MCIS6273 Data Mining (Prof. Maull) / Fall 2025 / HW4

Points Possible	Due Date	Time Commitment (estimated)
20	Monday December 8 @ Midnight	*up to* 20 hours

- **GRADING:** Grading will be aligned with the completeness of the objectives.
- **INDEPENDENT WORK:** Copying, cheating, plagiarism and academic dishonesty *are not tolerated* by University or course policy. Please see the syllabus for the full departmental and University statement on the academic code of honor.

OBJECTIVES

- Perform data fusion with a new dataset.
- Build a test and training dataset in preparation for a classifier.
- Build a RandomForest classifier to learn the Aurora prediction label.
- Complete the online assessment.

WHAT TO TURN IN

You are being encouraged to turn the assignment in using the provided Jupyter Notebook. To do so, make a directory in your Lab environment called `homework/hw4`. Put all of your files in that directory. Then zip or tar that directory, rename it with your name as the first part of the filename (e.g. `maull_hw4_files.zip`, `maull_hw4_files.tar.gz`), then download it to your local machine, then upload the `.zip` to Blackboard.

If you do not know how to do this, please ask, or visit one of the many tutorials out there on the basics of using zip in Linux.

If you choose not to use the provided notebook, you will still need to turn in a `.ipynb` Jupyter Notebook and corresponding files according to the instructions in this homework.

ASSIGNMENT TASKS

(25%) Perform data fusion with a new dataset.

Now that we have developed some expertise in dark skies, we want to use what we know about supervised learning to make some predictions.

In the last assignment, we learned about how to use imputation to fill in missing data. This, for the most part, gave us a "complete" dataset, even if some of it was synthetic. We must always weight the pros and cons of having a "good enough", but complete dataset or an incomplete dataset.

This final assignment will challenge us further to make a prediction about where we might find the best possibilities for seeing the Aurora Borealis or the "Northern Lights". If you are unfamiliar with this phenomenon, please take a look here:

- Northern lights (aurora borealis): What they are and how to see them? by Daisy Dobrijevic

What we will ultimately land on is a binary classifier which might be able to tell us which locations would have been good places to see it (and possibly good places in the future).

Some of the major features of such good places are:

- latitude (usually in the northern hemisphere),
- dark skies (usually > 21 bortle and without the moon),
- intense solar activity – without the appropriate minimum solar activity, there will not be anything to see.

You might notice, that we actually have two of these three features, but the third? Solar acitity is a dataset we're going to have to obtain from external sources.

So, in this first part we're going to begin to build the final data set which will allow us to train a classifier.

In this part we will:

- obtain the data,
- assemble and merge a particular field back into our updated GaN dataset which has complete (imputed) `SQMReadings` based on time-date information.

One of the fun things about this is that we are going to learn how to merge external data with our original dataset and ultimately learn the process of how to build a classifier based on that data. We will ultimately have a rudimentary classifier that allows us to know or to be, let's say, more aware of where in the world we would best see the Aurora Borealis by exploiting what we've already know about dark skies.

§ Task: 1.1 Load the *Matza, et al* dataset into a DataFrame.

In this process will have to learn a little bit more about the data that we are dealing with. That data will come from the following repository:

- The GFZ Helmholtz Centre for Geosciences

The dataset is:

- Matzka, J., Stolle, C., Yamazaki, Y., Bronkalla, O. and Morschhauser, A., 2021. The geomagnetic Kp index and derived indices of geomagnetic activity. Space Weather, https://doi.org/10.1029/2020SW002641

Which can be downloaded directly from:

- https://kp.gfz.de/app/files/Kp_ap_Ap_SN_F107_since_1932.txt

You will need to set your `read_csv()` parameters appropriately:

- set `sep="\s+"`,
- skip the first 40 rows with `skiprows=`,
- add back the columns (you must look at the original file and on the 40th skipped line are the columns).

§ Task: 1.2 Remove all data prior to Jan 01, 2014 and save this data in a CSV file.

- You may want to merge the date fields into a datetime field – it will definitely make your life easier (e.g. convert YY MM DD into a `datetime` object),
- save the file into `data/014_2024_solar_activity.csv`.

§ Task: 1.3 Update your GaN dataset so it includes all original and imputed `SQMReadings` as well as 4 new columns.

- you have already done the imputation in HW3, you just need to go back to your prior work and integrate,
- the new columns will be `"season_winter"`, `"season_summer"`, `"season_fall"`, `"season_spring"` which will be binary columns and be based on the month of the measurment. You can easily do this with with the `DataFrame.apply()` method by looking at the date and assigning a binary `1` to the appropriate column and `0` everywhere else. For example, if the month for a GaN datapoint is March

(3) `season_spring` will be assiged `1` and all other seasons `0`.

Use the following as a guideline:

- `season_spring` is any month in March (3), April (4) or May (5),
- `season_summer` is any month in June (6), July (7) or August (8),
- `season_fall` is any month in September (9), October (10) or November (11),
- `season_winter` is any month in December (12), January (1) or February (2).

§ Task: 1.4 Merge the `Ap` field from the *Matza, et al* dataset into your GaN dataset.

