CSC311 Winter 2025 Machine Learning Challenge

Alice Gao

Last Updated: February 10, 2025

Contents

Acknowledgement

These instructions are adapted from the work of Professor Lisa Zhang at the University of Toronto, Mississauga.

1 Introduction

Welcome to the CSC311 Machine Learning challenge. We hope this will be a practical and run project for you. You will complete this challenge in teams of 3-4 students.

Overview

You will receive a CSV file containing student responses about several popular food items. Your task is to build a classifier that predicts which food item a student is referring to in their response. We will share the final details of the task after the course staff explores the data.

Objective

Your goal is to create a classifier that performs well on an unseen test set. This test set, compiled from responses by TAs and instructors, will not be shared with you. Your model should generalize well to this unseen data.

Guidelines

Feel free to use any materials from this course, such as any model or an ensemble of models. Make sure that your model file submission does not exceed 10MB. We recommend that you pay careful attention to ensure that your model does not underfit or overfit and achieves a reasonable performance on the test set.

Prize

There will be a prize for the team(s) that perform the best on the unseen test set! We will announce the details near the deadline.

2 Components

The ML challenge is worth 15% of your final course grade. It has the following components.

Component	Weight
Data Collection	1%
Team Formation	1%
Prediction Accuracy	4%
Final Report	9%

Table 1: Project Grading Scheme

2.1 Data Collection

Due 1:00 pm on Thursday, January 23, 2025.

Worth 1% of your final course grade.

Complete the Quercus survey titled **ML Challenge: Data Collection**. This should take 5-10 minutes to complete.

2.2 Team Formation

Due 1:00 pm on Thursday, February 13, 2025.

Worth 1% of your final course grade.

Complete both tasks below.

- Form a group of 3-4 students. Create your group for the `project_proposal` and `project_final` assignments on MarkUs.
- In order to earn this 1% mark, you should either invite other students or accept the invitation on MarkUs. You will not get the credit if you do not accept the group invitation on MarkUs by the deadline.

  Your team members can be from any section of the course. Feel free to use the "Search for Teammates" post on Piazza.
- Complete and submit the **group contract** on MarkUs.

2.3 Model Prediction

Due 1:00 pm on Tuesday, April 1, 2025.

Worth 4% of your final course grade.

Submit a Python3 script named **pred.py** in the **project_final** assignment on MarkUs.

TODO: Provide pred.py with predict_all function in the starter code.

The script must include a function `predict_all` that takes the name of a CSV file as a parameter and returns predictions for the data in the CSV file.

Allowed Imports

Your **pred.py** script can use the following imports: Python 3, numpy, pandas, and basic Python imports such as sys, csv, and random. However, it cannot import sklearn, PyTorch, or TensorFlow.

We encourage you to explore different families of models and build your final model using any tools, including sklearn, PyTorch, or TensorFlow. You may reuse the code provided or written by you in any of the labs. However, your final **pred.py** script must generate predictions for the test data using your final model **without** using any of these imports (sklearn, PyTorch, or TensorFlow).

Environment

We will use Python 3.10 on the MarkUs system.

Code Requirements

- Your code only needs to be able to make predictions. You can use any code or data to build your models.
- You can submit additional files used by `pred.py` (e.g., to store your model parameters). All files combined should not exceed 10MB.
- Your script should not require networking or download any new files.
- Ensure your model script uses memory resources reasonably **(TBD determine limit)**, and can make approximately 60 predictions within 1 minute **(TBD when we get the test set)**.

Grading Criteria

The basic criteria include: runnable script, file size at most 10MB, only use allowed imports and uses reasonable memory.

For the results over the test set, we will set the **reasonable** threshold such that groups who follow good machine learning practices and choose reasonable models should be able to pass the threshold.

Grade	**Meets basic criteria**	**Quality of the results**
4/4	Yes	Produces **reasonable** results over the test set.
3/4	Yes	Produces **reasonable** results over the test set. The model is clearly overfit to the training data.
2/4	Yes	Produces **better than random** results on the test set.
1/4	Yes	Produces about **about as good as random** results on the test set.
0/4	No	Not applicable

Table 2: Model Prediction Grading Criteria

2.4 Report

Due 1:00 pm on Tuesday, April 1, 2025.

Worth 9% of your final course grade.

- Submit a PDF file named **report.pdf** in the **project_final** assignment on MarkUs.

  This file describes your final model and outline the steps to develop this model.

  We highly recommend typesetting the file using LaTeX or Microsoft Word.

- Submit a ZIP file named **code.zip** in the **project_final** assignment on MarkUs. This file should contain all the **.py** and/or **.ipynb** files used to develop your final model.

  Please exclude the `/data` folder from the starter code.

  These files will not be graded but serve as evidence of your work. The files do not need to be runnable by the TA and can rely on external imports (e.g. sklearn) and datasets you don't submit.

Your report should include the following sections:

- Data (2 points)
  - **Exploration:** Present a thorough exploration of the data. For example, describe the distributions of the features and how the features correlate with the target. The labs contain examples of such explorations.
  - **Feature Selection:** Explain how you determined your input features. Justify your choice with logical or empirical evidence. Ensure that important features are not overlooked or removed for "ease."
  - **Data Representation:** Describe how you represented the data in your models.
  - **Data Splitting:** Describe how you split your data into various sets. If using k-fold cross-validation, explain its application.
  - **Figures:** Use figures where appropriate. Explain your interpretations of the figures.
  - **Consistency:** Ensure the descriptions align with your `.py` and/or `.ipynb` files.
- Model (2 points)
  - Explore at least three families of models, even if you don't ultimately use some of these models.
  - Describe the models you evaluated.
  - Explain the rationale for applying each model to this task.
- Model Choice and Hyperparameters (4 points)
  - **Model Selection:** Explain how you chose the final model(s) for `pred.py`. Ensure that the evaluation metrics for various models are comparable.
  - **Evaluation Metrics:** Describe the metrics used to evaluate your model. Provide justification for using these metrics.
  - **Hyperparameter Tuning:** Describe the hyperparameters you tuned, the combinations tried, and the corresponding evaluation metrics. Provide evidence that your hyperparameter choices are reasonable. We do not expect an exhaustive search of all possible hyperparameter combinations.
  - **Final Model:** Clearly describe your final model choice as implemented in `pred.py`.
  - **Consistency:** Ensure the descriptions align with your `.py` and/or `.ipynb` files.
- Prediction (1 point)
  - **Performance Estimate:** Provide a point estimate of your model's performance on the test set. Providing a range will earn no marks.
  - **Reasoning:** Provide an explanation of your expected model performance on the test set. Support your explanation with empirical evidence.
- Workload Distribution: Include a 1-2 sentence description of each team member's contribution to the project. Each description must be written by the respective team member to receive credit for the project.

3 Recommendations

- **Start Early:** Begin data exploration as soon as the data is available.
- **Communicate Regularly:** Frequent communication is key for task coordination and decision-making.
- **Keep a Journal:** Use a shared document (e.g., Google Docs or Overleaf) to record your experiments. Clear, reproducible records will simplify writing the Model Choice and Hyperparameter section of your report.
- **Experiment with Models:** Test various models using sklearn before implementing your own. Utilize code from your labs.
- **Plan for Training Time:** Be aware that some machine learning models require significant time to train.