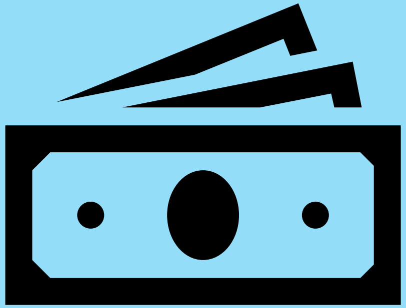
TripActions

Predicting Flight Delays

Group 2B
DDT & A.I. II
16.04.20

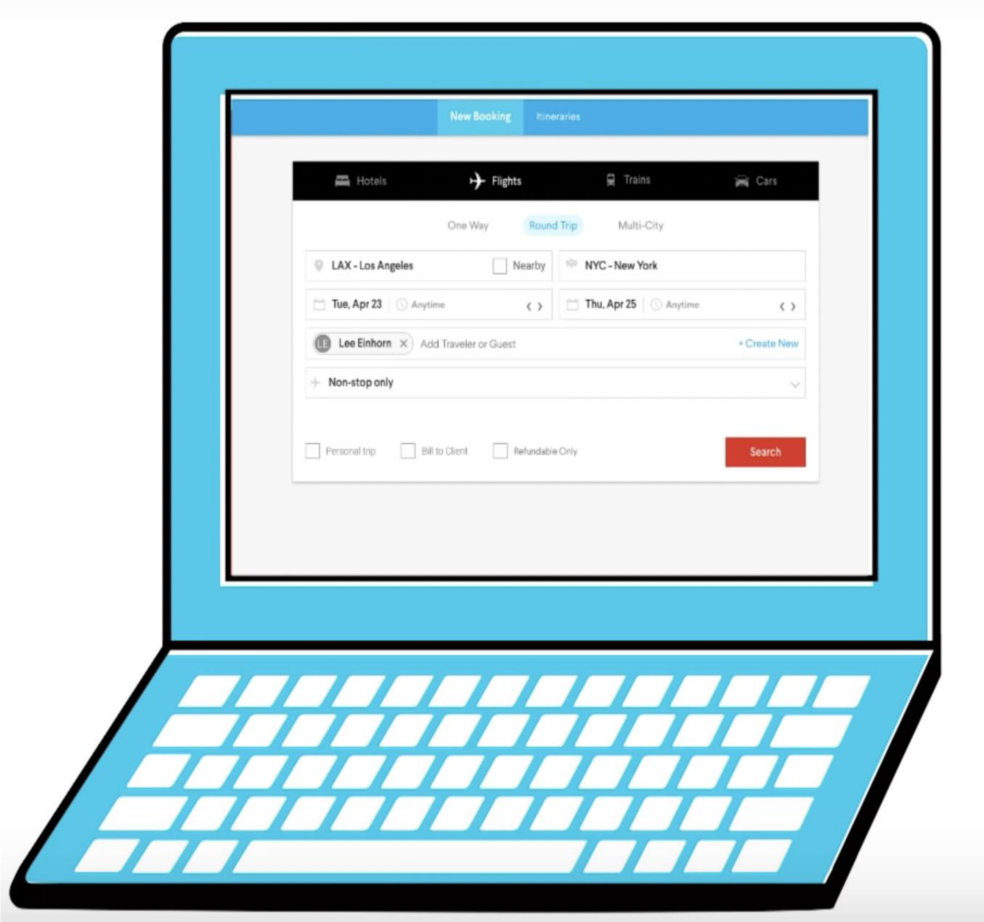


TripActions

A platform that takes care of business and your travelers.

TripActions ensures business travel continuity and the safety of your traveling employees.

Using the right data and developing useful insights, TripActions provides a platform that enables companies to build, manage and scale a world-class corporate travel management program with ease.



Business-Case Definition

TripActions' Story

TripActions was **founded** in **May 2015**, receiving a **$4million seed** round.

TripActions used its earnings to **expand out of the United States**. Opening up offices in Amsterdam, London and Sydney.

2015 2016 2017 2018 2019 2020

TripActions **grew rapidly** in the 2 years following its founding, securing contracts with many large corporates and airlines in the United States.

TripActions **is currently valued at $4billion**. The company is constantly looking for ways to improve its product offering. This attitude is why they ranked as one of Fast Company's most innovative companies in travel for 2019.

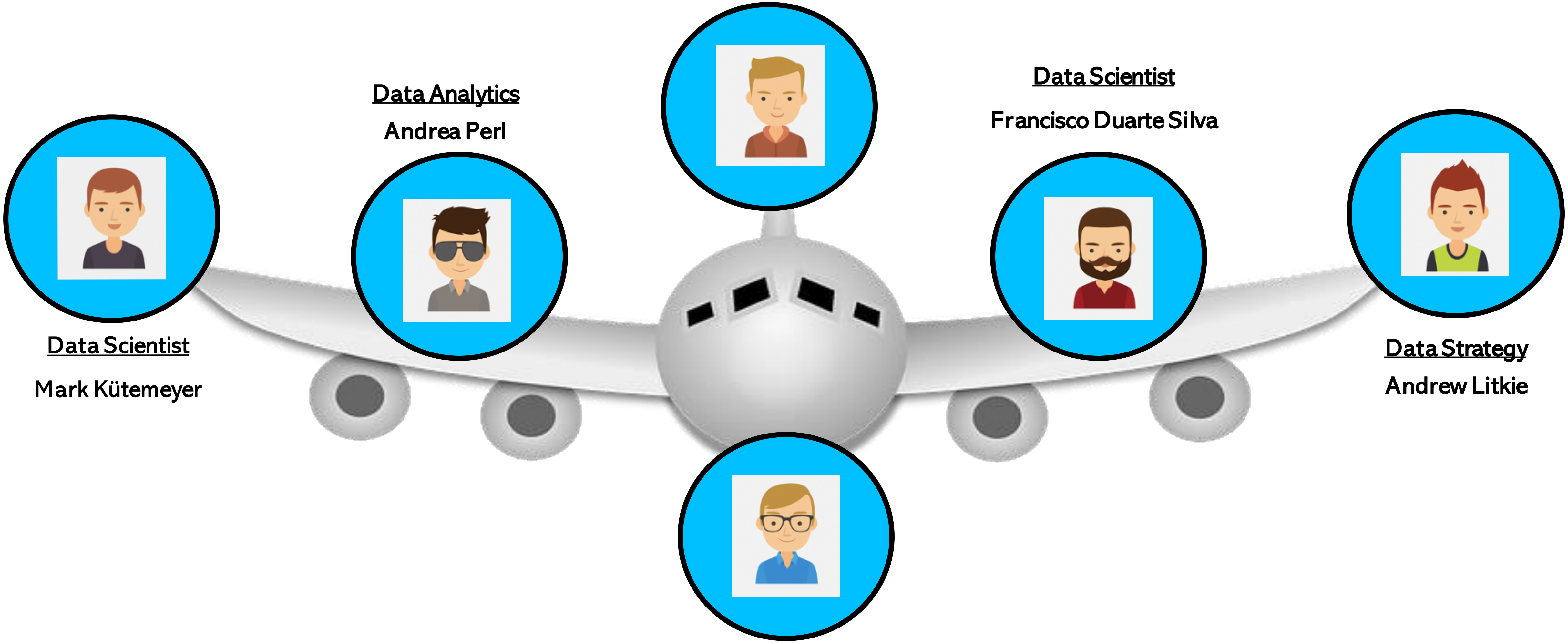
Ariel Cohen, the CEO of TripActions, has recently announced that he is **employing the help of a data driven transformation company, known as The A Team**, in order to help TripActions stay ahead of competition.

