Important Documents

[w261-Final-Project-2022](https://docs.google.com/document/u/0/d/1dIh9RDSp8TLZ1JPbuJqZn1PKIMVGv-1PeOMjSYuf7XM/edit)

[w261 Final Project Rubric - SU20](https://docs.google.com/document/u/0/d/1w4fcCvaHGAO06xWJfue2WjswgX5dFwcyimTvSc9efPE/edit)

Master To Do List

Project Deliverables

* ~~Phase 1 Notebook~~
* ~~Phase 1 Video (Presentation & Script)~~

Action Items - Everyone

* Read [carvalho\_review.pdf](https://drive.google.com/file/d/1r6Po1Oqh6X9UOUZXEcaYZuutCeLGFY4v/view?usp=sharing)
* Read [A Comparative Analysis of Models for Predicting Delays in Air Traffic Networks.pdf](https://drive.google.com/file/d/1wPnv1PG5Tootj6KQ7UHcI20nGVzQO47c/view?usp=sharing)
* Read [Prediction\_of\_weather-induced\_airline\_delays\_based\_on\_machine\_learning\_algorithms.pdf](https://drive.google.com/file/d/1_c2V1E9BOfCjncHD1Hj3ZST-YZnke8wJ/view?usp=sharing)
* ~~HW 5 :( - work in the shared space~~
* ~~Read through starter\_nb\_fp introduction (Phase 1 updates, Question Formulation)~~
* ~~Make final decision about 2020-2021 data → need this before the join~~

Final Stretch

* [~~Jennifer Conde~~](mailto:jconde@berkeley.edu) ~~- Jimi’s Baseline Model Description~~
  + [~~Dante Malagrino PhD~~](mailto:dantemalagrino@berkeley.edu) ~~- metrics for model that returns all positives~~
  + [~~Dante Malagrino PhD~~](mailto:dantemalagrino@berkeley.edu) ~~- metrics for model that returns all negatives~~
* ~~Code snippets in final notebook that can be executed~~
  + ~~Self-referenced join~~
  + [~~Ram Senthamarai~~](mailto:ram.senth@berkeley.edu) ~~- Model from scratch~~
* Final Notebook
  + [~~Ram Senthamarai~~](mailto:ram.senth@berkeley.edu) ~~- Clean up code organization section~~
  + [~~Dante Malagrino PhD~~](mailto:dantemalagrino@berkeley.edu) ~~- Update cross validation section to discuss strategies we tried (no cross validation, rolling window, normal cross validation)~~
  + Model Information [Section 5 & 6]
    - [~~Jennifer Conde~~](mailto:jconde@berkeley.edu) ~~- Model from scratch~~
      * ~~Model description~~
      * ~~Model parameters (ex. Input, tuning)~~
      * ~~Summary of model performance (ex. Time to run, metrics against training, metrics against testing)~~
      * ~~Code~~
    - Logistic regression
      * [~~Joy Moglia~~](mailto:jmoglia@berkeley.edu) ~~- Model description~~
      * ~~Baseline: CV=none~~
      * ~~Model parameters (ex. Input, tuning)~~
      * Summary of model performance (ex. Time to run, metrics against training, metrics against testing)
    - Random Forest
      * [~~Joy Moglia~~](mailto:jmoglia@berkeley.edu) ~~- Model description~~
      * ~~Baseline: CV=none~~
      * ~~Model parameters (ex. Input, tuning)~~
      * Summary of model performance (ex. Time to run, metrics against training, metrics against testing)
    - [Ram Senthamarai](mailto:ram.senth@berkeley.edu) - Summary Table for all PySpark Models
    - [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) - Error Analysis on All Models
    - Final Model Selection - Logistic Regression with Cross Validation
      * Note: Cross validation didn’t actually make a difference lol
      * “Our baseline was so good that there was no room for improvement” - Ram
  + [Ram Senthamarai](mailto:ram.senth@berkeley.edu) - Data Auditing section (Section 4: Feature Engineering)
  + [~~Jennifer Conde~~](mailto:jconde@berkeley.edu) ~~- Addressing Imbalance~~
  + Conclusion
    - [~~Jennifer Conde~~](mailto:jconde@berkeley.edu) ~~- Key Takeaways~~
    - [~~Jennifer Conde~~](mailto:jconde@berkeley.edu) ~~- Novelty~~
    - [~~Jennifer Conde~~](mailto:jconde@berkeley.edu) ~~- Challenges~~
    - [~~Jennifer Conde~~](mailto:jconde@berkeley.edu) ~~- Limitations & Future Work~~
  + [~~Jeffrey Adams~~](mailto:jeffrey.adams@berkeley.edu) ~~- Application of Course Materials~~
    - ~~Parallelization & Map-reduce~~
    - ~~Spark and RDDs~~
    - ~~Joining~~
  + [~~Ram Senthamarai~~](mailto:ram.senth@berkeley.edu) ~~- Add metrics against 2020 and 2021 data to metadata~~

