academic, and then find more related papers from their website.

Citation Oriented You could use search engine, like Google Scholar, to find papers with most citations in that sub-domain. In this way, you can find many classical and influential (but maybe not up-to-date) papers in this sub-domain.

[1] DBLP only provides paper titles and sources, you need to download the paper from somewhere else.

Top Conference/Top Journal Oriented You could find a lot of up-to-date papers on your topic from the top conferences or top journals. This would help you understand the state of art of your selected topic. To see which conferences or journals are top ones, you may refer to MS academic, or Google Scholar.

3.2.2 How to Read Papers?

There are some *Research skills* articles available on *How to Read*:

- Michael J. Hanson and Dylan J. McNamee. *Efficient Reading of Papers in Science and Technology*

In general it is unnecessary to understand all the details of all papers you collected, let alone that it is difficult to understand all. When you are reading a paper, you'd better keep in mind to answer following problems:

1. What is the problem concerned and why this problem is important?
2. How is the problem solved, completely solved or partially solved?
3. Does it have any relationship with other papers you have read?

3.2.3 How to Write a Paper?

There are some *Research skills* articles available on *How to Write*:

- Mike Ashby. *How to Write a Paper*

A first-time academic author usually lists everything he collected in a survey. A better way is to find a clue from motivations of different works: What's the goal and what's the problem? What's the first step, what problems it solved and what problems remained? So came the second step...Following your clue to answer these kinds of questions.

More Tips

1. When you are writing your report, please assume that your readers know nothing about your topic.
2. When you are writing your report, please keep your mind clear. It is better to first write down a outline.
3. After finishing your report, please check out your language and logic.

No Plagiarism You should be cautious about the writing. The *TurnItIn* system will be used for all SIT742 assignment submissions. Whenever you are using words and works of others, citations should be made clear such that one can tell which part is actually yours. IEEE Document "Introduction to the Guidelines for Handling Plagiarism Complaints" provides details about how IEEE will identify and handle a plagiarism.