The University of Melbourne
School of Computing and Information Systems
COMP30027 Machine Learning, 2019 Semester 1

Project 1: Gaining Information about Naive Bayes

Due:	1pm, 5 Apr 2019
Submission:	Source code (in Python) and (inline) responses
Marks:	The Project will be marked out of 20, and will contribute 20% of your total mark. This will be equally weighted between implementation and responses to the questions.
Groups:	You may choose to form a group of 1 or 2. Groups of 2 will respond to more questions, and commensurately produce more implementation.

Overview

In this Project, you will implement a supervised Naive Bayes learner, evaluate it with respect to various supervised datasets, and examine the impact of attribute correlation on the classifier. You will then use your observations to respond to some conceptual questions about Naive Bayes.

Naive Bayes classifiers

There are some suggestions for implementing your learner in the "Naive Bayes" lecture; ultimately, the specifics of your implementation are up to you, and will depend on which question(s) you choose to answer.

For marking purposes, a minimal submission will have a `preprocess()` function, which opens the data file, and converts it into a usable format. It will also define the following functions:

- `train()`, where you calculate counts (or probabilities) from the training data, to build a Naive Bayes (NB) model
- `predict()`, where you use the model from `train()` to predict a class (or class distribution) for the test data
- `evaluate()`, where you will output your evaluation metric(s), or sufficient information so that they can be easily calculated by hand
- `info_gain()`, where you will calculate the Information Gain (IG) for one (or each) attribute, relative to the class distribution

There is a sample iPython notebook `2019S1-proj1.ipynb` that summarises these, which you may use as a template. You may alter the above prototypes to suit your needs; you may write other helper functions as you require.

The use of external packages (specifically `sklearn`) is generally forbidden where it would replace implementation of the core functionality of the Project requirements; if you are curious about a specific case, you should post to the LMS Discussion Forum.

Data

For this Project, we have adapted nine of the classification datasets available from the UCI machine learning repository (`https://archive.ics.uci.edu/ml/index.html`):

> `anneal`, `breast-cancer`, `car`, `cmc`, `hepatitis`, `hypothyroid`, `mushroom`, `nursery`, `primary-tumor`

These datasets vary in terms of number of instances, number of attributes, number of different class labels; these are described in more detail in the accompanying README. It is not a strict requirement that your submission can automatically process every one of the datasets, and it is technically possible to complete this Project while only examining one or two datasets. However, it is strongly recommended that you examine all (or at least most) of the data available, so that you reduce the likelihood that you arrive at faulty conclusions due to a small sample space.

Questions

The following problems are designed to pique your curiosity when running your classifier(s) over the given data sets:

1. The Naive Bayes classifiers can be seen to vary, in terms of their effectiveness on the given datasets (e.g. in terms of Accuracy). Consider the Information Gain of each attribute, relative to the class distribution — does this help to explain the classifiers' behaviour? Identify any results that are particularly surprising, and explain why they occur.

2. The Information Gain can be seen as a kind of correlation coefficient between a pair of attributes: when the gain is low, the attribute values are uncorrelated; when the gain is high, the attribute values are correlated. In supervised ML, we typically calculate the Infomation Gain between a single attribute and the class, but it can be calculated for any pair of attributes. Using the pair-wise IG as a proxy for attribute interdependence, in which cases are our NB assumptions violated? Describe any evidence (or indeed, lack of evidence) that this is has some effect on the effectiveness of the NB classifier.

3. Since we have gone to all of the effort of calculating Infomation Gain, we might as well use that as a criterion for building a "Decision Stump" (1-R classifier). How does the effectiveness of this classifier compare to Naive Bayes? Identify one or more cases where the effectiveness is notably different, and explain why.

4. Evaluating the model on the same data that we use to train the model is considered to be a major mistake in Machine Learning. Implement a hold–out or cross–validation evaluation strategy. How does your estimate of effectiveness change, compared to testing on the training data? Explain why. (The result might surprise you!)

5. Implement one of the advanced smoothing regimes (add-k, Good-Turing). Does changing the smoothing regime (or indeed, not smoothing at all) affect the effectiveness of the Naive Bayes classifier? Explain why, or why not.

