The University of Melbourne
School of Computing and Information Systems
COMP90049: Introduction to Machine Learning, 2020 Semester 2

Project 1: Naive Bayes and K-Nearest Neighbour for Predicting Stroke

Due:	5 PM 11 Sep 2020
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. This is an **individual** assignment which you are expected to complete on your own.

Overview

In this project, you will implement Naive Bayes and K-Nearest Neighbour (K-NN) classifiers. You will explore inner workings and evaluate behavior on a data set of stroke prediction, and on this basis respond to some conceptual questions.

Implementation

The lectures and workshops contained several pointers for reading in data and implementing your Naive Bayes classifier. You are required to implement your Naive Bayes classifier from scratch. You may use Python libraries of your choice for implementing K-NN, evaluation metrics, procedures and data processing.

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

- `split_data()`, where you split your data sets into a training set and a **hold-out** test set.
- `train()`, where you build Naive Bayes and K-NN classifiers from the training data. You can create train your data as you answer the related question.
- `predict()`, where you use a trained model to predict a class for the test data. You can also do prediction as you answer the related question.
- `evaluate()`, where you will output the **accuracy** of your classifiers, or sufficient information so that it can be easily calculated by hand.

There is a sample iPython notebook `S2020S2-proj1.ipynb` that summarizes 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. Depending on which answers you choose to respond, you may need to implement additional functions.

Packages

To provide the possibility of running your codes by markers, limit utilising the packages to the following packages in this project:

- `pandas` to read, split and preprocess the data
- `sklearn` to develop K-NN model and evaluate models
- `numpy` to implement scientific computing
- `math` to access to the mathematical functions
- `matplotlib` to create plots and visualizations

Data

For this project, we have adapted the Stroke data that have been used for stroke prediction [1], available online at (`https://data.mendeley.com/datasets/x8ygrw87jw/1`):
Some critical information:

1. File name: `stroke_update.csv`
2. 2740 instances
3. 10 attributes that include numeric and nominal attributes. The attributes `avg_glucose_level`, `bmi` and `age` are numeric. the rest of attributes are nominal.
4. The file `stroke_features.txt` explains each attribute
5. 2 classes, corresponding to the stroke outcomes: {0: No stroke, 1: Having stroke}

Questions

You should respond to **questions 1-3**. In question 2 (b) you can choose between two options for smoothing and two options for Naive Bayes formulation. A response to a question should take about 100–200 words, and make reference to the data wherever possible.

Question 1

a Explore the data and summarise different aspects of the data. Can you see any interesting characteristic in features, classes or categories? What is the main issue with the data? Considering the issue, how would the Naive Bayes classifier work on this data? Discuss your answer based on the Naive Bayes' formulation [2]. (3 marks)

b Is `accuracy` an appropriate metric to evaluate the models created for this data? Justify your answer. Explain which metric(s) would be more appropriate, and contrast their utility against accuracy. *[no programming required]* (2 marks)

Question 2

a Explain the independence assumption underlying Naive Bayes. What are the advantages and disadvantages of this assumption? Elaborate your answers using the features of the provided data. *[no programming required]* (1 mark)

b Implement the Naive Bayes classifier. You need to decide how you are going to apply Naive Bayes for nominal and numeric attributes. You can combine both Gaussian and Categorical Naive Bayes (option 1) or just using Categorical Naive Bayes (option 2). Explain your decision.
For Categorical Naive Bayes, you can choose either `epsilon` or `Laplace` smoothing for this calculation. Evaluate the classifier using accuracy and appropriate metric(s) on test data. Explain your observations on how the classifiers have performed based on the metric(s). Discuss the performance of the classifiers in comparison with the Zero-R baseline. (4 marks)

c Explain the difference between `epsilon` and `Laplace` smoothing. *[no programming required]* (1 mark)

Question 3

a Implement the K-NN classifier, and find the optimal value for K. (1 mark)

b Based on the obtained value for K in question 4 (a), evaluate the classifier using accuracy and chosen metric(s) on test data. Explain your observations on how the classifiers have performed based on the metric(s). Discuss the performance of the classifiers in comparison with the Zero- R baseline. (2 marks)

c Compare the classifiers (Naive Bayes and K-NN) based on metrics' results. Provide a comparatory discussion on the results. *[no programming required]* (1 mark)

Submission

Submission will be made via the LMS, as a single Jupyter Notebook file. Submissions will open one week before the submission deadline.

Assessment

5 of the marks will be assigned to whether the Python functions work correctly.

15 of the marks will be assigned to accurate and insightful responses to the questions, divided among the questions as been stated in question descriptions. 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 behavior 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.

Late submission penalty

There will be a penalty of 10% (2 marks) for every day (including working days and holidays) of late submission.

Academic Misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualization 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, plagiarizing 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

[1] T. Liu, W. Fan and C. Wu. A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset. Artificial Intelligence in Medicine, volume. 101, No. 101723, 2019, DOI 10.1016/j.artmed.2019.101723

[2] M. A. Munoz, L. Villanova, D. Baatar and K. Smith-Miles. Instance spaces for machine learning classification. Machine Learning, volume. 107, pp. 109-147, 2018, DOI 10.1007/s10994-017-5629-5