School of Computing and Information Systems
The University of Melbourne
COMP90049, Introduction to Machine Learning, Semester 1 2023

Assignment 3: Job Salary Prediction

Released: Monday, April 17th 2023.

Due: **Stage I**: Friday, May 19th 5pm
Stage II: Wednesday, May 24th 5pm

Marks: The Project will be marked out of 30, and will contribute 30% of your total mark.

1 Overview

In this assignment, you will develop and critically analyse models for predicting the **the salary of jobs**. That is, given a job description, your model(s) will predict the salary offered for the job. You will be provided with a data set of job descriptions that have been labelled with their salary. In addition, each job description is labelled with the gender balance of the occupation class it falls under: *balanced male-female (0)* or *male-dominated (1)* or *female-dominated (2)*. For example, 'engineering' occupations tend to be male-dominated, while education-related occupations are more often taken up by more women than men. You may use this additional information to investigate whether your models work equally well across occupations with a different gender skew. The assessment provides you with an opportunity to reflect on concepts in machine learning in the context of an open-ended research problem, and to strengthen your skills in data analysis and problem solving.

The goal of this assignment is to **critically assess** the effectiveness of various Machine Learning algorithms on the problem of determining a job's salary, and to **express the knowledge that you have gained in a technical report**. The technical side of this project will involve applying appropriate machine learning algorithms to the data to solve the task. There will be a Kaggle in-class competition where you can compare the performance of your algorithms against your classmates.

The focus of the project will be the report, formatted as a short research paper. In the report, you will demonstrate the knowledge that you have gained, in a manner that is accessible to a reasonably informed reader.

2 Deliverables

Stage I: Model development and testing and report writing (due May 19):

1. One or more programs, written in Python, including all the code necessary to reproduce the results in your report (including model implementation, label prediction, and evaluation). You should also include a README file that briefly details your implementation. *Submitted through Canvas.*

2. An anonymous written report, of 2000 words ($\pm 10\%$) **excluding** reference list. Your name and student ID should **not** appear anywhere in the report, including the metadata (filename, etc.). *Submitted through Canvas/Turnitin.* **You must upload the report as a separate PDF file**. Do **NOT** upload it as part of a compressed archive (zip, tar, ...) file or in a different format.
3. Predictions for the test set of job descriptions submitted to the Kaggle[1] in-class competition described in Sec 7.

Stage II: Peer reviews (due May 24th):

1. Reviews of two reports written by your classmates, of 200-400 words each. *Submitted through Canvas.*

3 Data Sets

You will be provided with

- A *training* set of 13,902 job descriptions. The first 8,000 descriptions are labeled with the job's salary (target label) and gender balance of the job's occupation group (demographic label). The remaining 5,902 descriptions are *unlabelled*. You may use these for semi- or unsupervised learning approaches.

val set
- A *development* set of 1,738 labeled job descriptions, with target and demographic labels which you should use for model selection and tuning;

- A *test* set of 1,737 job descriptions, with no target (but demographic) labels, which will be used for final evaluation in the Kaggle in-class competition

3.1 Target Labels

These are the labels that your model should predict (y). We provide this label in two forms:

- the mean expected salary (float; in the column named `mean_salary` in the `raw` data *.csv files); and

- a categorical label indicating the salary band, where we binned the mean salaries into 10 equal-frequency bins (in the column named `salary_bin` in the raw data *.csv files).

You may use either label representation in your experiments, but different representations might call for different machine learning approaches.

3.2 Demographic Labels

Demographic labels provide additional meta information about the gender skew in the occupation category of a job ad (in the column named `gender_code` in the raw data *.csv files). They should *only* be used to evaluate models on specific subgroups of employees (male vs female), but *not* be predicted (and probably not used as features, although you can discuss this in your report). In the provided data set, each job ad is labelled with one of three possible demographic labels indicating the gender balance of the job's occupation category:

- 0: gender-balanced occupation category (e.g., consultants, accountants, ...)

[1] https://www.kaggle.com/

- 1: female-dominated occupation category (e.g., nurse, social workers, . . .)
- 2: male-dominated occupation category (e.g., engineers, construction workers, . . .)