Action Items - Individuals

* Jenny
  + EDA
  + Outlier detection
  + ~~Preprocessing notebook~~
  + ~~Phase 1 Deliverables~~
    - ~~Data wrangling subsection in EDA~~
* Joy
  + ~~EDA~~
  + ~~Problem statement and evaluation metrics~~
  + ~~Get list of US holidays (5-6 major holidays), create holiday feature in dataset~~
  + ~~Phase 1 Deliverables~~
    - ~~Finalize EDA in Notebook (add descriptions)~~
* Dante
  + ~~Problem statement and evaluation metrics~~
  + ~~Phase 1 Deliverables~~
    - ~~Train/Test Split Strategy (see notes from Lea’s OH on 3/11)~~
    - ~~Slides & Draft Script~~
    - ~~Update notebook to reflect 1 stakeholder~~
* Ram
  + ~~Joins~~
  + ~~Train/Test Split~~
  + ~~Time Zone Conversion~~
  + ~~Phase 1 Deliverables~~
    - ~~Add to Dataset section (in notebook):~~
      * ~~What datasets are we using (weather, stations, airport code, flights, aircrafts, etc.)~~
      * ~~Modify the features diagram~~
      * ~~Description of joins~~
    - ~~Description of Pipeline setup~~
* Jeff
  + ~~HW 5~~
* Unassigned
  + Finish joined dataset
    - ~~Clarify discrepancy in weather stations count~~
      * Future: Historic averages for each airport as fallback
    - ~~Include flight data from 2020/2021~~
      * [~~Ram Senthamarai~~](mailto:ram.senth@berkeley.edu) ~~- Pull 2019 out and create proper conditions in data pipeline for train and test~~
    - ~~Agree on which weather station/info will be used for the joined dataset~~
      * ~~Currently using 1 closest station~~
      * Future: ideally aggregate across multiple stations
    - Self-reference join to include incoming aircraft delay
      * ~~Option 1 - Has the incoming flight been delayed (0/1)~~
      * Future: Option 2 - Has the incoming flight been delayed - do we already know by how much (make sure we limit this info to whatever can be known 2 hours before departure of the predicted flight)
    - Future: Add tail number to aircraft information (join with flight data)
    - ~~Convert more times into UTC timestamps (e.g. scheduled and actual arrival times as well as actual departure time) - might be useful later in feature engineering~~
  + Data cleanup
    - ~~What do we do with CANCELLED flights?~~
    - ~~What do we do with international flights? Do we have any in the dataset?~~
    - ~~Finalize columns to drop/one hot encode~~
    - ~~Implement dropping/one-hot encoding~~
    - ~~Addressing null values (drop vs fill in with other data)~~
      * Future: Tail Number Data – ~2,000 missing values (in non-identifier rows)
    - Regularization
    - [Ram Senthamarai](mailto:ram.senth@berkeley.edu)- Saving normalization parameters, models, and one-hot encoding models
    - [Joy Moglia](mailto:jmoglia@berkeley.edu)Future: Log-scale features
    - Future: Revisit data normalization method
  + EDA - What EDA will be most relevant?
    - Agree on strategy for EDA - do we use the full dataset (i.e. including weather and flight data from 2020 and 2021) or do we limit the EDA to the training dataset only (i.e. we exclude 2019, 2020 and 2021)
    - Delay by airline
    - Delay by day of week
    - …
    - Correlation matrix (or matrices)
  + Modeling
    - Choose a Cross Validation strategy and implement it
    - Get functional model
    - Implement F-Beta Score
    - Implement model “from scratch”
  + ~~Slides & Presentation Scripting~~
  + Feature engineering and new feature creation
  + [Joy Moglia](mailto:jmoglia@berkeley.edu)- Look into undersampling/oversampling/weights
* Add more features & non-linear features
  + Ex. squaring
  + More EDA post-join

Meeting Notes

## Apr 4, 2022

* EDA
  + ~~More EDA on the weather dataset~~
* Data Wrangling
  + Nulls
  + Outliers
* Modeling
  + [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) & [Joy Moglia](mailto:jmoglia@berkeley.edu)
    - Strategy for running models – What types of models? What hyperparameters to tune?
    - Look into running XGBoost - what are the hyperparameters to tune and values to try?
  + [Ram Senthamarai](mailto:ram.senth@berkeley.edu) - Consolidate into functions so we can easily run multiple iterations easily
    - Databricks AutoML - does grid search for you, tracks model history
    - Databricks MLFlow
    - Models and plots folders - model artifacts
  + [Jennifer Conde](mailto:jconde@berkeley.edu) - ~~Implement undersampling, FBetaEvaluator, consolidated hyperparameter lists~~
  + [Jennifer Conde](mailto:jconde@berkeley.edu) & [Ram Senthamarai](mailto:ram.senth@berkeley.edu) - get working master function for Random Forest
  + Model\_summary for random forest like Logistic Regression Cmd 11
* Error Analysis
  + Error analysis on 2019, 2020, and 2021
  + [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu)
* Report notebook
* Notebook Cleanup
  + [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) EDA - clean only joined notebook
* Presentation slides
  + Compile existing slides (with emphasis on 2020 & 2021 data)
  + Add modeling slides
    - Model summary and results
  + Gap analysis
  + Code segments
* Manual Implementation of Logistic regression

## Apr 3, 2022

* Notes for final presentation
  + Emphasize EDA from full dataset:
    - Phase III presentation – frequency of flight cancellations (can add for another year besides 2020)

## Mar 31, 2022 | [W261 Section 2 Team 11 - Weekly Sync](https://www.google.com/calendar/event?eid=dGUzM2VjNmNqbzFwMDdlbjVsaTh0NzVpN29fMjAyMjA0MDFUMDAwMDAwWiBqY29uZGVAYmVya2VsZXkuZWR1)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* What needs to get done before Sunday?
* Cross Validation Strategy
  + What strategy works the best?
  + Completely randomize - remove time component and compare the time-dependent version
  + Cross validation code works
* Using F-Beta as a metric → CV Search would need to use this metric
* Baseline model - What is it? Existing logistic regression model without cross validation.
  + We can probably just define it ourselves
  + Fine with what we have right now - Logistic regression model that we currently have, without cross validation.
* What is our test set? 2019. Test it once on each “version” of the model (ex. baseline)
  + Don’t do error analysis on 2019
* Baseline pipeline - should be easy to convert to Spark pipeline
* What types of models do we want to try? Different features, cross validation, solving imbalance
  + Function to iterate on all these options and pull final results?
* Each model type in its own file (ex. Baseline, logistic regression, random forest), with a final command to get the results
  + Save results into blob storage to have final results summary table

Action items

* Joy, Jenny, & Ram - implement code to address imbalance
  + ~~.withColumn() – weight~~
  + ~~Pass weights into model~~
  + Model Search with Cross Validation - Cmd 13
  + ~~Not sure how to implement undersampling~~
  + Note:
    - Logistic Regression: F\_beta with class weights = 0.66
    - Logistic Regression: F\_beta with under/oversampling (ratio = 2) = 0.72
      * Ratio = 3, Ratio = 4 performed worse
    - Random Forests: F\_beta with class weights = 0.66
    - Random Forests: F\_beta with under/oversampling (ratio = 2) = 0.51
    - Random Forests: F\_beta with under/oversampling (ratio = 3) = 0.66
    - Random Forests: F\_beta with under/oversampling (ratio = 4) = 0.66
* ~~Dante - More EDA on joined dataset~~
* ~~Dante - Create additional notebooks for models beyond baseline~~
* Feature Engineering
  + Joy - Logged & Quadratic Information
  + Time-based features - integrating cyclic structure
* Jeff & Jenny - Incorporate tailnum data
  + More EDA
    - \_DISTANCE\_GROUP → how many flights fall into each group
  + How to fill in missing data
* Jeff & Joy - Add more features to model
  + Feature importance
* Phase III Presentation, Script, Video
  + ~~Saturday, 2 pm - Sync up on Slide deck~~
  + Sunday, 2 pm PST - Record