The A Team

Chief Data Officer
Daniel Weirig

Data Analytics
Andrea Perl

Data Scientist
Francisco Duarte Silva



Data Scientist
Mark Kütemeyer

Data Strategy
Andrew Litkie

Data Architect
Peter Bruins

A Project in 4 Sprints

ALL
Team PINK
Team ORANGE
Team GREEN

Kick-Off 10.02	Review 1 24.02	Review 2 09.03	Review 3 30.03	Final Rev. 12.04

	Sprint 1 10.02. - 24.02.	Sprint 2 25.02 - 09.03.	Sprint 3 10.03. - 30.03.	Sprint 4 31.03. - 12.04.
Business-Case Definition	Company Research; Decide on project; Calculate BIZ potential; Industry research; Datasets/ Resources			
Data Preparation	Data Exploration; Pre-processing 1	Feat. Engineering 1; Pre-processing 2; Feat. Engineering 2	Add Full-year data; Create final Dataset	
Model Creation		First models in Python; Try models with new dataset in python	Optimize models with new data in Python; Push Recall/ Precision; Optimize models with new data in BigML; Create Different Models for KPIs	Decide final KPI metric; Decide final model
Model Interpretation		Analyze Accuracy	Recomme nder concept; Build recommender 1.0	Finalize recommender
Model Implementation		Create front-end draft	Establish Tabpy connection 1; Build front-end skeleton	Establish Tabpy connection 2; Implement recommender; Polish front end

Building the Vision

Must Have Data

Flight Delay Data for 2019 including:

Was the flight delayed?
Origin and Destination airports
*Arrival time & d*eparture time
Airline

Enrichment Data

Additional Flight Data for 2019 including:

Taxi in and Taxi out duration
Date (Month and day of the week)
Distance travelled

Business Objectives

Build a flight recommender system based on delay probabilities and flight preferences

Acceleration Bricks

Data engineering
Forecasting (Ensemble Model)
Interface design using Tableau

Parameters

Arrival Time
Origin & Destination
Airline
Time threshold

Use Cases Prioritization

1 **Business travelers**
High frequency
High spending capacity

2 **Business travelers**
Average frequency
Average spending capacity

3 Tourists
High frequency
High spending capacity

4 Tourists
Average frequency
Average spending capacity

Business-Case Definition

Defining the Business Case

Sprint 1 2 3 4

Description of Opportunity

Travel management platforms currently have no indication for customers on arrival delays. As a fast growing startup, TripActions can use a flight delay predictor to be the go-to travel management platform for business travel and differentiate itself from.

Industry

Online travel management

Value / ROI Calculation

Revenue per corp. customer	$ 200K
Revenue estimated at $20 million (2019)	
Company has around 1000 customers (2019)	
Increase in customers	5%
Value generated	**$ 1M**
Investment	$ 450K
Team of six developers	
Avg. salary per developer in California is 75K	
ROI (first year)	**220%**

Model

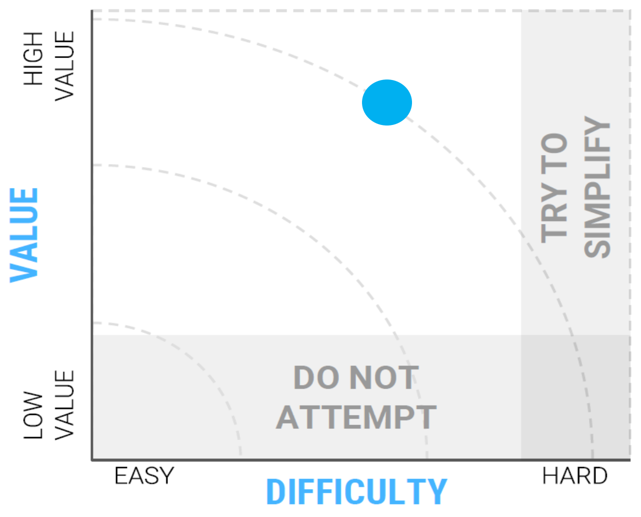
We are aiming to build a classification model, predicting delays using 10+ variables regarding flight and airline data

Context

Travel management companies will need to differentiate in light of corona pandemic

Opportunity Size

Global business travel expected to rise to $1.6 trillion by 2020



Implementation

We are aiming to build a recommender system showing flights with low chances of delay according to several criteria:

1. Desired arrival time and flexibility (hour threshold)
2. Date
3. Origin and destination airport
4. Desired Airline

System will be built using Tableau

Source: Owler, TravelPerk

Preprocessing

Merge monthly 2019 flights data from the US Bureau of Transportation Statistics

Explore the data and drop NA’s

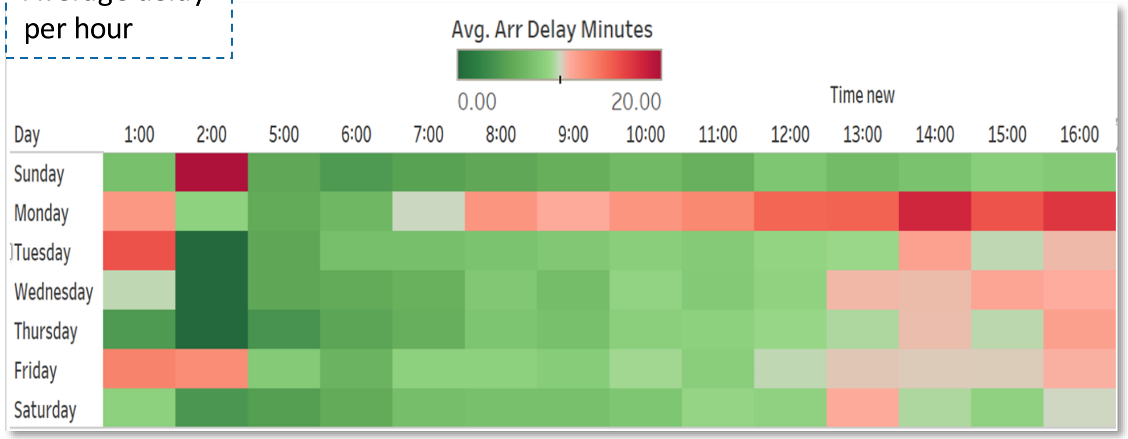
Change departure time to departure hour & date to day of week to extract value

Consider top 10 busiest airports for business case

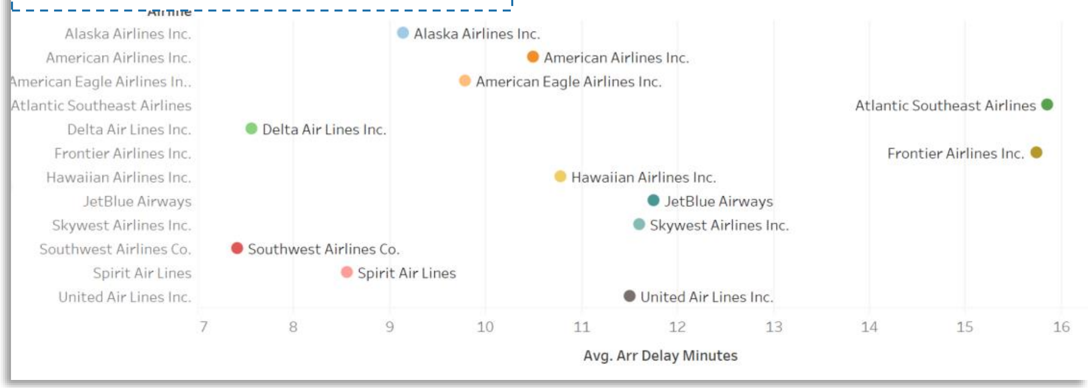
Define the binary target: Delays
Delayed if >=15 minutes behind schedule

Dataviz

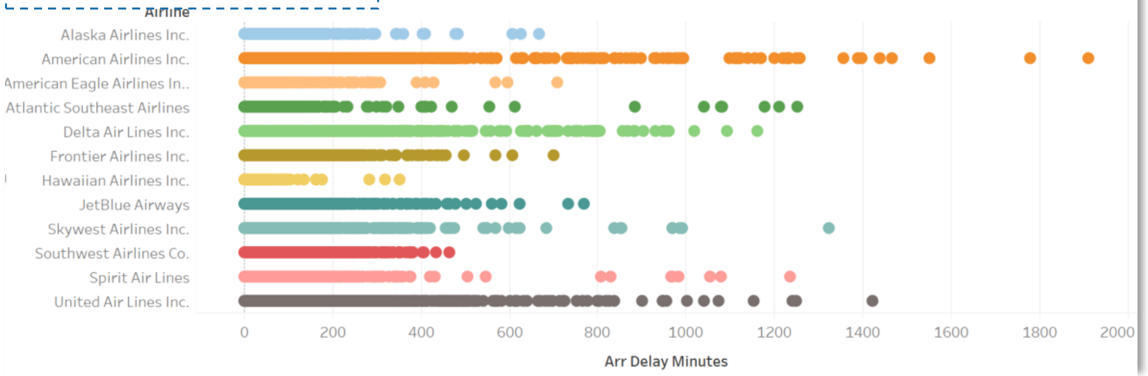
Average delay per hour



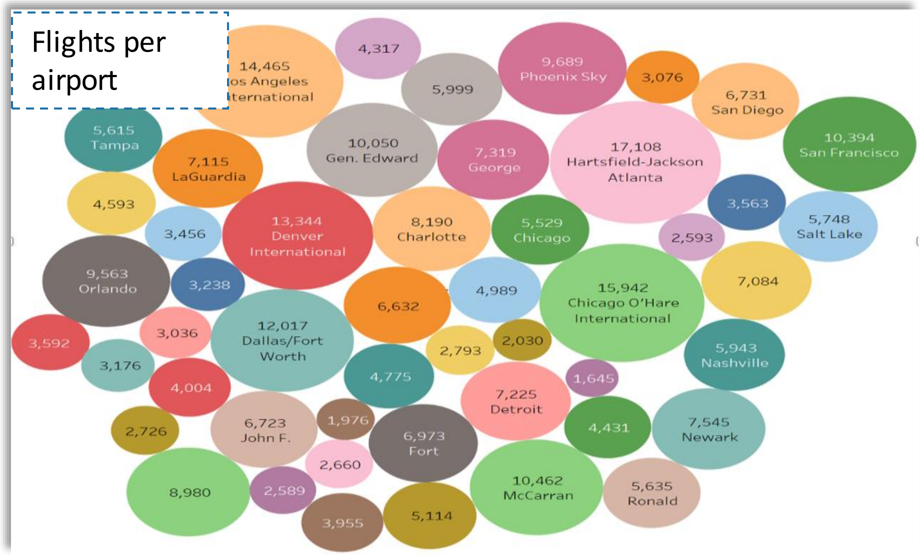
Average delay per airline



All delays per airline



Flights per airport



Data Preparation

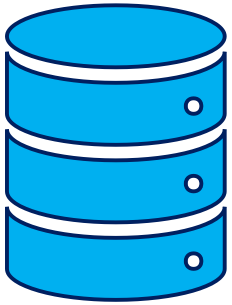
Feature Selection and Data Optimisation

Sprint

1 2 3 4

- Data set contains over 150 features
- Drop 120 features which are not relevant for project
- Drop 20 features based on tableau analyses and model tests
- Create new features based on insights
- Use anomaly detection to delete 500 biggest anomalies from train data
- Let’s Build the model!

