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.

Brian Hu is a first-year undergraduate student at Caltech majoring in Computation and Neural Systems. Currently, he works with Professor Thanos Siapas to develop machine-learning techniques for animal tracking and behavior quantification in live experimental settings. He is interested in researching human cognition and computational methods for modeling the brain.

Nika Chuzhoy is a first-year undergraduate student at Caltech majoring in Computer Science. Her primary interests lie in theoretical machine learning. She currently does research involving interpretability methods for biological deep learning models.

2. What motivated you to compete in this challenge?

We chose to compete in this challenge primarily to gain experience in the implementation of machine learning algorithms for data science. The Pushback to the Future Challenge provided us with the opportunity to do so while also working on a problem with real-world data and applications.

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

Our approach was divided into three main stages:

- Benchmark: We trained basic interpretable models such as linear regression and random forest regression to analyze feature importances and determine cross-dependencies.
- Analysis of the data: We identified potentially impactful features that could be extracted from the data, such as aircraft type, number of recent departures, and runways in use.
- Feature extraction: We extracted and encoded features which had been identified as important and filtered the data in order to prevent leakage.
- Modeling: We trained one CatBoost regressor for each of the ten airports provided and optimized hyperparameters.

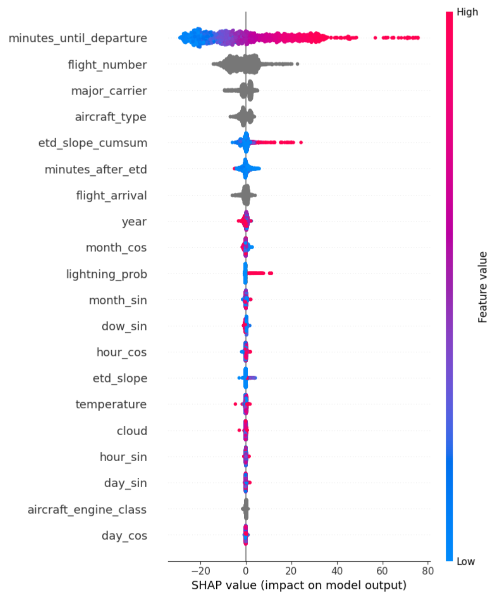
We iterated upon these stages multiple times as new insights were gained from the process.

4. How did knowing that you might need to federate this model change your approach to this challenge problem?

Our approach was not significantly affected by the knowledge that our model might be federated, primarily because we felt that a global model was not ideal for the small number of airports we were given to work with.

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

Pictured below is a chart of the SHAP values of the features in our final model.



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

The following line of code is used to extract features related to the estimated time of departure of flights. For each flight, a quotient is calculated representing the change in the most recent predictions for the departure time of the flight divided by the difference in the time at which the predictions were made. It also serves as an example of how we prevent data leakage, as NaNs are substituted to prevent future data from being accessed.

```
   etd = etd.sort_values(['gufi', 'timestamp']).reset_index(drop=True)
   etd['hg'] = etd['gufi'].shift(1)
   etd['hts'] = etd['timestamp'].shift(1)
   etd['hd'] = etd['departure_runway_estimated_time'].shift(1)

   etd.loc[etd.gufi != etd.hg, ['hts', 'hd']] = np.nan

   etd['ts_diff'] = (etd['timestamp'] - etd['hts']).dt.total_seconds()
   etd['etd_diff'] = (etd['departure_runway_estimated_time'] -
etd['hd']).dt.total_seconds()
   etd['etd_slope'] = etd['etd_diff'] / etd['ts_diff']
   etd = etd[(etd.etd_slope != np.inf) & (etd.etd_slope != -np.inf)]
```

The following block of code demonstrates how we handled timestamp features. Features such as the month and day of week were cyclically encoded. For example, “Monday” and “Tuesday” were treated as equally as close as “Sunday” and “Monday.” Both sine and cosine functions were used to prevent the same encoding from being assigned to different values.

```
   features.insert(loc = 2, column = 'hour_cos', value =
np.cos(features.timestamp.dt.hour*(2.0*np.pi/24)))
   features.insert(loc = 2, column = 'hour_sin', value =
np.sin(features.timestamp.dt.hour*(2.0*np.pi/24)))
   features.insert(loc = 2, column = 'dow_cos', value =
np.cos(features.timestamp.dt.day_of_week*(2.0*np.pi/7)))
   features.insert(loc = 2, column = 'dow_sin', value =
np.sin(features.timestamp.dt.day_of_week*(2.0*np.pi/7)))
```

Finally, the following line of code drops columns of the provided data that we found were unimportant. We experimented with including a variety of combinations of features, and determined that dropping these features resulted in the best model performance.

```
features.drop(columns=['cloud_ceiling', 'flight_type', 'wind_speed', 'isdeparture'],
inplace=True)
```

7. Please provide the machine specs and time you used to run your model.

- CPU (model): 2 vCPU
- GPU (model or N/A): N/A
- Memory (GB): 25 GB
- OS: Ubuntu
- Train duration: 6 hours
- Inference duration: 15 minutes on the provided data

8. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?

Running our model prints out "no objects info loaded," which seems to be an unresolved issue with the CatBoost package. We didn't observe any impact on performance.

9. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?

N/A.

10. How did you evaluate performance of the model other than the provided metric, if at all?

When evaluating our model, we considered Mean Squared Error and tendency to overfit. However, ultimately we selected our final model primarily based on performance under the provided metric (Mean Absolute Error).

11. What are some other things you tried that didn't necessarily make it into the final workflow?

(quick overview)

Approaches that were not incorporated into the final model include:

- A feature representing the length of the queue ahead of each flight, i.e. the number of flights scheduled to depart before a given airplane.
- Features incorporating the rates of arrivals/departures prior to a given flight.
- Partial globalization of the model, such as incorporating delays at airports other than the airport a flight is departing from.