6. Naive Bayes is said to elegantly handle missing attribute values. For the datasets with missing values, is there any evidence that the performance is different on the instances with missing values, compared to the instances where all of the values are present? Does it matter which, or how many values are missing? Would a imputation strategy have any effect on this?

If you are in a group of 1, you will respond to question (1), and one other of your choosing (two responses in total). If you are in a group of 2, you will respond to question (1) and question (2), and two others of your choosing (four responses in total). A response to a question should take about 150–250 words, and make reference to the data wherever possible. Note that not all questions are equally difficult. Also note that not all questions are equally interesting. `(-:`

Submission

Submission will be made via the LMS, as a single file or archive of files. Submissions will open one week before the submission deadline. If you are in a group of two students, only one group member should submit.

The submission mechanism will stay open for one week after the deadline; late submissions will be penalised at 10% per business day, up until one week has passed, after which we will no longer accept submissions.

Assessment

10 of the marks available for this Project will be assigned to whether the five specified Python functions work in a manner consistent with the materials from COMP30027. Any other implementation will not be directly assessed (except insofar as it is required to make these five functions work correctly).

10 of the marks will be assigned to accurate and insightful responses to the questions, divided evenly among the questions that you are required to attempt. We will be looking for evidence that you have an implementation that allows you to explore the problem, but also that you have thought deeply about the data and the behaviour of the relevant classifier(s).

Changes/Updates to the Project Specifications

If we require any (hopefully small-scale) changes or clarifications to the Project specifications, they will be posted on the LMS. Any addendums will supersede information included in this document.

Academic Misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualisation and framing of the problem. For example, what the Project is asking you to do, or what you would need to implement to be able to respond to a question.

However, sharing materials beyond your group — for example, plagiarising code or colluding in writing responses to questions — will be considered cheating. We will invoke University's Academic Misconduct policy (`http://academichonesty.unimelb.edu.au/policy.html`) where inappropriate levels of plagiarism or collusion are deemed to have taken place.

Data references

`anneal` (`https://archive.ics.uci.edu/ml/datasets/Annealing`) is thanks to:

> David Sterling & Wray Buntine. [original source unknown]

`breast-cancer` (`https://archive.ics.uci.edu/ml/datasets/Breast+Cancer`) and `primary-tumor` (`https://archive.ics.uci.edu/ml/datasets/Primary+Tumor`) are thanks to:

> Matjaz Zwitter & Milan Soklic (physicians)
> Institute of Oncology
> University Medical Center
> Ljubljana, Slovenia.

`car` (`https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`) is derived from:

> Zupan, Blaz, Marko Bohanec, Ivan Bratko, and Janez Demsar (1997) Machine Learning by Function Decomposition, in *Proceedings of the International Conference on Machine Learning*, Nashville, USA.

`cmc` (`https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice`) is derived from a subset of the 1987 National Indonesia Contraceptive Prevalence Survey, and is described in:

> Lim, Tjen-Siem, Wei-Yin Loh, and Yu-Shan Shih (1999) A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms, in *Machine Learning*, Vol. 40, pp. 203–229.

`hepatits` (`https://archive.ics.uci.edu/ml/datasets/Hepatitis`) is thanks to:

> G.Gong (Carnegie-Mellon University), via
> Bojan Cestnik
> Jozef Stefan Institute
> Ljubljana, Slovenia.

`hypothyroid` (`https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease`) is derived from:

> Quinlan, J. Ross (1986) Induction of Decision Trees, in *Machine Learning*, Vol. 1, pp. 81–106.

`mushroom` (`https://archive.ics.uci.edu/ml/datasets/Mushroom`) is derived from:

> Schlimmer, Jeff (1987) *Concept Acquisition through Representational Adjustment*, Technical Report No. 87-19, University of California, Irvine.

`nursery` (`https://https://archive.ics.uci.edu/ml/datasets/Nursery`) is derived from:

> Olave, Manuel, Vladislav Rakovič and Marko Bohanec (1989) An application for admission in public school systems, in *Expert Systems in Public Administration*, Elsevier, pp. 145–160.