Final Data set



1 target

4 numerical features

6 categorical features

475,706 observations

MONTH	ABC	475,706	0	0
DAY_OF_WEEK	ABC	475,706	0	0
OP_CARRIER	ABC	475,706	0	0
ORIGIN	ABC	475,706	0	0
DEST	ABC	475,706	0	0
ARR_DEL15	ABC	475,706	0	0
CRS_DEP_TIMEHour	123	475,706	0	0
Speed	ABC	475,706	0	0
avg_tax_out	123	475,706	0	0
avg_tax_in	123	475,706	0	0
airline_delay	123	475,706	0	0

Building the model

Models tried in Python and BigML:

Trees
- Random Forest
- Bootstrap Forest

Boosted Trees
- Xgboost
- Lightgbm
- Catboost

Deep Learning
- ANN

We attempted several models on Jupyter Notebook

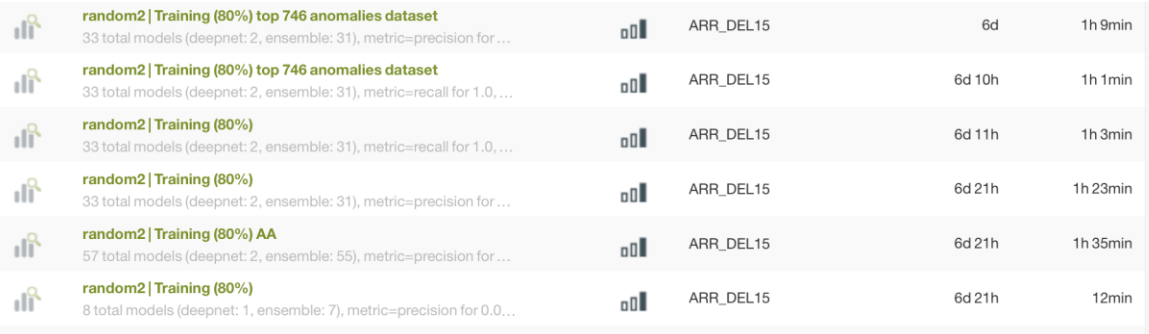
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer

cv = StratifiedKFold(n_splits = 5, shuffle=True, random_state = ra

rdf = RandomForestClassifier(random_state = random_state)
scoring = {'Recall': make_scorer(recall_score),
           'f1_score': make_scorer(f1_score)
          }

params = {'max_depth': [15, 20, 25, 30],
          'min_samples_split': [3, 5, 7],
          'min_samples_leaf' : [2, 4, 6],
          'n_estimators' : [350, 400, 450]
         }

grid_clf = GridSearchCV(estimator = rdf, param_grid = params, cv =
```

```
def evaluation_times(X,y,n_splits=10, shuffle=True, random_state=0):

    X_std = StandardScaler().fit_transform(X)
    X_minmax = MinMaxScaler().fit_transform(X)

    kfold = KFold(n_splits=n_splits, shuffle=shuffle, random_state=random_state)

    features = [('Normal',       X),
                ('Standardized', X_std),
                ('MinMax',       X_minmax)]

    models = [('LogisticRegression',        LogisticRegression(solver="lbfgs",multi_class="auto")),
              ('KNeighborsClassifier',      KNeighborsClassifier()),
              ('Naive Bayes',               GaussianNB()),
              ('DecisionTreeClassifier',    DecisionTreeClassifier()),
              ("RandomForestClassifier",    RandomForestClassifier()),
              ("GradientBoostingClassifier", GradientBoostingClassifier())]

    results = []

    for model in models:

        for feature in features:

            res = cross_val_score(model[1], feature[1], y, cv=kfold)

            [results.append((model[0],feature[0],model[0]+'_'+feature[0], r)) for r in res]

            print(res)

    result = pd.DataFrame(results,columns=['Model','Feature','FeatureScaled_Model', 'Result'])

    return result

evalu = evaluation_times(X,y,n_splits=10, shuffle=True, random_state=0)
```

```
sampler = SMOTETomek(sampling_strategy=1, random_state=42,n_jobs =2)

model = XGBClassifier(random_state = 42,
                      n_jobs = 4,
                      scale_pos_weight =1.2,
                      learning_rate =0.05,
                      n_estimators=1000,
                      max_depth = 7,
                      min_child_weight = 1,
                      gamma=0.4,
                      subsample=0.85,
                      colsample_bytree=0.95,
                      reg_alpha=0.005,
                      objective= 'binary:logistic',
                      seed=42)

steps = [('sampler', sampler), ('model', model)]

pipeline = Pipeline(steps=steps)

pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)
```

```
els import Sequential
ers import Dense

 the ANN
equential()

nput layer and the first hidden layer
(Dense(units = 24, kernel_initializer = 'uniform', activation = 'relu', input_dim = 46))

econd hidden layer
(Dense(units = 24, kernel_initializer = 'uniform', activation = 'relu'))

ird hidden layer
(Dense(units = 24, kernel_initializer = 'uniform', activation = 'relu'))

utput layer
(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))