3.3 Features

To aid in your initial experiments, we have created different **feature representations** from the raw job descriptions. You may use any subset of the representations described below in your experiments, and you may also engineer your own features from the raw descriptions if you wish. The provided representations are

I. Raw The raw descriptions represented as a single string. We lowercased all words, and removed punctionation. E.g.,

"develop implement evaluate health nutritional programmes develop interactive health programme content activities"

The job description in plain text is provided in the column `requirements_and_roles` in the raw data *.csv files).

II. TFIDF We applied term frequency - inverse document frequency pre-processing to the job ads for feature selection. In particular, we (1) removed all stopwords and (2) only retained the 500 words in the full raw job description data set with highest TFIDF values. As a result, each job ad is now represented as a 500 dimensional feature vector, each dimension corresponding to one of the 500 words. The value is 0 if the word did *not* occur in the job description, and the word's TFIDF score if the word occurs in the description. Note that most values will be 0.0 as job descriptions are short. E.g.,

[0.0, 0.0, 0.0, 0.0, . . . 0.998, . . . 0.0]

A 500-dimensional list of numbers

TFIDF score of word in job description

Word not in job description

The Feature Selection lecture and associated Code (in week 5) provides more information on TFIDF, as well as Schütze et al. (2008).

The file `tfidf_words.txt` contains the 500 words with highest TFIDF value, as well as their index in the vector representations. You may use this information for model/error analysis.

III. Embedding We mapped each job description to a 384-dimensional embedding computed with a pre-trained language model, called the Sentence Transformer (Reimers and Gurevych, 2019).[2] These vectors capture the "meaning" of each job ad so that similar job ads will be located closely together in the 384-dimensional space. E.g.,

[2.05549970e-02 8.67250003e-02 8.83460036e-02 -1.26217505e-01 1.31394998e-02 . . .]

a 384-dimensional list of numbers

[2] https://pypi.org/project/sentence-transformers/

Data format The **Raw** data are provided in csv format (train.csv, valid.csv and test.csv). The labeled data sets also contain both target label formats (real value and bin) and the demographic label, with the following columns:

`job_id` unique identifier for each instance
`requirements_and_role` input features (raw job description text)
`salary_bin` target label (binned categorical)
`mean_salary` target label (float)
`gender_code` demographic label (categorical)

The **TFIDF** and **Embeddings** representations are provided as dense numpy matrix (files ending *.npy).[3]. Line numbers for the same data set type refer to the same instance, e.g., line 5 in train.csv, train-embeddings.npy and train-tfidf.npy are different representations of the same job ad.

4 Project Stage I

You should formulate a research question (two examples provided below), and develop machine learning algorithms and appropriate evaluation strategies to address the research question.

You should *minimally* implement and analyse in your report **one baseline**, and **at least two different** machine learning models. **N.B.** We are more interested in your **critical analysis** of methods and results, than the raw performance of your models. You may not be able to arrive at a definitive answer to your research question, which is perfectly fine. However, you should analyse and discuss your (possibly negative) results in depth.

4.1 Research Question

You should address **one** research question in your project. We propose two research questions below, for inspiration but you may propose your own. Your report should clearly state your research question. Addressing more than one research question does **not** lead to higher marks. We are more interested in your *critical analysis* of methods and results, than the coverage of more content or materials.

Research Question 1: Does Unlabelled data improve Job salary prediction?

Various machine learning techniques have been (or will be) discussed in this subject (Naive Bayes, 0-R, clustering, semi-supervised learning); many more exist. These algorithms require different levels of supervisions: some are supervised, some unsupervised and some combine both strategies. Develop machine learning models that leverage different amounts of supervision, using `labeled` an `unlabeled` portions of the train data set. You may also want to experiment with different feature representations (Sec 3.3). Alternatively, you may want to develop meaningful approaches for predicting mean salaries vs. salary bins (different y-label representations). You are strongly encouraged to make use of machine learning software and/or existing libraries in your attempts at this project (such as `sklearn` or `scipy`). What are the strengths and weaknesses of the different machine learning paradigms? Can you effectively combine labelled and unlabelled training data?

