TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

Week 5 Assignment CS7CS4/CSU44061 Machine Learning

Assignment: Write A Short Group Project Proposal

Your task is to pick a project topic and write a short description of your proposed project. There are two main types of project:

1. *Application project* (most common). Pick an application that interests you, and explore how best to apply learning algorithms to solve it.

2. *Algorithmic project.* Pick a problem or family of problems, and develop a new learning algorithm, or a novel variant of an existing algorithm, to solve it.

Projects can of course also combine elements of applications and algorithms. It's often a good idea to either pick an application area that you're interested in or a machine learning technique that you want to explore more. Try not to be over-ambitious though - its far too easy to end up spending a lot of time on a project. Bear in mind that the project is worth 30% of the overall marks and 2-3 people are working together on it, so use that to calibrate the amount of time and effort you put into it.

You project must include the following components:

1. Gathering of raw data. For example, by scraping it yourself off the web (from twitter, news etc) or taking it from open data sources (many countries, including Ireland have open data initiatives with e.g. live city data, social and health data). Do *not* use pre-packaged data from Kaggle, UCI or the like - its important to get experience of working with raw data.

2. Pre-processing/cleaning up of the raw data. If you use only official data sources where the data is quite "clean" the data prep will be less and so you should plan on spending correspondingly greater effort on modelling, feature engineering, evaluation etc. If using such datasets it's often a better idea to augment the data with data from other sources, e.g. scraped from the web, since then you will have a more unusual and interesting dataset to work with.

3. Feature engineering/selection. The choice of features should be informed by the data e.g. by fitting a linear model and using the learned weights to identify important/unimportant features, by progressively adding/removing features to see the effect on performance, and of course by plotting the data to visualise the depence on each feature.

4. Model selection. Try to use more than one significantly different type of model e.g. a logistic regression and kNN classifier (but not logistic regression and SVM classifier, unless the SVM classifier is kernalised). Do not just use various flavours of linear model e.g. linear regression, ridge and lasso.

5. Performance evaluation. This must include comparison against a reasonable baseline and, importantly, a critical discussion/reflection on results e.g. drilling down into the data to analyse the factors affecting performance. Discuss why your choice of baseline is reasonable - rather than just using the sklearn defaults try to come up with a baseline suited to the data at hand e.g. for time series data its often better to use a baselines that predicts that the next value will be the same as the current value, for geographic data the mean in each region/district might be a more reasonable baseline than the overall mean. If you use a complex model such as a CNN then the baseline should include a simpler model e.g. logistic regression.

Some examples of project topics that have been done to death and should be avoided:

1. Predicting house sales/rental prices from housing data such as number of bedrooms, area, location etc. This is a Kaggle etc favourite, and usually all that's found is that house prices depend on (i) location and (ii) house size/type. If you are still v keen on working with housing data then you need to find a new and interesting angle e.g comparing airbnb and longer term rentals, and you need to explain this clearly in your project proposal.
2. Predicting stock market/cryptocurrency gains/losses from time series data of stock movements - generally this fails as if there really was an easily discovered trend then traders would have already exploited it.
3. Predicting climate/C02/temp a few days/months/years into the future from government/official time series data. Similarly health data (covid etc), agricultural data etc. Such forecasts are largely worthless since they ignore important external events. If you would like to look at such topics (and they are undoubtedly interesting) try to find a new angle on them e.g. by combining multiple data sources, taking account of external events, looking for interesting cross-over effects (such as between climate/pollution and health).
4. Anything that has already been done on Kaggle. Otherwise you run the risk of suspicion of plagiarism - your project should be new and your own.

Documents

For the project you need to generate two documents. The first is a short project proposal, which is this assignment. Then later you need to write the project report. Both the project proposal and project report should be submitted on Blackboard.

The project proposal document should give the title of the project, the full names of the team members and their student IDs and a 300-500 word description of what you plan to do. The description should include:

1. Motivation: What problem are you tackling?
2. Dataset: What data will you use and how will you collect it?
3. Method: What machine learning techniques are you planning to apply or improve upon?
4. Intended experiments: What experiments are you planning to run? How do you plan to evaluate your machine learning algorithm?

As a guide, the final report should be about 5 pages long (including figures and tables) - excessively long reports may be penalised as its important to learn to present results succinctly yet clearly. Generally it should consist of the following sections:

1. Introduction (about 0.5 page)
   - Explain the problem and why it is interesting.
   - State what the input and output are, e.g. "The input to our algorithm is an {image, amplitude, patient age, rainfall measurements, grayscale video, etc.}. We then use a {SVM, neural network, linear regression, etc.} to output a predicted {age, stock price, cancer type, music genre, etc.}."
   - Comment on why this input and output can potentially achieve the intended aim (if it cannot then of course you need to think again).