# Compiling the ANN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# Fitting the ANN to the Training set
classifier.fit(X_train, y_train, batch_size = 5, epochs = 2)
```

We attempted some more in BigML



random2 | Training (80%) top 746 anomalies dataset 33 total models (deepnet: 2, ensemble: 31), metric=precision for…	ARR_DEL15	6d	1h 9min
random2 | Training (80%) top 746 anomalies dataset 33 total models (deepnet: 2, ensemble: 31), metric=recall for 1.0,…	ARR_DEL15	6d 10h	1h 1min
random2 | Training (80%) 33 total models (deepnet: 2, ensemble: 31), metric=recall for 1.0,…	ARR_DEL15	6d 11h	1h 3min
random2 | Training (80%) 33 total models (deepnet: 2, ensemble: 31), metric=precision for…	ARR_DEL15	6d 21h	1h 23min
random2 | Training (80%) AA 57 total models (deepnet: 2, ensemble: 55), metric=precision for…	ARR_DEL15	6d 21h	1h 35min
random2 | Training (80%) 8 total models (deepnet: 1, ensemble: 7), metric=precision for 0.0…	ARR_DEL15	6d 21h	12min

Model Interpretation

Recall vs Precision

Sprint 1 2 3 4

Model Evaluation

Which metrics should we opt for?

1. **Recall:** Minimize False negatives
   True class would be delay
2. **Precision:** Minimize False positives
   True class would be on-time

Metric Results

	Recall	Precision	AUC
True Class Delays	**64.4%**	**85.2%**	**68.3%**
True Class Non-delays	**68.9%**	**85.4%**	**68.2%**

Ensemble optimized for recall of delays

+

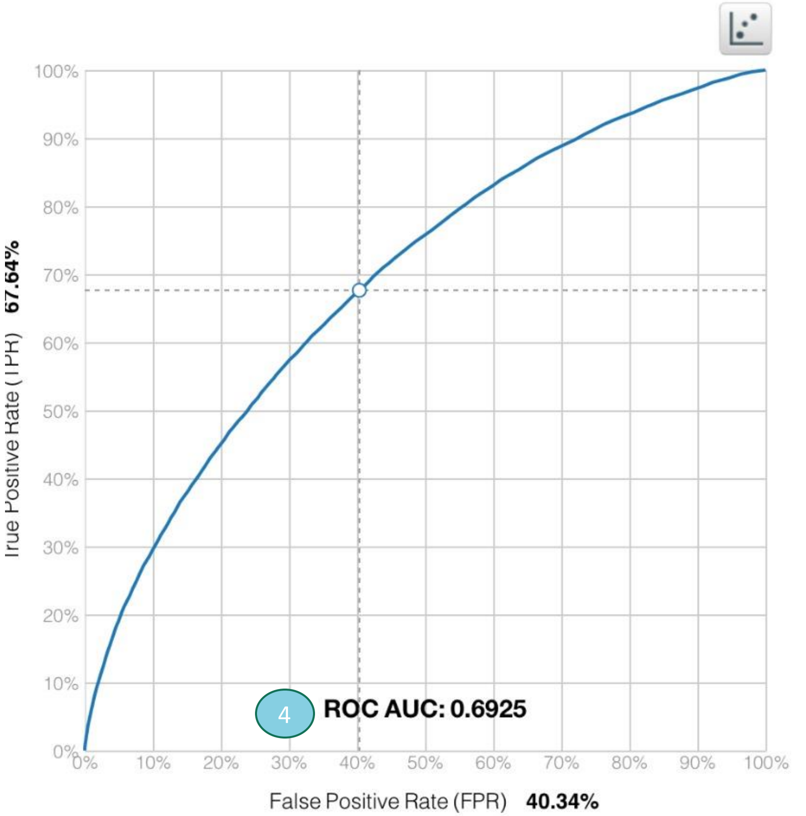
Ensemble optimized for precision of no delays

Final model: Fusion of two ensembles

Business Case: Recall vs. Precision

TP FN FP TN

ACTUAL VS. PREDICTED	1.0	0.0	ACTUAL	RECALL	F	Phi
1.0	14,818	1 7,089	21,907	2 67.64%	0.45	0.23
