Machine Learning Challenge

The machine learning challenge is intended to be a practical and fun, cumulative project, to be completed in teams of 3-4.

We will be collecting data that asks students questions about several popular food items. A TA will collect the responses from students and share the data with you in CSV file. Your task is to use the provided data to build a classifier related to this data set, likely to predict which food item a student is referring to in their response. The exact task will be confirmed after the course staff has a chance to experiment with the data ourselves.

Your goal is to build a classifier that can perform well on an unseen test set that will *not* be shared with you. We will collect this test set by asking TAs and instructors to answer the same set of data collection questions as students. Your model is expected to perform reasonably on this unseen data.

You are welcome to use any materials you want from this course. You can choose any model you'd like, or use an *ensemble* of multiple models, as long as your model files do not exceed 10MB in size.

One important aspect of this challenge is that the test set is not shared with you, and that your model should perform reasonably on this data. To do well, be mindful of whether when your model underfits/overfits.

To make this fun, there will be a prize (to be determined) for the group(s) who perform the best on the unseen test set!

Machine Learning Challenge Components

The ML challenge is worth 14% of your final grade in this course. It has three components:

Data Collection (1%) – Due Jan 24, 10pm

Completion of the data collection, which will be in the form of a Quercus quiz with several questions about 3 popular food items. This should take 5-10 minutes to complete. Your grade will be based on timely completion.

Group Formation (1%) – Due Feb 14, 10pm

Form groups of 3-4 on Markus. In order to earn this point, a student should:

- Either invite other students in the group *or* accept the invitation. **You will not get the credit if you do not accept the group invitation on Markus by the deadline.**
- Submit a description of how your group will work together by answering the following questions:
  - When will you meet to collaborate on the project? (Please set regular meeting times, and set aside time close to the deadline.)
  - What will be your main method of communication? (e.g., Slack, Discord, email, . . .)
  - What platform will you use to write the report? (e.g., Overleaf, Google Doc, . . .)
  - How will you distribute the work? What is each person responsible for? It is totally okay to have multiple people work together on one thing, but please still assign a "lead" person who is responsible for making sure that the collaboration happens. Please provide specific answers, even if those answers may change. For example, an answer like "we will decide on meeting times collaboratively" is *not* acceptable; instead, have those collaborative discussions as you are forming teams. Likewise, "we will work on all parts of the lab collaboratively" is not acceptable. These discussions are important to have prior to making the final decision to work as a team.

Model Predictions(4%) – Due Mar 28, 10pm

Submit a python3 script called `pred.py` that has a function `predict_all` that takes as parameter the name of a CSV file containing the test set, and returns predictions for that test set. Your script can use the following imports:

- numpy
- pandas
- basic python imports: sys, csv, random

We will be using python3.10 on the Markus system.

You will not be able to use sklearn, pytorch, tensorflow, etc in the final submitted code, so that you have a chance to build models yourselves. You may reuse any code that you wrote (or provided to you) in any of the labs, provided that they do not use the prohibited imports.

Remember that your prediction script only needs to perform **inference**.

Thus, you can still use these above imports for **learning**. That is, during the exploration phase of your project, you are welcome to use whatever tools you want before deciding what model you would like to build. You are also welcome to use any code/data you want to *build* the models you use. (For example, you might choose to use `sklearn` to explore several linear models. If it turns out that a logistic regression model is the "best", you might extract the weights from the `sklearn` model as a numpy array, and write a prediction script that uses that numpy array and no other `sklearn` code.)

You can submit additional files that are used by `pred.py`, e.g. to store your model parameters. All of your files, taken together, should be at most 10MB in size. Your script should not require networking, and should not download any new files. Your model script should be reasonable in its use of memory resources (TBD-determine limit), and should be able to make ~60 predictions within 1 minute (TBD when we get the test set).

Your model will be graded for the following:

- **(4/4) Full credit** Model script is runnable, meets the criteria for file size, imports, and such, and produces reasonable results over the test set. We will set the "reasonable" threshold such that groups who follow good machine learning practises and chose reasonable models should be able to pass the threshold.
- **(3/4) Partial credit** Model script is runnable, meets the criteria for file size imports, and such, and produces reasonable results over the test set. The model is clearly overfit to the provided training data.
- **(2/4) Partial credit** Model script is runnable, meets the criteria for file size imports, and such, and produces results over the test set that is better than random.
- **(1/4) Runnable Script** Model script is runnable, meets the criteria for file size imports, and such, and produces results over the test set that is about as good as random.
- **(0/4) Script not runnable** Model script is not runnable, fails the criteria for file size or imports, and such.