2. Dataset and Features (about 1 page)

    - Describe your dataset. How/where did you gather the data, how many data points are there, what date range is covered, what preprocessing did you do, what normalization or data augmentation. Show some examples from your dataset. Is it time-series data (where values close in time can be expected to be correlated) or not?
    - Discuss how you mapped the data to features e.g how did you map categorical values to numerical ones, text to numbers. Explain the rationale. If this mapping involves hyperoparameters (e.g. min_df in a bag of words model) then use cross-validation to choose values for these.
    - Use the data to analyse the appropriateness of the features. E.g. at a minimum plot the target value vs each feature to visualise the dependence, if any. If the target value doesn't show much dependence on any of the features then perhaps you need addition data or to construct new/augmented features - just marching on with the rest of analysis when an initial look at the data already suggests it might be inadequate for the task is not a good idea. If your data is time-series data then consider whether to use lagged data as features, and what choice of lags might be appropriate.

3. Methods (about 1 page)

    - Describe the learning algorithms that you used. For each algorithm, give a short description (a paragraph) of how it works - we are looking for your understanding of how these machine learning algorithms work, what hyperparameters they have, what trade-offs the models involve. This is especially important if you use methods not covered in the module since using methods in a black box fashion is terrible practice and you need to clearly communicate that you have not done that.

4. Experiments/Results/Discussion (about 2-3 pages)

    - Be sure to give enough detail to allow the reader to understand exactly what inputs/features are used for each model considered, and what outputs - a common mistake is to be too vague about this.
    - Give details about what parameters and hyperparameters you chose, and how you chose them. Be sure to use cross-validation to select hyperparameters and to state the number of folds used. How many training/validation/test examples did you use? Explain whether you think you may have overfit your training set and what, if anything, you did to mitigate that. For model parameters did you use gradient descent or some other algorithm?
    - Before you describe your results explain what your primary metrics are: accuracy, precision, AUC, etc. Also explain what data you use for performance evaluation, especially whether its is held out data not used for training/cross-validation.
    - For results, you want to have a mixture of tables and plots. If you are solving a classification problem, you should include a confusion matrix and AUC/ROC curves. Include performance metrics such as precision, recall, and accuracy. For regression problems, state the average error. Make sure to refer to and discuss the figures/tables in your main text throughout this section (no standalone plots please!). Your plots should include legends, axis labels, and have font sizes that are legible when printed. Plot some example predictions vs the real data so as to make it easier to assess the quality of the predictions, don't just quote metrics such as accuract.

- Be sure to compare performance against a reasonable baseline - this is mandatory.
- You should have a mix of both quantitative and qualitative results.
- Its a good idea to drill down into your data to try to understand the factors affecting performance.

5. Summary (100-200 words)

   - Summarize your report and reiterate key points e.g. which algorithms were the highest performing, why you think that some algorithms worked better than others - backing any opinions up with reference to supporting data.

6. Contributions. This doesn't contribute to the 5 page limit.

   - Describe what each team member worked on and contributed to the project. It's important to give a decent level of detail e.g. what code was written by whom, which experiments were carried out by whom, who wrote each part of the report. Each team member needs to initial this section (electronic initials are fine).

7. Include a github link (or similar) to the code written as part of the project.

Notes

- *Deep learning.* If you decide to work on a deep learning project, please make sure that you use other material you learned in the class as well. For example, you might set up logistic regression or SVM as a baseline for evaluation. Also, be aware that training deep learning models can be very time consuming (don't underestimate this!) and make sure you have the necessary computing resources.

- *Using methods not covered in class.* It's ok to use ML methods not covered in the class but you must take care to properly demonstrate in your report that you really do know how the methods work – it's far too easy to just use tools in a black box way but that's really poor practice and is a good way to haemorrhage marks.

- *Pre-prepared datasets.* While we don't want you to have to spend much time collecting raw data, the process of inspecting and visualizing the data, trying out different types of pre-processing, and doing error analysis is an important part of machine learning. We therefore ask that you do not use pre-prepared datasets (e.g. from Kaggle, the UCI machine learning repository, etc.).

- *"Official" datasets.* Government and statistical office datasets are often heavily pre-processed and quite "clean". If you use such a dataset you will have to spend less time on data cleaning/prep etc but you are expected to spend a corresponding greater amount of time on modelling, feature engineering, performance evaluation etc. If using such datasets it's often a good idea to augment the data with data from other sources e.g. scraped from the web.

- *Replicating results.* Replicating the results in a paper or online post can be a good way to learn. However, instead of just replicating previous work also try using the technique on another application or do some analysis of how each component of the model contributes to final performance.

- *Team working.* The project involves a team because working with raw data often involves a decent amount of effort to gather and clean up the data. A team allows this work to be shared and so leave enough time to focus on the more interesting stuff e.g. results and discussion. However, we understand that problems can arise when working in a team. With that in mind:

- Do keep a record of the work carried out by the individual team members. You'll need to detail this in the final report and it will help if any problems arise.
- If team related problems do occur during the course of the project, do contact us as early as possible. We can, for example, re-organise teams or take other steps but those are often best done relatively early during the course of the project.
- Every team member should make a significant contribution to the machine learning aspects of the report e.g. just doing the report writing or just collecting the data is not enough and will lose you marks.