0.0	29,540	43,695	73,235	59.66%	0.70	0.23
PREDICTED	44,358	50,784	95,142	63.65% AVG. RECALL	0.58 AVG. F	0.23 AVG. Phi
PRECISION	33.41%	3 86.04%	59.72% AVG. PRECISION	61.50% ACCURACY		



PNG

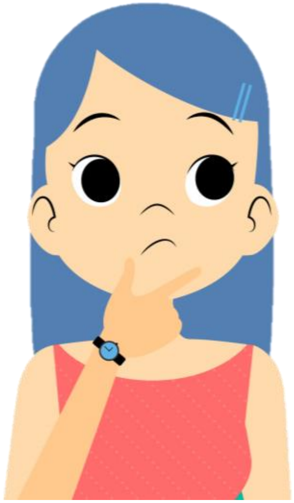
61.5% Accuracy	0.4472 F-measure	
33.4% Precision	67.6% Recall	0.2304 Phi coefficient
40.3% FPR	46.6% % positive instances	145.1% Lift
27.6% K-S statistic	0.2313 Kendall's Tau	0.2833 Spearman's Rho

1. False Negatives (7,089)
2. Recall Metric (67.6%)
3. Precision Metric (86 %)
4. AUC Metric (0.70)

What does the customer want?

… from a recommendation system I want to:

1. Be shown the flights with the lowest chance of being delayed…
2. … not arriving later than a certain time …
3. … but also not arriving too early …
4. … and preferably from my favorite airline



Taking into account the customer's specifications, we have built a recommender system:

Using python for the hidden workings of the system

Using Tableau to build a user-friendly interface

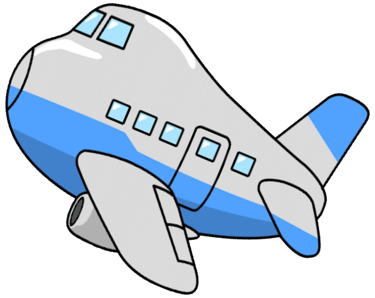
Model Implementation

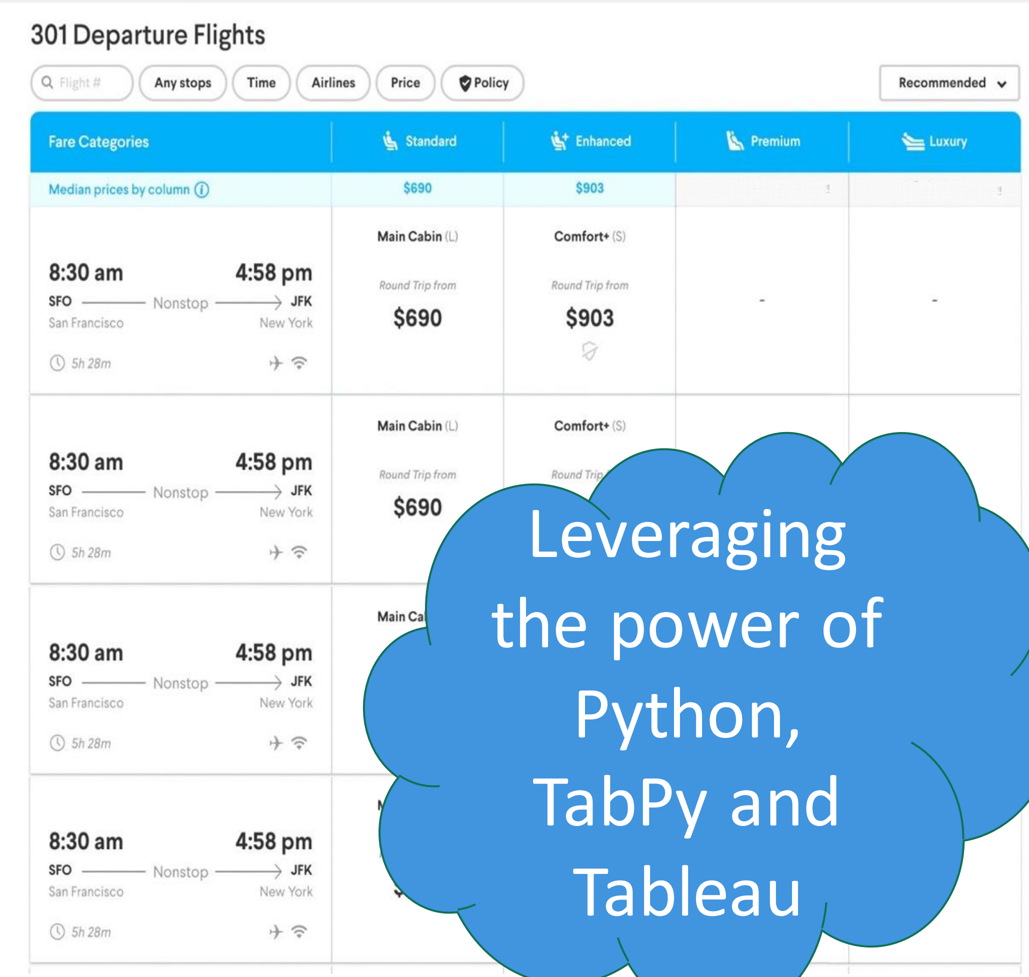
Integration of the Recommender System

Sprint
1 2 3 4

The goal is to smoothly and successfully integrate the delay predictor into TripActions business model.

The premium and luxury columns are empty. Why? Not available for domestic flights!





Leveraging the power of Python, TabPy and Tableau

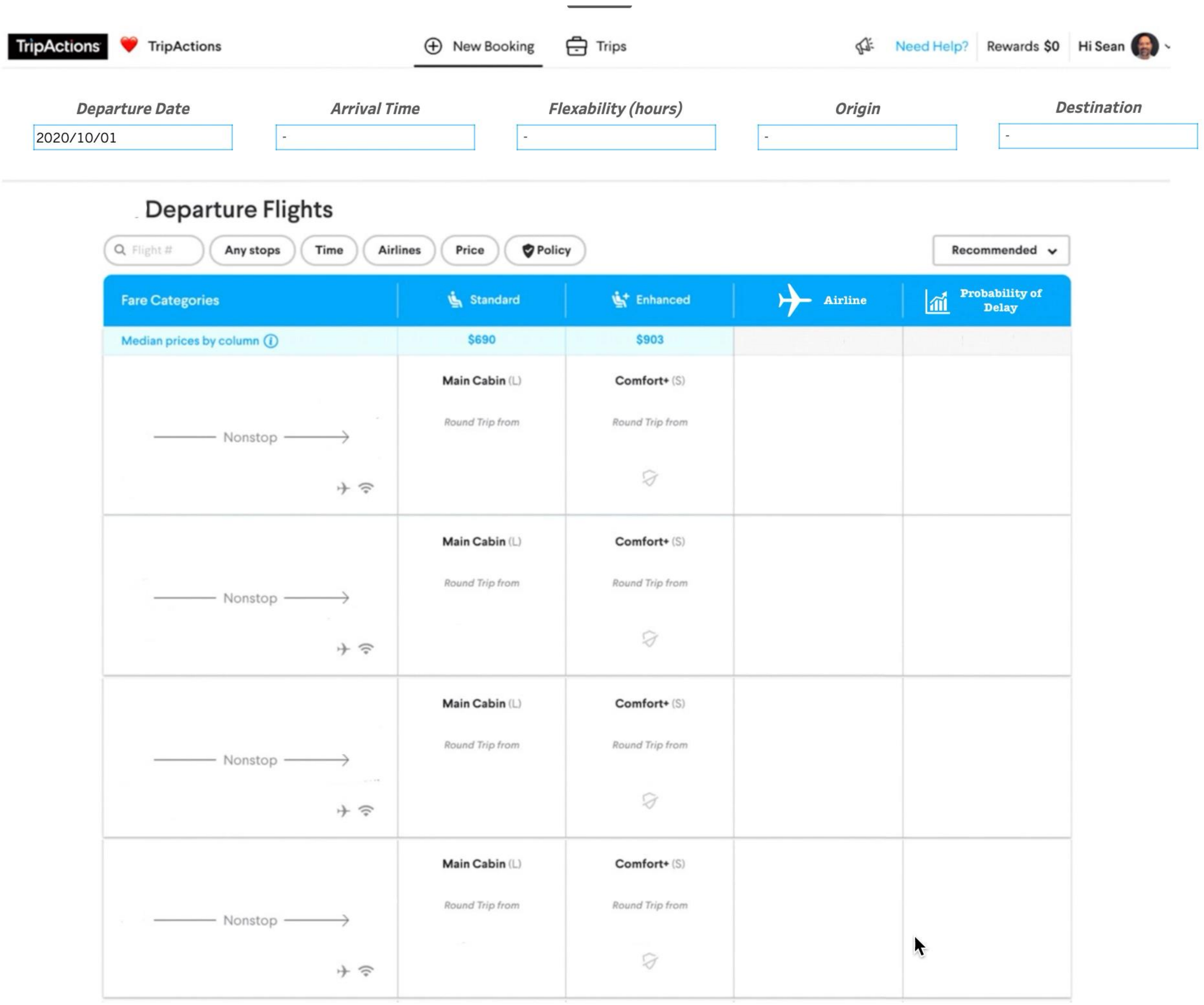
Therefore these two columns can be replaced to show the airline and the probability of delay.

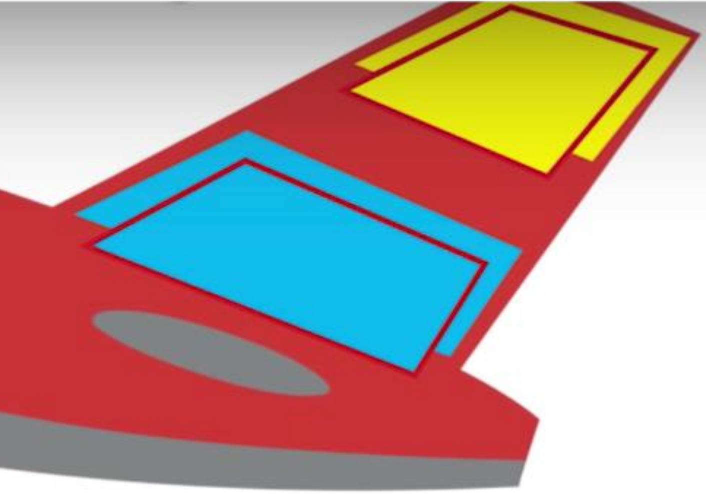
Putting the Recommender to use

A business traveller wants to book a flight on the 20th of November 2020.