Notes from presentations

* JIMI wants code snippets in the slides
* JIMI wants pandemic data
* JIMI wants NOVELTY, AMBITION…. No “me too” pipelines!
* Script, rehearse, record — don’t be awkward
* Challenge conventions in a principled manner
* Figure out where you’re failing (airport? Day of week?)
* Find creative ways to improve results!
  + Use forecast data
* Team10: Range join > 7 min with 2hr lookback
  + Prove that they can get away with less data
  + CV approach shuffles data
* Team ?:
  + For missing weather data, looked at prior hour, or prior 2 hour. Invalid after a certain point.
  + sin/cosine for day of week?
  + undersampling/over increases performance
  + Model doesn’t generalize well > need more sophisticated options
  + CRS DEP TIME, percent delay tail num, and DAY\_of\_week\_sin = 90% importance for DTree
    - Convert categorical into numerical
    - sin/cosin captures cyclicality
  + Top 10 features are 90% important for logistic regression
  + Problem space is not linear > trees would be better
  + Considering a weighted CV to emphasize recent data
  + Has an irregular weather event flag/feature (ex: visibility lower than 1000 meters)

Our feedback from JIMI

* Great slides
* Compute baseline
* Use vectorized UDFs

## Mar 24, 2022 | [W261 Section 2 Team 11 - Weekly Sync](https://www.google.com/calendar/event?eid=dGUzM2VjNmNqbzFwMDdlbjVsaTh0NzVpN29fMjAyMjAzMjVUMDAwMDAwWiBqY29uZGVAYmVya2VsZXkuZWR1)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* Working session: baseline model
* Self-referenced join notes:
  + Has the prior flight arrived?
  + Was the prior flight delayed? → DEP\_DEL15
    - Departure time must be 2.25 hours before the current flight
    - Delayed by how much? → might be harder to use to prevent data leakage
  + Has the prior plane left yet?

Action items

## 

## Mar 22, 2022 | [FlightXI Team Meeting](https://www.google.com/calendar/event?eid=Xzhnc2pjaDFqOGNzMzRiYTM4NTMzZ2I5azYwc2syYjlwNmwwamFiOWk4b3A0NmNobjc1MGs0ZTIzOG8gamNvbmRlQGJlcmtlbGV5LmVkdQ)

Attendees: [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* Finish join
  + Data leakage issues / Aggregation strategy
  + Window weather - Ram has code
    - Get the closest weather recording
    - Airports have 1-5 weather stations within a 20 mile radius. Weather stations can have multiple readings per hour
    - If all weather stations report the weather, get an average of all readings (or some other aggregation for binary features)
      * Removes missing data (ideally)
    - Weather data should be in the format of 1 entry per airport per hour
    - For each weather station, can get a monthly average as a substitute for null values
  + 2 datasets → one for EDA, one for training?
    - Raw join and clean join
    - Use clean join in pipeline/for modeling
* Final Dataset
  + Self-referenced join (get delay of prior flight)
    - Sort by tail number and time, partition by tail number, pull this in the main join
    - Ex. limit scope of data to the same-day
    - What do we want to know about the incoming flight? Departure delay (binary and length), assuming the delay of the prior flight is known by 2 hours of the current flight departure time
* Feature engineering
  + How many flights were canceled for an airline by 2 hours before departure time?
  + How many flights were canceled in an airport by 2 hours before departure time?
* Data prep – after join but before modeling
  + Handling missing values on case-by-case basis
    - Drop columns where everything is null
  + Need to address outliers
    - Wind speed
    - Precipitation
    - Visibility
  + Drop? Fill in with average?
* Split
  + Also use 2019 as test
  + 2020 and 2021 is a totally separate set of test data since we don’t know how the model will react to this data
* Modeling
  + Have 1 notebook for model building → need skeleton in place
  + Baseline model → how do we improve after we have this in place?
* Post-baseline model:
  + Hyperparameter tuning
  + Feature engineering
  + Self-referenced join
* Data transformations
  + One-hot encoding
  + Data normalization
  + Null value handling
  + Outlier removal
* Documentation → separate notebook (model training pipeline):
  + Started with x rows
  + Dropped x rows for y reason
  + Applied x transformation
  + Data transformations, cross validation

Action items

* Meet tomorrow at 3 pm
* Finalize dataset
  + Jenny - Fix weather data to be 1 entry per hour per airport
  + Data Transformations
    - Jeff - Propose how to handle each column with nulls: [Nulls](https://docs.google.com/spreadsheets/d/1uBqNKW215wIDS0hbBQO_iKCkjFOebdylNeIkK-0XdiY/edit?usp=sharing)
      * “Diverted” columns → drop
    - Jeff - Look into outliers, propose how to handle them
    - Joy - One-hot encoding & data normalization
* Research how do we even build a model in PySpark lol
* Dante - look at how to perform self-referenced join
* In-Class Presentation Work
  + Introduce the business case
  + Introduce the dataset
  + Summarize EDA and feature engineering
  + Summarize algorithms tried, and justify your baseline prediction pipeline (all the steps) choice
  + Dante - Discuss crossfold validation and evaluation metrics in light of the business case
  + Discuss performance and scalability concerns
  + Talk about interesting pipeline errors, debugging experiences
  + Summarize limitations, challenges, and future work.