PDF Report (8%) – Due April 5, 10pm

Submit a pdf file called `csc311challenge.pdf` that describes your final model, as well as the steps that you took to develop this model.

Along with the PDF file, please also submit any other `.py` or `.ipynb` files that you used while developing your model. This latter file will not be graded, but is used as evidence that you completed your own work. The `.py` or `.ipynb` files do not need to be runnable by the TA, and can rely on external imports (e.g. sklearn) and data sets that you don't submit. Only your final `pred.py` file needs to be runnable.

Your report should describe the following:

- (2 points) Data: Describe how you explored your data, and the process through which you determined your input features. How did you end up representing your data? What else did you try? We are looking for:
  - A thorough exploration of the data, similar to what is presented in labs. What are the distributions of the features? How do these features correlate with the target?
  - If you use figures to answer those questions, an explanation of how you are interpreting those figures. Take care that your figures can be interpreted meaningfully.
  - A clear and logical description of how you determined your input features, with convincing logical or empirical evidence justifying your choice. Important features are not overlooked (e.g., not removed for "ease").
  - A clear description of the way(s) that you are representing the data in your models.
  - The descriptions should be consistent with the `.py` and/or `.ipynb` files that you used while developing your model.
  - A clear description of how you are splitting your data into various sets. You may use k-fold cross validation if you'd like, but if you do, describe how you are applying that idea.
- (1 points) Model: A description of the model(s) that you are evaluating/exploring. We are expecting a thorough exploration of at least 3+ family of models[1], even if you don't ultimately use those models. We are looking for:
  - How you are applying these models. You don't need to reiterate what the models are and how they work. Instead, we're looking for a description of the choices you are making to apply the model to this task: e.g., what features from the "Data" section are you using for each model? What adjustments (if any) did you need to make?

[1] A "family" of models are a group of similar models. For example, linear models of various types are considered to be a part of the same family.

- (4 points) Model Choice and Hyperparameters: How you are determining what model(s) to use in your final `pred.py`, as well as an exploration of hyperparameters. We are looking for:
  - A convincing explanation of how you ensured that the evaluation metrics for various models are comparable (i.e., are you using a consistent test set?)
  - A clear description of the evaluation metric(s) used to evaluate your model, as well as justification for its use.
  - A description of the hyperparameters that you are tuning, hyperparameter value combinations that you have tried, and the value of the evaluation metric(s) for those hyperparameters. We are not looking for an exhaustive search of all possible hyperparameter combinations, but there should be enough evidence to demonstrate that your hyperparameter choices are reasonable. (Note that just saying "X was the best hyperparameter value based on metric Y out of the hyperparameter values we tried" is not enough. Please present the values of the evaluation metrics for each hyperparameter value to *show* that your choice was the best one.)
  - A clear description of what your final model choice looks like in the submitted `pred.py` file.
  - The descriptions should be consistent with the `.py` and/or `.ipynb` files that you used while developing your model.
- (1 point) Prediction: How well would you expect your model to perform on the test set? We are looking for:
  - A point estimate for your performance, **not a range**.
  - A reasoned explanation of your expected model performance, with empirical evidence supporting your explanation. You are not graded on the closeness of your estimate to the final test accuracy, but you are graded on your reasoning.
- Workload Distribution: A description of what each person in the group contributed to the project. A 1-2 sentence description of each person's role is sufficient. Each student's description must be written by that student in order for them to receive credit for the project.

Recommendations

Start early! The data exploration portion can be done as soon as data is available.

Communicate early and often. There are a lot of tasks to do as a team and decisions to be made.

Keep a journal (e.g. google doc or overleaf) and record the experiments that you perform. If you keep clear, reproducible records, then writing the **Model Choice and Hyperparameter** section of your report would be much easier.

Try out various models using sklearn before writing your own implementation. Leverage the code that you have already written for the labs.

Be mindful that machine learning models can take a while to train, and that some models are more time-consuming to train than others.