Wanting to arrive at 13:30 pm latest and her schedule allows for flexibility of arriving 2 hours before this time.

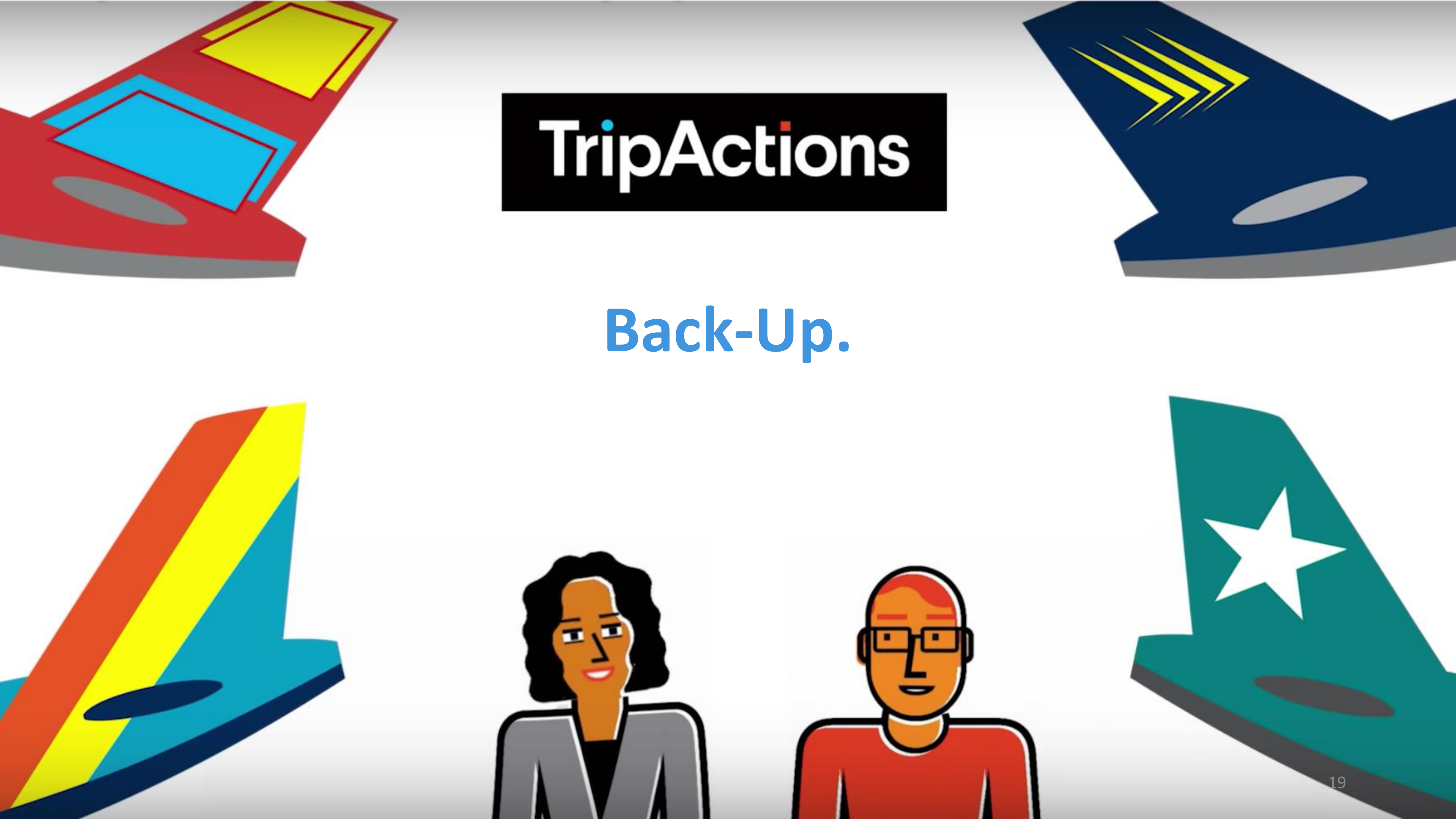
And travelling from Chicago to San Francisco.





TripActions

Thank you.
Do you have any questions?



Back-Up.

GitHub Link

[Github with the whole project]

The ML Canvas is now complete

The machine learning canvas

Designed for: Tripaction | Designed by: The A team | Date: 13-04 | Iteration: 4

Decisions
How are predictions used to make decisions that provide the proposed value to the end-user?

- Precision and recall vs accuracy
- Minimum precision for no delay prediction of 77% required

ML task
Input, output to predict, type of problem.

- Binary classification based on multiple numerical and categorical features
- Unbalanced dataset True/False ratio 2:7

Value Propositions
What are we trying to do for the end-user(s) of the predictive system? What objectives are we serving?

- Predict chance of flights being delayed or not, to make sure business traveler are on time as much as possible
- Model outperforms random decision of always no delay prediction, which is right 77% of the time. Added value is difference between model and 77%.

Data Sources
Which raw data sources can we use (internal and external)?

- U.S. Bureau of Transportation Statistics (part of United States Department of Transportation)

Collecting Data
How do we get new data to learn from (inputs and outputs)?

- Visualize features in tableau to gain wide overview
- Input features with slight changes to understand relation with delay

Making Predictions
When do we make predictions on new inputs? How long do we have to featurize a new input and make a prediction?

