Assignment 2

Start Assignment

- Due 8 May by 22:00
- Points 10
- Submitting a file upload
- Available until 8 May at 23:59

The overall goal of the exercise is to get hands-on experience with the implementation of a very popular and fast machine learning algorithm for a real-world problem. The task is to implement the Naive Bayes algorithm that is able to predict the domain - one of Archaea, Bacteria, Eukaryota or Virus - from the abstract of research papers about proteins taken from the MEDLINE database. You will then apply your implementation on a test set without class labels and submit the predictions of your implementation.

The challenge

The ultimate goal of this programming project is to come up with an implementation of a (possibly extended or modified) Naive Bayes algorithm, that achieves a high predictive accuracy on the test data. As a minimum requirement your implementation should be at least as good (or better) than a standard version of Naive Bayes as explained for example in Mitchell's "Machine Learning" textbook. Your algorithm should still be "Naive Bayes" in the sense that it makes the assumption that all attributes are conditionally independent of each other given the class. You might, however, change any other assumptions, representations or algorithm in your implementation. Basically, there are three task to be solved:

1. First of all, you need to decide about a suitable representation for the text in the abstract. An easy way to obtain an attribute-value representation is to identify all occurring words and generate 0-1 attributes stating whether or not the word occurs in the corresponding example. There are other possible representations, e.g. one could take the occurrence frequency of a word in the abstract into account.
2. Implement the standard Naive Bayes algorithm as outlined for example in Mitchell's "Machine Learning".
3. Improve the model obtained using standard Naive Bayes algorithm. The improvement can include: adding new features such as n-grams (phrases of n words, for some n that is tunable hyperparameter), which features to be excluded (e.g. choose uncorrelated features), transformations that downweight common words and upweight rare words etc. For other suggestions, you can also look at **Rennie at al. (2003) "Tackling the Poor Assumptions of Naive Bayes Text Classifiers". (https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf)**

You have to implement from scratch the Naive Bayes algorithm. This means you are not allowed to use already existing Naive Bayes implementations. However, you are welcome to use numpy, pandas, nltk and sklearn for data prepossessing, model evaluation/validation.

Tips

- **'class' is a word in the abstracts.** One of the words in the abstracts is 'class'. So if you call the target feature 'class' and you include the word 'class' then there will be two columns in the data table called 'class'. Careful if you store the data in table format.
- **Compute probabilities in log space and add them instead of multiplying them; log(ab) = log(a) + log(b).** The result will be the same in theory but multiplying tiny numbers can give computational issues.
- Naive Bayes assumes that all words appear independent of each other conditional on the class.
    - *homo sapiens* appears in 8 training instances and they are all in class E (E = eukaryote, humans are eukaryotes). If the word homo appears then sapiens is likely to come with it, and vice versa. Using the same piece of evidence twice when really it's just one piece of evidence, can lead to wrong classifications
    - Concatenate into one word homo-sapiens.
    - Only use one of the words eg. homo.
    - Other examples: *escherichia coli, human immunodeficiency virus*.

The challenge is hosted at Kaggle. (https://www.kaggle.com/t/6de52a3d03f3032b8d4bd9f28d994e38) You can find the data set there. Have Fun!

What to submit

Canvas submission

You need to submit a Jupyter notebook with the experiments you run, that is the source file (raw notebook file) and an HTML generated from the notebook. The notebook should include two sections: A report section, followed up by the code section.

The report section should

1. explain and motivate the chosen representation & data preprocessing,
2. explain the idea behind the model improvements and their implementation (including the implementation of the standard Naive Bayes)
3. explain the evaluation procedure (e.g., cross-validation or training/validation split)
4. include and explain the training/validation results for the standard and improved Naive Bayes model. You can summarize results using tables (or plots), but all results have to be explained descriptively as well.
5. be written in plain English and should not be longer than **two A4 pages (**export the notebook as pdf to see if the report section fits in two pages). It should.

Kaggle submission

You have to submit your predictions for the test set to Kaggle. **Log in Kaggle with your university email (Google) account, and use your UPI as the team name. If we do not find your upi you will get zero marks for parts of the assignment.**

You only need to submit **your best model (typically the one you developed in task 3)**. Make sure that your submission appears on the leaderboard.

To avoid overfitting to the test set **you can make a maximum of 4 Kaggle submissions per day** ("Participants will need to wait until the next UTC day after submitting the maximum number of daily submissions."). You can always check the public leaderboard for the accuracy of your submission.