## Mar 17, 2022 | [W261 Section 2 Team 11 - Weekly Sync](https://www.google.com/calendar/event?eid=dGUzM2VjNmNqbzFwMDdlbjVsaTh0NzVpN29fMjAyMjAzMThUMDAwMDAwWiBqY29uZGVAYmVya2VsZXkuZWR1)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* Ram just got of meeting with Luis
  + Can now get the compressed files in blob storage → need to open the files
    - Challenges in opening these files in DataBricks/Spark
  + Have an approach now
* HW5
  + Jeff going to office hours to figure out Q8
* Data processing to be done
  + Get distinct list of weather stations and airports
  + Join flight and (new) weather data
    - Goal to have 1 parquet file with 1 row per flight, airplane information, weather, etc. → everything in 1 row for training the model
    - Bin by hour of day
  + Would it be worth joining the existing/old data anyway?
    - Yes, just in case
  + Incoming flight data: use tail number, date, airport to join
    - Join incoming flights regardless of whether or not it’s supposed to arrive within the 2 hour window
    - Was the prior flight delayed on departure? Departure delay could lead to arrival delay
      * Difficult because: departure might have happened on the previous day
      * Solve for scale by partitioning the data by date (?)
    - Incoming flight delay will be the best predictor of departing flight delay
      * Delayed on departure → “two hop query”
        + Connect to arrival first, then connect to airport from which it departs and get the information from there
      * Incoming flight delay in a different row of the same dataset (flights) → delay in arrival and departure in the same record
      * Look up the tail number and pull the previous flight
* Join
  + Merge flight with weather at departure and arrival airports
  + Incoming flight information
* Index of notebooks → where is everything lol → need links to subnotebooks from the main one
  + “Data Pipeline” for joins
* Phase 2 Goals: Join & EDA on join
* Tarfile and gzip → load\_weather\_from\_url file
  + Files in Azure → move to DataBricks and should still be compressed
  + Copy to temp folder in cluster
  + Use one of these libraries to extract the CSVs
  + Read the CSVs as dataframe
  + Write back to blob storage as parquet
  + Copy tar file from blob into temp location on cluster (the Spark driver) → explode on the driver
  + Read all CSV files and put back as parquet
  + `getmembers()`
  + Start at “Unzipping Files” section & Initialize notebook (ignore other cells between initialize and unzipping)

Action items

* Ram - get distinct list of weather stations and airports
* Ram - implement basic join (flights & weather, avoid self-reference joins)
* Finish HW 5 THEN run the join (join will take up basically all capacity on the cluster)
* Dante - think through notebook structure (modular structure)

## 

## Mar 15, 2022 | Team Meeting

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* Prepare weather and flight data
* HW5
* Phase 2 Priorities
  + Get all the new data
  + Join
  + EDA on final dataset

Action items

## Mar 13, 2022 | [W261 Section 2 Team 11 - Sun Sync](https://www.google.com/calendar/event?eid=MTN2aHE5cnI5OGNzNzdhaTZxZWl1bzI0bGIgamNvbmRlQGJlcmtlbGV5LmVkdQ)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* Preprocessing notebook for flight data
  + Ex. add holiday data (holiday and holiday window (within 5 days of a holiday))
* Final dataset will have additional holiday column
* Entire pipeline as 1 notebook ideally
* Careful about dropping data arbitrarily

Action items

## Mar 13, 2022 | [W261 - Vinicio's OH 2022-0103 DATASCI W261 Vinicio De Sola](https://www.google.com/calendar/event?eid=XzhncTNnY2E0OGQzNDRiOWc2NHMzaWI5azY0cWo2YjlvOGdvajZiYTI4Y3FrMmdxMjZsMzRjYzluNzQgY18zM2Y3NTNxZnQ3NzhyNXVicTBnOTB0dWcza0Bn)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu)

Notes

* Vinicio thinks we should not exclude airports (e.g. SJU) from the data analysis
* The big join will take 8-9 hours once we scale the cluster
* It takes a very long time (5 hours) to analyze the weather data
* EDA is part of phase II
  + Phase I is all about planning
  + EDA in phase I is mostly about getting a sense of the data (missing values, ranges,...)
* The notebook is a living document
* There is only one central notebook that will be submitted with the main findings
  + We will have multiple notebooks, dedicated to specific functions, but they should just be linked from the main notebook, not everything in one place
* The aircraft number (head? Or maybe tail?) is the data that carries information, more than the flight number
* Multitask learning? ⇐ Not sure what this is, but it got mentioned - It’s something that we would have to code ourselves - Probably was mentioned in Vinicio’s class
  + They are looking for some kind of novelty:
    - Get the data from 2020/2021 to check the model in COVID era
    - Multi-tasking ⇒ Instead of doing just classification or just regression, you can do a hybrid approach, where you take into consideration both. The loss function is not a standard cost function, but it will be a custom loss function, which will have to be implemented by hand by us
      * There might be a dual output, each with a loss associated with it - and we’ll have to figure out how to use both
      * Output instead of being only one, it may be more than one (two?) - The loss function will always be one, though
      * It’s essentially about combining two vectors of features (parameters) from two models
  + Bottom line ⇒ They are looking for novelty in our approach. What special things are we doing?
  + Here are a couple of links from a quick Google Search to start reading about Multi-task learning, if we are interested about it:
    - <https://en.wikipedia.org/wiki/Multi-task_learning#:~:text=Multitask%20Learning%20is%20an%20approach,tasks%20as%20an%20inductive%20bias>.
    - <https://towardsdatascience.com/multi-task-learning-in-machine-learning-20a37c796c9c>
* The 15 minute delay threshold is associated with something actionable by the airline (passenger compensation,...)
  + Use classification as first part of the problem (yes/no) and then use regression afterwards
    - How good is my model when the predicted time loss is 16, 17, 18,...60 minutes? This could be a secondary part of the analysis - Great idea to measure the quality of the model (still binary) based on the expected delay of the flight
* For the weather information, we should think about using a window of weather information (e.g. 6 to 2 hours before departure?)
  + These windows might be different depending on the specific piece of information that we are planning to leverage (snow, window speed,...)
  + Maybe we do a max/avg/last/… of those values, we’ll have to think about it on a case by case basis
* For the delay of the incoming flight, we can only use information that are from at least 2 hour before, i.e. we will most likely be unable to use the delay on arrival
* Cross-Validation
  + Two approaches (rolling window or expanding window)
  + Rolling window (e.g. 3 months)
    - Train (Jan-Feb-Mar) - Test (Apr)
    - Train (Feb-Mar-Apr) - Test (May)
    - …
  + Expanding window
    - Train (Jan-Feb-Mar) - Test (Apr)
    - Train (Jan-...-Apr) - Test (May)
    - …
    - You need to do some exponential weighting of the measurements to take into account the differences between sets
  + Rolling window is the preferable approach
    - We can decide whether to have an overlap (between windows) or not
    - It depends on how many models we want to use
    - It also depends on how much time it takes to run the models
    - We can make the decision later