[3] Learn here how to read and process these files: `https://numpy.org/doc/stable/reference/generated/numpy.load.html`.

Research Question 2: Exploring Bias in Job salary prediction

Compare different models and/or feature representations in their performance of on the different demographic groups in the data set (balanced vs. female-dominated vs. male-dominated jobs) separately. Some models will *not* work equally well for all these groups. Critically analyse the gap, and try to explain it in the context of the concepts covered in this subject. Can you adapt your models to close the performance gap? How? *Note:* your grade does not depend on your success in closing the performance gap. Interestingly, failed attempts with in-depth analyses are perfectly acceptable.

4.2 Feature Engineering (optional)

We have discussed three types of attributes in this subject: categorical, ordinal, and numerical. All three types can be constructed for the given data. Some machine learning architectures prefer numerical attributes (e.g. k-NN); some work better with categorical attributes (e.g. multivariate Naive Bayes) – you will probably observe this through your experiments.

It is **optional** for you to engineer some attributes based on the `raw` job description dataset (and possibly use them instead of – or along with – the feature representations provided by us). Or, you may simply select features from the ones we generated for you (tfidf, and embedding).

4.3 Evaluation

The objective of your learners will be to predict the labels of unseen data. We will use a **holdout strategy**. The data collection has been split into three parts: a training set, a development set, and a test set. This data is available on the LMS.

To give you the possibility of evaluating your models on the test set, we will be setting up a **Kaggle In-Class competition**. You can submit results on the test set there, and get immediate feedback on your system's performance. There is a Leaderboard, that will allow you to see how well you are doing as compared to other classmates participating on-line.

4.4 Report

You will submit an **anonymised** report of 2000 words in length ($\pm 10\%$), **excluding** reference list. The report should follow the structure of a short research paper, as discussed in the guest lecture on Academic Writing. It should describe your approach and observations in the context of your chosen research question, both in engineering (optional) features, and the machine learning algorithms you tried. Its main aim is to provide the reader with **knowledge** about the problem, in particular, **critical analysis of your results and discoveries**. The internal structure of well-known machine learning models should only be discussed if it is important for connecting the theory to your practical observations.

- Introduction: a short description of the problem and data set, and the research question addressed
- Literature review: a short summary of some related literature, including the data set reference and at least two additional relevant research papers of your choice. You might find inspiration in the Reference list of this document. You are encouraged to search for other references, for example among the articles cited within the papers referenced in this document.

- Method: Identify the newly engineered feature(s), and the rationale behind including them (Optional). Explain the ML models and evaluation metric(s) you have used (and why you have used them)
- Results: Present the results, in terms of evaluation metric(s) and, ideally, illustrative examples. Use of tables and diagrams is highly recommended.
- Discussion / Critical Analysis: Contextualise** the system's behavior, based on the understanding from the subject materials as well as in the context of the research question.
- Conclusion: Clearly demonstrate your identified knowledge about the problem
- A bibliography, which includes Bhola et al. (2020), as well as references to any other related work you used in your project. You are encouraged to use the APA 7 citation style, but may use different styles *as long as you are consistent* throughout your report.

** Contextualise implies that we are more interested in seeing evidence of you having thought about the task, and determined *reasons* for the relative performance of different methods, rather than the raw scores of the different methods you select. This is not to say that you should ignore the relative performance of different runs over the data, but rather that you should think beyond simple numbers to the reasons that underlie them.

We will provide LaTeXand RTF style files that we would prefer that you use in writing the report. **Reports are to be submitted in the form of a single PDF file.** If a report is submitted in any format other than PDF, we reserve the right to return the report with a mark of 0.

Your name and student ID should **not** appear anywhere in the report, including any metadata (filename, etc.). If we find any such information, we reserve the right to return the report with a mark of 0.

