6.790 Mini-Project 1: Fighting Fires

Please hand in your work via Gradescope via the link at https://gradml.mit.edu/info/homeworks/. If you were not added to the course automatically, please use Entry Code R7RGGX to add yourself to Gradescope.

1. Please type your solution. We highly encourage LaTeX although it is not required.
2. Project is due on **Friday October 18 at 11PM**. However, you can turn it in up until **Monday October 21 at 11PM with no penalty**, but be careful with your time management because we have an exam on Thursday October 24.
3. Lateness and extension policies are described at `https://gradml.mit.edu/info/class_policy/`.

1 Background

This assignment is based on the paper: A Data Mining Approach to Predict Forest Fires using Meteorological Data, Paulo Cortez and Aníbal Morais, *13th Portuguese Conference on Artificial Intelligence*, 2007.

You *do not* have to use the methods or metrics they describe in the paper. We are inspired by the problem, but have substituted a dataset that is more tractable for learning methods.

The data consists of 500 data points, each corresponding to a fire, described with the features:

- c_x, c_y coordinates within a national park, in $\{1, \ldots, 9\}$
- month : integer : omitted in our notebook
- day : integer : omitted in our notebook
- fine fuel moisture code : integer
- duff moisture code : integer
- drought code : integer
- initial spread index : integer
- outside temp (in degrees c)
- outside relative humidity (in percent)
- outside wind speed (in km/h)
- outside rain (in mm/m^2)

The label, Y, is the area burned (in hectares (ha) = 10,000 m^2). Note that there are several examples with the label 0. In fact, that means that there was a fire, but it was less than .01ha in size.

In the data we give you, these features have all been scaled to be between 0 and 1.

2 Problem setting

Our goal is to use this data in a way that goes beyond the original paper. In particular, we are going to use it to make decisions about how much equipment to send to fight the fire.

We will assume the following *completely made up and wrong* model of fire response, outcomes, and utilities:

- We can make the following responses: *none*, *small*, *medium*, *large*.
- Each response r to a fire that would naturally have actual terminal size (that is, if left alone to burn itself out, how much area would it burn) s has the following probability of quickly containing the fire:

$$q(r, s) = \begin{cases} .9 & \text{if } s < s_r \\ .1 & \text{otherwise} \end{cases}$$

  where the sizes s_r are constants indicating about how large of a fire each response can probably contain: $s_{\text{none}} = 0$, $s_{\text{small}} = 40$, $s_{\text{medium}} = 120$, and $s_{\text{large}} = 2000$.
- Given that, we have the following outcome distribution (in terms of ha burned):

$$o(r, s) = \begin{cases} \max(0.01, 0.05s) & \text{with probability } q(r, s) \\ 0.85s & \text{with probability } 1 - q(r, s) \end{cases}.$$

- There is an economic cost for sending out each type of response ($c_{\text{none}} = 0$, $c_{\text{small}} = 30,000$, $c_{\text{med}} = 100,000$, $c_{\text{large}} = 500,000$).
- There is an economic cost of $20,000$ of burning 1 hectare of land.

3 Tools

Here are some hints about tools you might find helpful:

- Read about *mean absolute deviation* and *median absolute deviation* as alternatives to *mean squared error* as a measure of real-valued predictions.
- Consider using *cross validation* to select your hyper-parameters or choose between hypothesis classes. *Do not use your validation data to pick hypothesis class, algorithm, or hyperparameters.*

You might find these tools useful. Be sure to read the documentation. No need to use all (or any!) of them.

- `matplotlib.pyplot.hist` for making histograms
- `sklearn.metrics.accuracy_score`
- `sklearn.metrics.cross_val_score`
- `sklearn.metrics.confusion_matrix`
- `sklearn.metrics.mean_squared_error`
- `sklearn.metrics.root_mean_squared_error`
- `sklearn.metrics.median_absolute_error`

- `sklearn.metrics.PredictionErrorDisplay`
- `sklearn.linear_model.LogisticRegression`
- `sklearn.linear_model.LinearRegression`
- `sklearn.linear_model.RidgeRegression`
- `sklearn.tree.DecisionTreeClassifier`
- `sklearn.tree.DecisionTreeRegressor`
- `sklearn.dummy.DummyClassifier`
- `sklearn.dummy.DummyRegressor`
- `sklearn.neural_network.MLPClassifier`
- `sklearn.neural_network.MLPRegressor`

4 Questions

We have several questions about alternative formulations of this problem, and then we ask you to implement one approach.