## Mar 12, 2022 | [W261 Section 2 Team 11 - Sat Sync](https://www.google.com/calendar/event?eid=MnRiMm82MHE5bW40OXBtNnFyaTEzcjM5NDAgamNvbmRlQGJlcmtlbGV5LmVkdQ)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* Blob storage
  + Raw data
  + Folder with master data set – want to get 1 row for each flight with all the relevant info (airport information, weather info, airline, etc.)
* Need documentation of join strategy for Phase 1 presentation
* Ram working on a visualization of airports and weather stations
  + Do this for ATL and ORD
* Pull 2020 & 2021 data sooner rather than later
  + Can keep these just in test data set if we want, but would be good to process these now
  + Ask Lea about weather data for 2020 & 2021
* Add feature for holidays
  + Fourth of July, Thanksgiving, Christmas
* DEP\_DELAY\_GROUP → do we need this feature?
  + This feature could be helpful for multiclass classification as a stretch goal
  + Probably don’t need to consider early arrivals
* Data Pipeline Notebook
  + Airports Master
    - Data timezone (tz) → IANA timezone takes care of daylight savings
    - Primary weather station
  + Weather stations closest to airport → 5 closest stations within 20 mile radius
    - If an airport as its own weather station, do we need the “5 nearest”
      * Yes for airports without weather stations
    - May want to average the closest 5 observations based on data quality
* Instructors have deleted some weather stations and have duplicated rows :o
  + For the purpose of *confusing us!!!* (confirmed with Lea)
* Late arrival of prior flight → what is the best way to get this information and prevent data leakage?
  + Delay network → given a root delay, what are the downstream effects/what’s the chain reaction? Need the network effect
  + How long has the incoming flight been delayed in its departure?
* Data joins
  + Keep all features and let ML select? Or limit ourselves first?
  + 2-pronged approach:
    - Correlation analysis to see how features are related
    - PCA/Ridge → where do these lead us?
  + Hybrid model since there’s *so much* data
  + Not advisable to eliminate features based on our data subsets since these are biased
  + Join entire dataset based on Ram’s criteria
    - Good to limit the weather dataset first since there is a lot of noise
* Dante working on aircraft type based on tail number
* Question Formulation - define problem, target audience, what is “success” (metrics)
  + Sell to airlines, airports, and passengers
  + Dataset
    - If we include the graphic, make sure we include some of these features
    - Because our data is limited, we are not including *all* of them, but we still use this as a reference, presents a nice high-level view of what the problem is
    - Should we note on the graphic which features we are able to include → Ram can do this
    - Make sure we document new datasets we are introducing
  + Metrics → Why is β equal to 0.5? The ultimate source of Wikipedia says you should use either 0.5 or 2 depending on how you value precision and recall.
    - Ideally we’d gather sentiments from customers (ex. via an experiment) and adjust this accordingly
* EDA - why we’re looking into this data, now that we are looking at it, what is interesting
  + % flights delayed vs not delayed
  + Flight delays by hour
  + Weather
  + Ram’s map
* Data wrangling and cleaning → will finalize these strategies next week
  + Data that is missing
  + Data that is missing but has a meaning
  + Data that is of poor quality
    - Summarize from multiple weather stations, potentially
  + Handling outliers

Action items

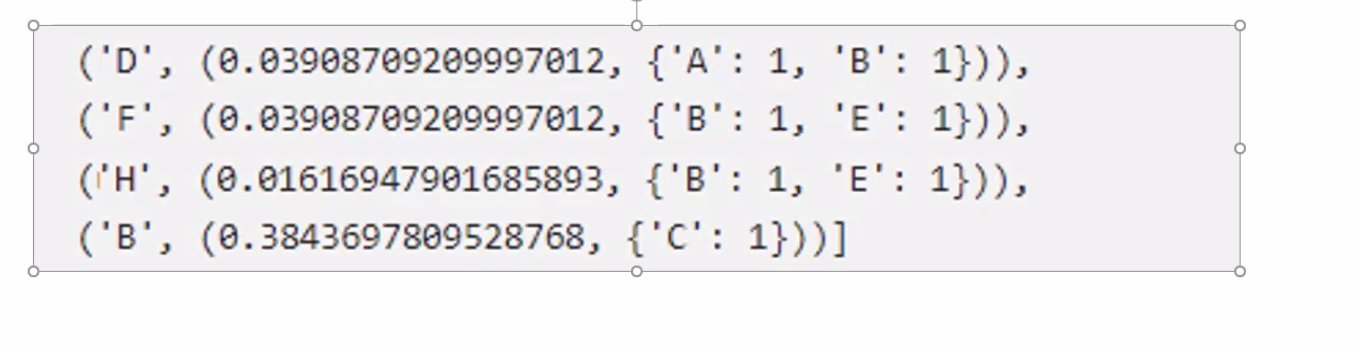
* ~~Jenny - Ask Lea about weather data for 2020 & 2021~~
* Ram - modify the “Flight Delay Prediction” features image to display which features we are and are not including

## 

## Mar 11, 2022 | [Lea's OH](https://www.google.com/calendar/event?eid=M2NvaDhyYW9kb2FpZTE3b2dzZmswNjNtZWkgamNvbmRlQGJlcmtlbGV5LmVkdQ)

Attendees:

Notes

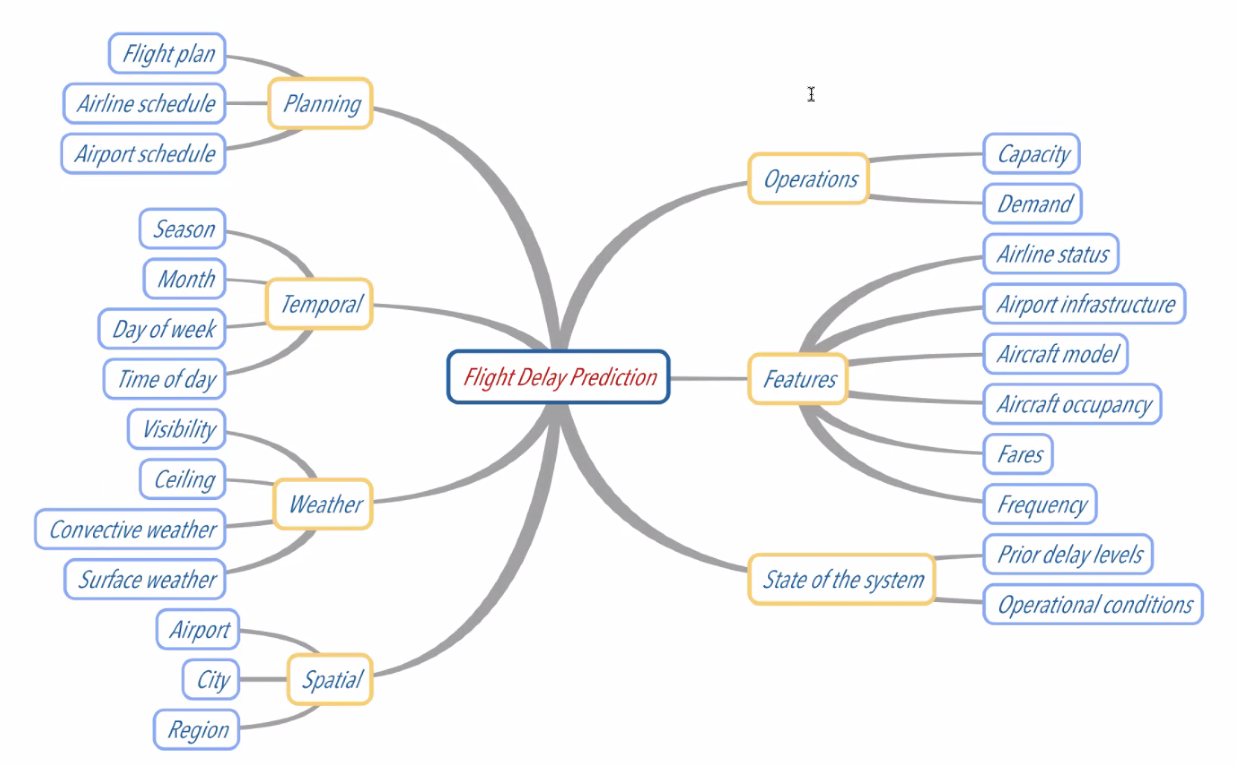
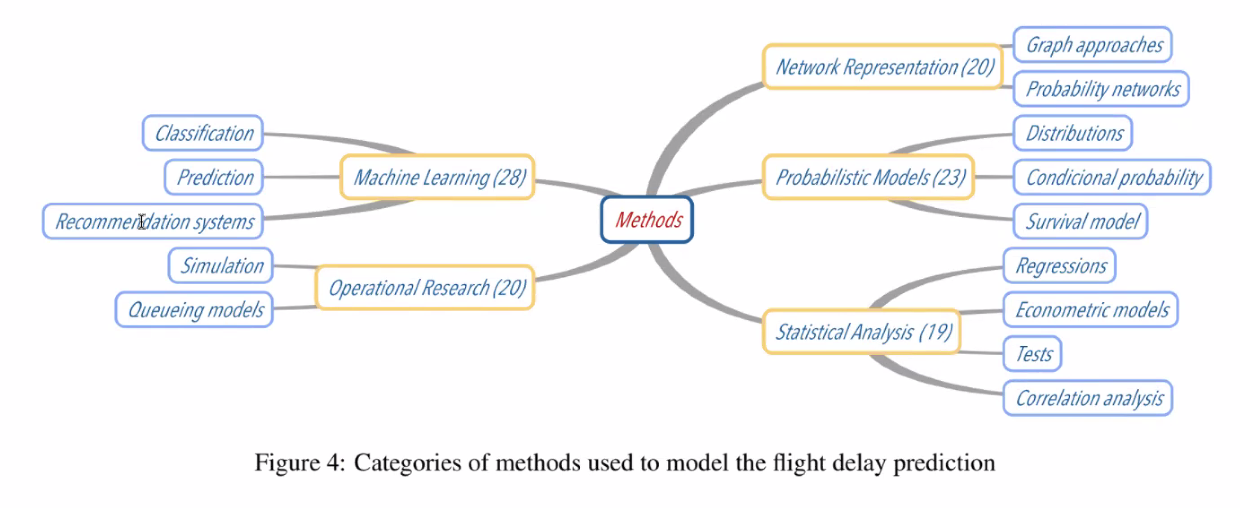
* EDA: focus on airline data, less so on weather data
  + Come up with features from weather data → dataset will be much smaller once you have features
    - Which features will you keep? Make the features, concentrate EDA on airline data
  + Then join with airline data and do more EDA
  + Might not use all weather data
* Don’t need to cache before writing to parquet
* Have all work in the same notebook but don’t need to actually run everything
  + When you push file to GitHub from DataBricks, it will be a .py file → can’t see what was executed
  + Jimi just wants to see that you’ve written code
* Station Dataset
  + Supplementary to get extra information for joins → ex. Join by distance to airport
  + Lea’s group did not use this dataset last year
  + Depending on how you join, you might not need this dataset
* CSV vs Parquet → we’re already using parquet so just answer the question generally
* Phase 1: Stretch goal to have some code, but mostly conceptual
  + Descriptive → no need for code yet (ex. Describe joins, don’t have to have it done yet)
  + Define the outcome: Are you doing a classification/regression? What’s the outcome of your model?
* Jimi likes to see code
* HW 5 Discussion
  + #6: Do we care about the number of times a given outlink occurs? Yes, this is the weight for the edge.
  + #7: Need to output a graph RDD with (node, (score, edges)) → can basically just update the decimal
    - Need to address dangling nodes in #7 → need to initialize the dangling nodes
* Final report → Ask Jimi what he wants, submit a notebook
* Jimi likes to see code even during presentations (not the 2-min videos)
  + Be able to walk through the code
* Jimi really wants someone to try something new!! Multi-step models, download new data
  + Need to download CSVs monthly but can still get the new data (2020, 2021) <https://www.transtats.bts.gov/DL_SelectFields.aspx?gnoyr_VQ=FGJ&QO_fu146_anzr=b0-gvzr>
* Consider taking away certain airports
  + Ex. focus on continental US flights → Hawaii, Alaska, Puerto Rico might not affect delays in continental US as much as others
  + See which airports are giving you trouble → do you actually care about these for your model?
* Files take a while to write to parquet
* Weather missing data
  + Do we care about the column?
  + If we don’t, then just get rid of it
  + What does the none mean if we do care about the column?
* Cross validation: 2015-2016 (train = 2015, validation = 2016), 2015-2017 (train = 2015-2016, validation = 2017), 2015-2018 (train = 2015-2017, validation = 2018)
  + PySpark Cross Validation → built-in only does random sampling
  + To preserve the time aspect of cross validation, need to write a custom cross validation or do it manually
* PageRank on Airports: <https://diybigdata.net/2019/12/airline-flight-data-analysis-airport-pagerank/>
  + GraphFrames doesn’t support weighted graphs → can use HW code
* Categorical data: too many categories = overfitting (ex. >20 categories, 24 for hours in day is okay but don’t do tail numbers)
  + Can make features based on tail number