We will grade your model based on the private leaderboard. The number of submissions eligible for the final private leaderboard is set to 2. ("Users can hand-select the eligible submissions, or will otherwise default to the best public scoring submissions.")

Evaluation criteria

Submissions that do not fulfill the requirements for the format will get reduced marks.

You will be evaluated based on

- the prediction performance of your classifier, relative to the null model that takes just just the majority class, and relative to the standard Naive Bayes;
- the implementation (code has to be clean, well-documented, and well-written);
- the summary report of the proposed method to improve standard Naive Bayes, implementation, evaluation procedure, and results.

A2

Criteria	Ratings					Pts
Overall style Expected: The code is clearly written and commented which makes it easy to follow. The report section is clearly structured and uses headers	**1 Pts** **Full marks**	**0.5 Pts** **Has flaws with both comments and code clarity**	**0 Pts** **Not sufficient** No comments and/or unreadable code and/or difficult to follow report.			1 pts
Kaggle Results The points here are calculated based on the competition score on the private test set. 0pts if the accuracy is lower than simply using the "Majority Class" classifier.	**2 Pts** **above 0.96**	**1.5 Pts** **0.92-0.959**	**1 Pts** **0.88-0.919**	**0.5 Pts** **0.55 - 0.879** 0.55 is performance of the "Majority Class" Classifier	**0 Pts** **[0, 0.549]** worse than 0.55	2 pts
Data Representation (and Preprocessing) for Standard Naive Bayes (SNB) Expected: Description and motivation for the data format used and why it makes sense with SNB. If pre-processing techniques were used, they are explained.	**1 Pts** **Full marks**	**0.5 Pts** **either no motivation or on description of data represention/preprocessing**	**0 Pts** **Not sufficient** No comments at all on data representation or preprocessing			1 pts
Proposed extensions Expected: Motivation and Description of implemented extensions. Why did you decide to implement this extension? How can you tell that it's a good idea?	**1.5 Pts** **Full marks**	**1 Pts** **Description but no motivation**	**0.5 Pts** **motivation and description are provdied but it is difficult to understand them;** clarity issues; not able to implement a code based on that description	**0 Pts** **Not sufficient** No comments and/or unreadable code and/or difficult to follow report.		1.5 pts
Implementation of SNB The code for the basic Naive Bayes has been written from scratch without the usage of any packages. numpy,	**1.5 Pts** **Full marks**	**1 Pts** **Minor issues** E.g. Minor parts of the code taken from somewhere	**0.5 Pts** **Major issues** Code "seems" to be correct, but some	**0 Pts** **No marks** Not provided, or used package implementation		1.5 pts

Criteria	Ratings				Pts
pandas, nltk and sklearn only allowed for data prepossessing/data representation, evaluation/validation and nothing else. Points are deducted if packages have been used for more than that. If Turnitin indicates that the code has been copied from anywhere, no points will be granted.		but properly cited and commented.	running errors. We do not decode coding errors.	or copied code without justification/citation.	
Implementation of improved model Expected: Extension could be changes in the SNB algorithm and/or preprocessing, and have to be provided as a separate function. Should be clearly indicated what hyper-parameters have been introduced and how these have set/tuned. Laplace smoothing and log-calculations do not count as an extension. If Turnitin indicates that the code has been copied from anywhere, no points will be granted	**1.5 Pts** **Full marks**	**1 Pts** **Minor issues** Only single code provided, not clear what is extension of what is basic model. E.g. Data preprocessing has been the proposed extension, but also the original model uses data preprocessing. Not clear how hyper-parameters were set;	**0.5 Pts** **Major issues** E.g. Code does not run, Code does not implement or not inline with what the report describes.	**0 Pts** **No marks** Not provided, used package implementation or copied code without justification/citation, or proposed code does not count as an extension.	1.5 pts
Evaluation Expected: A description of how model/performance evaluation (for both initial model and improved model) has been carried out is included in the report. Results are consistent with Kaggle Public Scores.	**1.5 Pts** **Full marks**	**1 Pts** **Unjustified Results** Evaluation procedure is discussed but methods are not justified by the results, e.g., extension has worse accuracy than original model	**0.5 Pts** **Major issues** A description of evaluation procedure is provided, but it is not clearly written or the report results are not consistent with Kaggle results.	**0 Pts** **No marks** Not provided	1.5 pts

Criteria	Ratings	Pts
		Total points: 10