III. Model documentation and write-up

1. Who are you (mini-bio) and what do you do professionally? If you are on a team, please complete this block for each member of the team.
   My name is Azin Al Kajbaf. I have recently earned my Ph.D. degree in Civil & Environmental Engineering from the University of Maryland. The area of my research/academic focus was “Disaster Resilience” and my research involved the application of machine learning and statistical methods in coastal and climate hazard assessment.
   Currently I have a joint appointment with Johns Hopkins University and the National Institute of Standards and Technology (NIST) as a postdoctoral fellow. I am leveraging data science techniques and geospatial analysis to collaborate in projects to support community resilience planning through the development of methods and tools that evaluate the economic impacts of disruptive events.

   My name is Kaveh Faraji. I am a Ph.D. candidate at the University of Maryland, and I work in the area of Disaster Resilience. My main research is focused on risk assessment of natural hazards such as flood and storm surge. I am employing geospatial analysis and machine learning approaches in my research.

2. What motivated you to compete in this challenge?
   We implement machine learning and deep learning methods in our research projects. We are interested to learn more about these approaches and their real-world application. Competing in these challenges allows us to gain experience with different machine learning and deep learning techniques that we can potentially employ in our research projects as well.

3. High level summary of your approach: what did you do and why?

   For the competition we only used the past airport configurations and training labels data. We preprocessed the data and for each data point. We extracted information about distribution of the past configurations, current and 10 last configurations, and the duration that each past configuration was active. Also, we considered current date and time (hour, day, week, month) and the lookahead as predictors. In preprocessing step, we used some of the benchmark code functions (https://www.drivendata.co/blog/airport-configuration-benchmark/) like “censor_data” and “make_config_dist” for selecting part of the data that we were allowed to use and for creating distribution of past configurations, respectively. For more information about these functions, please refer to the benchmark code. At the end of preprocessing step, we created a DataFrame (train_labels) as an input for the machine learning algorithms.
   In the main code, we trained XGBoost models for each airport. We then preprocessed the test data features and use pretrained XGBoost models for predicting probability of each configuration.

4. Do you have any useful charts, graphs, or visualizations from the process?

5. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

We extracted 10 last configurations and their durations:

```
for ii in range(10):
    try:
        index_false = max(subset_new.index[subset_new['airport_config'] !=
                                    current_new[ii]])
        timestamp_new.append(subset.timestamp[index_false])
        current_temp, subset_new = censor_data(airport_config_df,
                                        timestamp_new[ii+1])
        current_new.append(current_temp)
        duration.append((timestamp_new[ii] -
                    timestamp_new[ii+1]).seconds/60)
    except:
        current_new.append(0)
        duration.append(0)
```

We used XGboost for developing the prediction models for each airport:

```
model = XGBClassifier(max_depth=5,
                      learning_rate=0.02,
                      objective='multi:softprob',
                      eval_metric=["error", "mlogloss"],
                      n_estimators=300,
                      min_child_weight=1,
                      reg_alpha=0.01,
                      gamma=0,
                      subsample=0.7,
                      colsample_bytree=0.7,
                      tree_method="hist",
                      use_label_encoder=True)
model.fit(X, y)
```

6. Please provide the machine specs and time you used to run your model.
   Our code was run within the specified competition container runtime
   - CPU (model): 6 CPU Xeon E5-2690
   - GPU (model or N/A): Tesla K80
   - Memory (GB): 56 GB
   - OS: Both Linux and Windows
   - Train duration: ~ 5 hours
   - Inference duration: ~ 1 hour

7. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?
   No.

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?
   No.

9. How did you evaluate performance of the model other than the provided metric, if at all?
   We only used Mean Agg Log Loss.

10. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

    We started with extracting different features from different datasets including:
    - Extracting weather information from the lamp file
    - Extracting estimated number of arrivals/departures for each airline in specific timespans.

    We did not get the opportunity to optimize our code to include these features as predictors.

11. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?

    As for features, we would start with incorporating the features we mentioned in question number 10.
    We would also be interested to consider other machine learning techniques beside XGBoost.