Action items

## Mar 10, 2022 | [W261 Section 2 Team 11 - Weekly Sync](https://www.google.com/calendar/event?eid=N2pncmZidWdnNGphNDc3cHEzOTZzZWo0bGNfMjAyMjAzMTFUMDMwMDAwWiBqY29uZGVAYmVya2VsZXkuZWR1)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu)

Notes

* Keep starter\_nb\_fp as working notebook → each week fork it, clean it up, and submit to GitHub
* Read [carvalho\_review.pdf](https://drive.google.com/file/d/1r6Po1Oqh6X9UOUZXEcaYZuutCeLGFY4v/view?usp=sharing)
  + 3 Viewpoints → 3 different problems
    - Delay propagation
      * Graph-based approach for features
      * Given prior flight information, what’s the probability that this flight will be delayed?
    - Root delay → probability of a delay considering there hasn’t been one yet, the delay is starting in the chain
      * Typical classification/regression problem
      * Given a flight, what’s the probability that it is delayed?
    - Cancellation
      * Given that delays have happened and given this flight, what’s the probability that a flight will be canceled?
  + Researchers will focus on 1 viewpoint
  + Factors that play a role in flight delay:  
    
  + Models:  
    
* Read [A Comparative Analysis of Models for Predicting Delays in Air Traffic Networks.pdf](https://drive.google.com/file/d/1wPnv1PG5Tootj6KQ7UHcI20nGVzQO47c/view?usp=sharing)
  + Relate adjacent pairs in the network
* Graph ideas
  + Visualizing → use color/line thickness to display delay frequency
  + Dynamic plot → Where does the delay start? How does it change over time? (See the delay propagation)
* “Novelty” behind project
  + Can pull 2020/2021 data and see how the model performs under pandemic conditions?
  + Multi-Classification Problem (Ordinal Logistic Regression?)
    - Categorize delay >15 mins into multiple buckets
  + Include aircraft in prediction
* Weather data?
  + Bad weather may delay flights
    - Bad weather where and when?
      * Arrival
      * Departure
      * In real life, someone would check the weather forecast → we only have the measured weather information
  + Focus on weather data 2 hours before departure at departure location
    - Working on the arrival side could be feasible but a lot of work
      * Struggles: Timezone conversion
        + Should we convert time zones to UTC?
        + Standardize to a single time zone?
        + Airport + timezone + daylight savings (PDT vs PST)
        + Work with only local time
        + Weather data is in UTC
        + Map IATA Code to Timezone → Ram has dataset
  + Build model that predicts 2 hours before departure → we have weather data 2 hours before departure → we would ideally want the 2-hour-ahead forecast rather than the actual data 2 hours ahead
    - Actual weather 2 hours before at departure location
  + Build a weather forecasting model ? 👀 👀
    - Take T-2 through T-5 at arrival and departure → model may learn and try to use the data → proxy for weather prediction
    - Or just use the deltas between hours
* Flight preprocessing → what is the window 2 hours before? Use this to merge the weather data
* Adjust for daylight savings
* End of each phase should have working prototype
  + Ex. Simple: predict majority class
* Daily standups → What did you do the day before? What are you doing next? Are you blocked anywhere?
  + Past, Problems, Plans
  + 2-3x per week

Action items

## Mar 8, 2022 | [W261 Live Session](https://www.google.com/calendar/event?eid=MDRvcmpjNm1xZzJlM3UxcW5rdm5zaGMyZHEgamNvbmRlQGJlcmtlbGV5LmVkdQ)

Attendees:

Notes

* We’re still limited to 2 minutes for the video
  + Script what we want to say
  + Count the words (300-400 words)
  + “Three P” update: past, problems, plan
  + Script & rehearse
  + Focus on what’s important
* Phase 1: What is the problem you’re tackling? Who are the major stakeholders? Some envisioned challenges? Plan for future phases? What’s novel about what you’re doing?
  + What novelties will you explore?
* PageRank will be an important part of the project - Jimi
* Graphs for visualization
  + GraphFrames package in Spark
* Jimi says PageRank will be useful
  + Flights graph
  + Delayed flights graph
    - Popularity in delayed flights graph
  + Bias graph
* Jimi wants to see novelty
  + Ex. get new data
  + Ex. explore different loss functions → is there something else?
  + Multi-task loss function (first classify, then regress)
  + Old model + most recent model approach = blend the two

Action items

* Create script for Phase 1 Presentation

## Mar 7, 2022 | [w261 Team 11 Meeting](https://www.google.com/calendar/event?eid=MWxocWI5cGVuaTY0ajQ4a3NmbmtpcjMyZzUgamNvbmRlQGJlcmtlbGV5LmVkdQ)