We will focus particularly on your assessment of the accuracy and usefulness of the decisions your model makes. If you don't advise trusting its decisions in all or part of the input space, you should say something about that.

1. Examine your data before starting. What do you notice about the distribution of outputs?

2. In this question, we will consider a classic **regression** approach, in which we take in a description of the fire situation and make a prediction for the amount of *area burned*.
   (a) Assume for now that we know the exact area (number of hectares), a, that some particular fire would burn. What would be the optimal (risk minimizing) decision rule when deciding what response to make?
   (b) If we were using a Bayesian method that generates a posterior predictive distribution $p(a \mid x)$ on the burn area, what would be the optimal decision rule when deciding what response to make?
   (c) In the original paper, the authors evaluate regression methods using both squared error and MAD, the results are substantially different. Explain the trade-offs between these two metrics.

3. An alternative strategy, based on *classification according to response* is to train an ML model to directly predict, based on the fire features, what response action to take. One way to do this is to label each fire in the training set according to the optimal response (which you can compute from the current label, which is the area burned), and then phrase our problem as a classification problem, where the classes are the four possible responses.
   (a) Are we losing information about the output structure by encoding the outputs as independent classes? Explain.
   (b) In this problem, do all mis-classifications have the same economic cost? If so, argue why. If not, give two concrete mis-classification examples with different costs.

4. One more strategy is still to do a classification into responses, but to use a loss function that is more sensitive to the application. We can do this by directly minimizing the empirical risk (instead of the cross entropy) of the classification.

(a) Assuming we are going to address our problem via standard gradient descent, we will need the output to be continuous function of the input, so let's assume the output is a distribution $p(r \mid x)$ represented using a softmax over responses r. If we want to directly minimize the risk in this optimization, what should the loss function be, as a function of the risk of taking a response r to a fire of size a?

What else would we have to change in our model to make this loss function be continuous? (Just point out the problem, you don't have to solve it).

5. Implement *one* of the approaches above (regression, classification, direct ERM). The Google Colab notebook
`https://colab.research.google.com/drive/1WKoM3PnGaSVn-O6HgtOZwGRP1RTqAfRt`
contains code for downloading and splitting the data. The approaches in questions 2 and 3 can be pursued straightforwardly using tools from `sklearn`. The direct risk minimization approach from question 4 is harder and will require calling an optimizer directly (or moving to something like Tensorflow with a custom loss function).

   (a) Which approach did you decide to try?

   (b) In the paper, they included one very simple baseline. What was it? Why is it important to try a simple baseline? Can you think of one that might have been better? Try at least one baseline.

   (c) Which two hypothesis classes did you try? Explain the methods you chose, and the pros and cons of each.

   (d) What is your final answer? That is, what is your best decision rule and how did you select it? Provide the results of the tests you ran in the process of making this choice. (Include plots of residuals if you were using regression or a confusion matrix if using classification.)

   (e) Given an input x describing a new fire, how would you decide what response to send?

   (f) Provide the most crisp and accurate estimate you can make of the actual real-world utility of the decision rule you selected, when applied to new cases. Explain how you arrived at your answer.

6. How would your answer change if asked to make decisions for 2025, based on this data?

5 What to hand in and how it will be evaluated

Please submit a single pdf document. It should be typed, not handwritten, but does not need to be in latex. It should have clearly labeled answers to the questions above, written in good English paragraphs.

This is not a contest! We won't grade on your overall prediction error or cost. We will grade on your ability to formulate an ML problem, execute the formulation, and evaluate the result.

You may not submit code. Do please submit data, tables or graphs as necessary to make your points.

Grading rubric. For each question:

- 50%: Did your decisions make basic sense given the overall problem setting and the choices you had already made?
- 50%: Were your explanations clear and well thought out?