This will require you to match the datetime of each dataset so that every row of the GaN data has a new field `"Ap"` which represents the `"Ap"` value for the date in the GaN row.

§ Task: 1.5 Read the short paper which explains the `Ap` and `Kp` indices.

There is nothing to do here, except learn a little about the data we're using – as a data scientist, this is a necessary step to increase your awareness of the conxtext and domain. This doesn't make you an instant expert, but rather a *trusted partner* in data analysis.

- *Understanding Solar Indices* by By Ian Poole, G3YWX

(25%) Build a test and training dataset in preparation for a classifier.

In this second part, we're going to get our hands into building data sets for training and testing.

Training datasets are necessary inputs to any supervised learning model and as we learned, they must have labeled data. That is, there must be target labels included with the training data, else, there is nothing to learn from!

We first must realize that the best way to build a data set is to work with a data set that we already have. And in this particular case there is good news – we have a very good one with over 100,000 rows worth of data that span a fairly long period of time from 2014 to 2024.

In order to build a good data set for the classifier, we're going to filter the new data set we have, set labels appropriately – and in the case, yes, we're going to set them synthetically since we don't have the *actual* data label. Finally, we're going to split that labeled dataset into random sets of testing and training data.

We need to remember that we are preparing to build a binary classifier (*positive* class and *negative* class), so in the training set we will need to have both positive and negative examples that are labeled. Within our dataset labeling data instances will be a fairly easy task.

Since we now have the seasonal information (`season_winter`, etc), we can use the lat/lon data already in our original GaN data, the seasonal information, and the information about the solar intensity from the new dataset to make a determination on which locations would be best served to see the Northern Lights. The binary label with just tell us a location is a good candidate (labeled 1), it won't tell us it is guaranteed – and yes, our rudimentary model is lacking some additional granularity which will make it more useful and powerful.

To select a random subset of the data that we know would be in the positive case requires us to pick data that are within the correct season, that are within the correct lat/lon, and that have a solar intensity that would likely produce Auroral activity.

We need to split the full dataset into a subset, and use part of this subset as a positive set and use the other part of the data as negative sets. These two sets combined will represent your *training data*.

Once we have training data, we need testing data. The testing data will come from another random subsample of the original data which is labeled data. It *must not* be data that came from or that was not used within the training set. This is a form of cheating, that we do not want to participate in when building models.

§ Task: 2.1 Split the merged data set into a random sample of 25% of the data.

Make sure the data includes at least 2500 data points which have a latitude greater than 60.

The total amount of data will be between 25-35K data instances (rows).

§ Task: 2.2 Create a new column, `label` which has a `1` or `0` based on the criteria given,

Place a `1` in the `label` column when the following are satisfied:

- the `Latitude` is greater than 60,
- the `SQMReading` is greater than 21,
- the `Ap` reading is greater than 26.

Otherwise place a `0` in the column.

§ Task: 2.3 Create test and training files from the data.

Store the labeled data accordingly:

- save 70% of the newly labeled data (random subset) into a file called `train.csv` (use `to_csv()` as you have in the past),
- save the remaining 30% of the newly labeled data into file called `test.csv`.

(25%) Build a RandomForest classifier to learn the Aurora prediction label.

In this third part the fun really begins – believe it or not, the hard part is behind us.

We're going to build a binary classifier that's going to predict the outcome of whether a given unlabeled data point this is a good location for Borealis or not and of course now that we have `SQMReading` data (from our prio work) for all of the data set we can make a prediction based on that as well.

Intuitively, the darker skies with higher intensity solar activity (that are strong enough) alonside the seasonal parameters in line with the data, *should* allow our algorithm to learn the features that allow it to make a good prediction. Though this is a simple classifier it will still yield important information whether or not these are good locations for seeing (or have higher potential for seeing the) Aurora Borealis.

In this particular case, we know that a binary classifier might not be granular enough for a sophisticated determination, but our goal is actually just to get our feet wet in a classifier development cycle.

§ Task: 3.1 Use `sklearn.ensemble.RandomForestClassifier` to build the classifier from your training data.

- study sklearn.ensemble.RandomForestClassifier,
- implement the classifier and train it on the data in your `train.csv`

You will receive further guidance on the best parameters to choose.

(25%) Complete the online assessment.

Once done with the prior assignment the parts, the last part requires you to just paste your solution to the questions below.

This will allow you to complete the work *offline* then just submit it to your BB for the final points.

You should also make sure to leave your solution in your submitted notebook as well – they should match up completely and this match will be verified.

§ Task: 4.1 After turning in your BB notebook solution, paste the completed table into the online HW4 assessment.

You do not need to paste the code – just the table and the classification report output.

- learn to use `sklearn.classification_report()`,
- make sure you use your test dataset in `test.csv` for `y_true` and you will use your model output on the test set as `y_pred`, this will make certain you are *truly* testing your model on unseen data and **not** the data that you trained on.

§ Task: 4.2 Provide some insight into your report.

In your answer (notebook and your Blackboard assessment submission) provide two sentences on how you think your classifier did. Make a statement about whether it would be good enough to put into public/production use, say as a tool that a travel agent might use to specialize in highly effective Northern Lights tours.