Attendees: [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jennifer Conde](mailto:jconde@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* Discuss problem statement and perspective for project
* Work distribution
* Phase I deliverables
* Question formulation – Predict departure delay (binary prediction) at least two hours prior to a flight departure so that airlines can notify passengers
* Potential Perspectives – Who are we?
  + Airlines – Wants to know delay predictions to know how to reallocate resources (ex. Planes, passengers, etc.)
    - Corrective actions
    - Operational efficiency (ex. Loading food, cargo, crew allocation, rebalance based on plane availability)
    - Arrange for other passenger accommodations (ex. Restaurant/hotel coupons) to maintain customer satisfaction
    - Keep customers happy so they keep coming back
    - Financial implications
  + Airport – Wants to know delay predictions to improve airport experience of passengers, operations (ex. Coordinating arrivals/departures, arranging for airport staff to prepare for arrivals/departures)
    - Safety – need to limit the number of planes arriving/leaving at the same time
    - Air traffic control
  + Flyer Experience Consultant – Wants to know delay predictions so they can plan ahead
  + Startup airline 🤑 – an airline that is never delayed 👀
  + Prediction App for Passengers
    - A customer might not find a binary classification very useful
    - Future work could be in predicting the amount of delay (regression)
    - Can sell the technology to airlines, travel companies, Google, etc. in addition to individuals
  + What’s the value add for each “client”
  + The perspectives are not mutually exclusive
* Business Justification – Why are we doing this?
  + Predicting flight delay = airlines, airports, and passengers benefit
* Metrics to evaluate quality of prediction
  + Accuracy
  + Precision/Recall/F1 Score
  + Impact of having incorrect predictions (false positives vs false negatives)
* Perspective of Airline:
  + False positive (predict delay but it’s not actually delayed): Any corrective action has gone to waste, could still give airlines a warning that they need to look into it, passengers may miss the flight
  + False negative (predict no delay but it’s actually delayed): Airline will be caught off guard, customers angry because their flight is late
  + Depends on how information is used → at least the false positive is a warning
* Perspective of Prediction App
  + False positive: passengers may miss the flight
  + False negative: customers angry because their flight is late
* Based on the current status quo → false positives might have bigger impact
  + False negative might not damage the airline business since this is how things currently are (might be angry at the prediction app though)
* Explain data (EDA). Identify any missing or corrupt (i.e., outlier) data
* CSVs = bad, use parquet instead
* Join relevant datasets → diagram of keys/tables and merged
  + Progress on joining airline data and stations data
  + Need to join airline and weather
  + Describe what tables to join
  + Describe the workflow for achieving the joins (what are the keys, type of join)
  + Steps to deal with potential missing values
* Train/Test/Validation Split: Split by year
* Deliverables:
  + Slides - Business Problem, EDA, Joins, Challenges & Next Steps
  + Video
  + Notebook
* Airport codes & joins
  + Ram has pulled in another dataset
  + Includes airport size
  + Map IACO code to 3-letter airport code
  + Missing weather station data for 8 airports
    - ws\_id missing for San Juan, Guam → several flights from Chicago to these destinations
    - Can probably discard, overall % is pretty small
* Tail number of aircrafts
  + Some null values in TAIL\_NUM → could this be a canceled flight?
* Arrival and departure times are strings and in local times
  + Need to figure this out for weather information
* DEP\_TIME\_BLK and ARR\_TIME\_BLK → compute weather information in hourly chunks and map this to weather

Action items

* ~~Dante - Ask Luis/Jimi if we can get 30 seconds extra on video or have only 4/5 presenters (concern about crammed presentation)~~
* Fill in final report with Phase I deliverables
* Complete slides
* Jenny & Joy - EDA
* Dante (& Joy) - Problem statement and evaluation metrics
* Dante - CSVs vs Parquet
* Ram - Joins
* Ram - Train/Test Split

## Mar 6, 2022 | [w261 Team 11 Meeting](https://www.google.com/calendar/event?eid=XzhkMWs4YzFrODUwajRiOXA4ZDFqY2I5azZjcWplYjlvNjEyamFiYTQ4Z3JqaWhhMTcwcDNpaGkzNmcgamNvbmRlQGJlcmtlbGV5LmVkdQ)

Attendees: [Jennifer Conde](mailto:jconde@berkeley.edu) [Dante Malagrino PhD](mailto:dantemalagrino@berkeley.edu) [Ram Senthamarai](mailto:ram.senth@berkeley.edu) [Jeffrey Adams](mailto:jeffrey.adams@berkeley.edu) [Joy Moglia](mailto:jmoglia@berkeley.edu)

Notes

* For now work in the team space, then move into GitHub after
* Get type of airplane through tail number?
  + <https://www.faa.gov/licenses_certificates/aircraft_certification/aircraft_registry/releasable_aircraft_download/>
* Use time of day as a feature
* Arrival time of inbound flight that the airplane would be used for
  + If prior flight is late, the current flight is late
* Are times adjusted for time zones? —> local time
* Get a few planes (TAIL\_NUM) —> pull all records (not from limited set)
* Canceled flights
  + Is a cancellation a delay with infinite time??
  + We do care about cancellations
  + Limit scope to only delays?
  + Many cancellations on certain routes would likely be good indicators of delays
  + Why are flights canceled?
    - Not enough passengers, consolidate flights —> prior cancellations for this reason might not predict the delay of the next flight
    - Bad weather
  + Maybe not a great feature
  + Would be interested to predict cancellation —> if not canceled, will it be delayed? Why not?
  + Ex. T-4 to T-2 —> how many flights are canceled?
    - Aggregate and use as a feature
* Use Cause of Delay in EDA
  + Out of causes of delays, which are most common?
  + Matching inbound/outbound flights
  + NAS delays —> non-extreme weather (ex. not blizzard, hurricane, etc.)
* Top airlines with most delays
* Diverted flights
  + Typically diverted during flight to closest suitable airport
  + Ignore these fields
* Mapping airport to closest weather stations
  + Some weather stations are already mapped to airports —> text search match?
    - Probably won’t be 1:1 but still helpful
* How do we map airports to weather stations?
* How can we get weather from 2 hours before departure?
  + Need weather information relevant to departure
  + Need logic to get correct weather
  + Any weather information < 2 hours from departure —> can’t use
  + How many weather observations do we use?
    - Change in weather —> ex. dynamic weather might impact departure
    - + How do we compute this?
* Can use PCA/regularization for feature selection
* In reality, airlines probably use forecast data but we’re working with actual data
* Can use SparkSQL to run queries
* Skewed data
  + Might need to do bootstrapping to make sure the model learns properly
* Missing data —> diverted/canceled flights
  + Confirm this
* Delay group of incoming flight —> could impact delay of current flight
  + Delay of inbound flight
  + Scheduled departure time vs arrival time of inbound flight
* Airlines can also swap planes and change schedule for a particular plane
  + Airlines can reroute flights to prevent delays or use other airplanes available

Sample features to consider

* Time of day
* Type of airplane
* Number of cancellations previously in the day
* Weather
* Is the plane delayed from the prior flight?

Key Foreseeable Challenges

* Mapping airport to weather station
* Getting weather from 2 hours before departure
  + Getting a change in weather
* Using PCA/regularization for feature selection

Action items

* ~~Jenny - Set up Google Drive folder~~
* ~~Meet tomorrow (3/7) to further discuss and divide work~~
* Look into matching tail number to airplane type
* Continue EDA