5 Project Stage II

During the reviewing process, you will read two anonymous submissions by your classmates. This is to help you contemplate some other ways of approaching the Project, and to ensure that every student receives some extra feedback. You should aim to write 150-300 words total per review, responding to three *'questions'*:

- Briefly summarise what the author has done in one paragraph (50-100 words)
- Indicate what you think that the author has done well, and why in one paragraph (50-100 words)
- Indicate what you think could have been improved, and why in one paragraph (50-100 words)

6 Assessment Criteria

The Project will be marked out of 30, and is worth 30% of your overall mark for the subject. The mark breakdown will be:

Report Quality: (26/30 marks)
You can consult the marking rubric on the Canvas/Assignment 3 page which indicates in detailed categories what we will be looking for in the report.

Kaggle: (2/30 marks)
For submitting (at least) one set of model predictions to the Kaggle competition.

Reviews: (2/30 marks)

You will write a review for each of two reports written by other students; you will follow the guidelines stated above.

7 Using Kaggle

Task The Kaggle competition will be on predicting **salary_bin** (*not* mean_salary).

Instructions The Kaggle in-class competition URL will be announced on LMS shortly. To participate do the following:

- Each student should create a Kaggle account (unless they have one already) using your Student-ID
- You may make up to 8 submissions per day. An example submission file can be found on the Kaggle site.
- Submissions will be evaluated by Kaggle for accuracy, against just 30% of the test data, forming the public leaderboard.
- Prior to the closing of the competition, you may select a final submission out of the ones submitted previously – by default the submission with highest public leaderboard score is selected by Kaggle.
- After the competition closes, public 30% test scores will be replaced with the private leaderboard 100% test scores.

8 Assignment Policies

8.1 Terms of Data Use

The data set is derived from the resource published in Bhola et al. (2020):

> Akshay Bhola, Kishaloy Halder, Animesh Prasad, and Min-Yen Kan. 2020. Retrieving Skills from Job Descriptions: A Language Model Based Extreme Multi-label Classification Framework. In Proceedings of the 28th International Conference on Computational Linguistics, pages 5832–5842, Barcelona, Spain (Online). International Committee on Computational Linguistics.

This reference **must** be cited in the bibliography. We reserve the right to mark any submission lacking this reference with a 0, due to violation of the Terms of Use.

The demographic labels were added based on data from the Department of Statistics Singapore.

Please note that the dataset is a sample of actual data posted to the World Wide Web. As such, it may contain information that is in poor taste, or that could be construed as offensive. We would ask you, as much as possible, to look beyond this to focus on the task at hand. If you object to these terms, please contact us (`lea.frermann@unimelb.edu.au`) as soon as possible.

Changes/Updates to the Project Specifications

We will use Canvas announcements for any large-scale changes (hopefully none!) and Ed for small clarifications. Any addendums made to the Project specifications via the Canvas will supersede information contained in this version of the specifications.

Late Submission Policy

There will be **no late submissions** allowed to ensure a smooth peer review process. Submission will close at **5pm on May 19th**. For students who are demonstrably unable to submit a full solution in time, we may offer an extension, but note that you may be unable to participate in and benefit from the peer review process in that case. A solution will be sought on a case-by-case basis. Please email Hasti Samadi (`hasti.samadi@unimelb.edu.au`) with documentation of the reasons for the delay.

Academic Misconduct

For most students, discussing ideas with peers will form a natural part of the undertaking of this project. However, it is still an individual task, and so reuse of ideas or excessive influence in algorithm choice and development will be considered cheating. We highly recommend to (re)take the academic honesty training module in this subject's Canvas. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy[4] where inappropriate levels of collusion or plagiarism are deemed to have taken place. Content produced by generative AI (including, but not limited to, ChatGPT) is *not* your own work, and submitting such content will be treated as a case of academic misconduct, in line with the University's policy.

References

Bhola, A., Halder, K., Prasad, A., and Kan, M.-Y. (2020). Retrieving skills from job descriptions: A language model based extreme multi-label classification framework. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5832–5842, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Elazar, Y. and Goldberg, Y. (2018). Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Han, X., Shen, A., Cohn, T., Baldwin, T., and Frermann, L. (2022). Systematic evaluation of predictive fairness. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 68–81, Online only. Association for Computational Linguistics.

Joshi, M., Das, D., Gimpel, K., and Smith, N. A. (2010). Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 293–296, Los Angeles, California. Association for Computational Linguistics.

[4] http://academichonesty.unimelb.edu.au/policy.html

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.