- Model receives prediction request from recommender system which is imbedded into tableau to create a live feedback experience to userbase

Offline Evaluation
Methods and metrics to evaluate the system before deployment.

- Cost of False Negative
- Cost of False Positive
- Minimum benchmark accuracy

Features
Input representations extracted from raw data sources.

- Airport
- Airline
- Time
  - DotW
  - Hour
  - Month
- Taxi in/out time

Etc.

Building Models
When do we create/update models with new training data? How long do we have to featurize training inputs and create a model?

- Start with data from one month and optimize model for the data
- Input data of whole year and make slight adjustments

Live Evaluation and Monitoring
Methods and metrics to evaluate the system after deployment, and to quantify value creation.

- Update database with new monthly data and check validity

Key takeways

- Understand the implications of Precision/ Recall on the business case and its implication to the tradeoffs being made in the model
  - How "expensive" a false negative is
  - How "expensive" a false positive is
  - What is minimum accuracy required for the product
- Know how to cope with unbalanced data

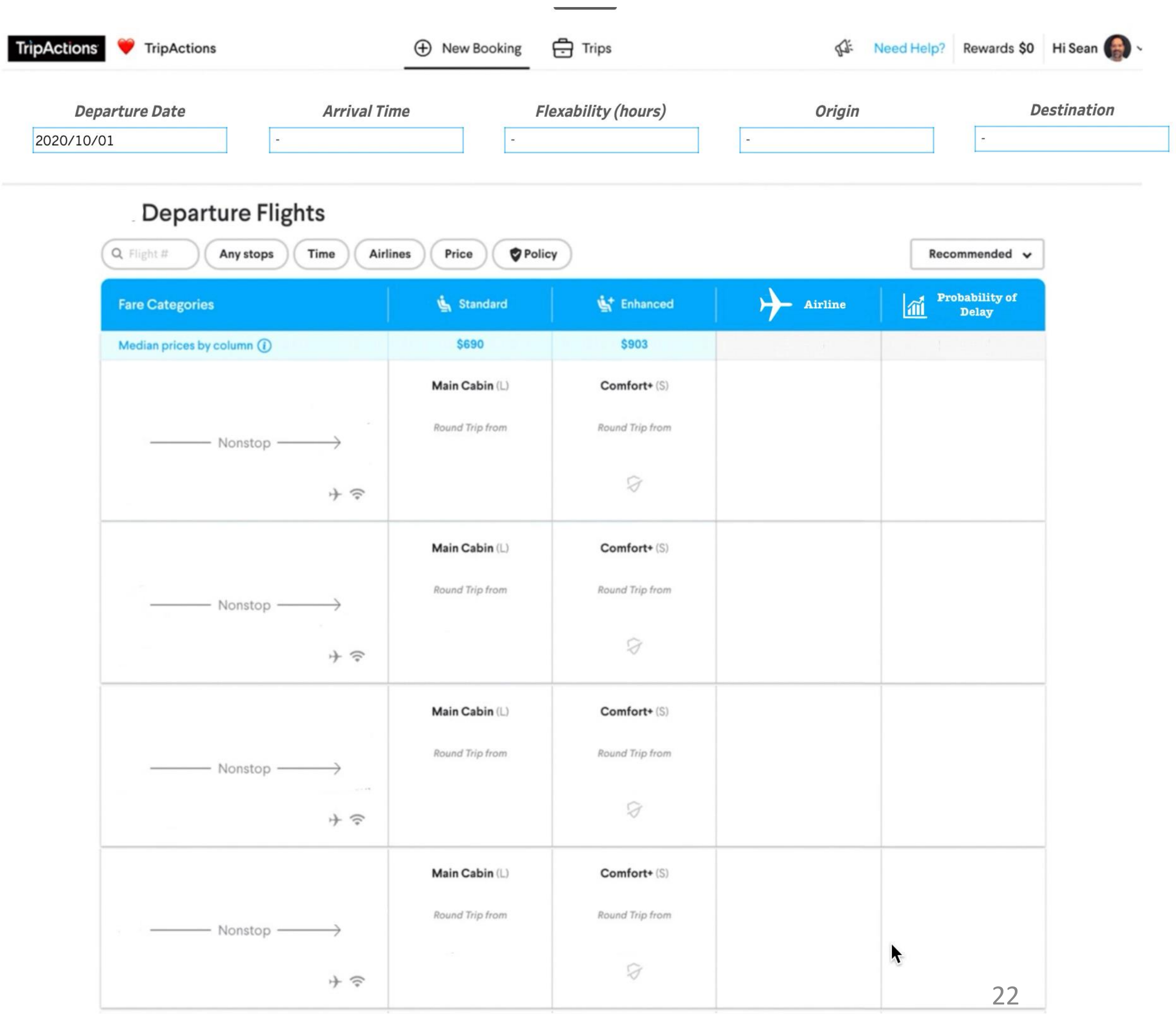
Model Implementation

Putting the Recommender to use e.g. #2

A business traveller wants to book a flight on the 2nd of November 2020.

She wants to arrive at 20:30 pm latest and her schedule allows for flexibility of arriving 2 hours before this time, but no earlier.

She is travelling from from Dallas to Denver. The reason for this trip is for an important meeting with a new client.



Model Implementation

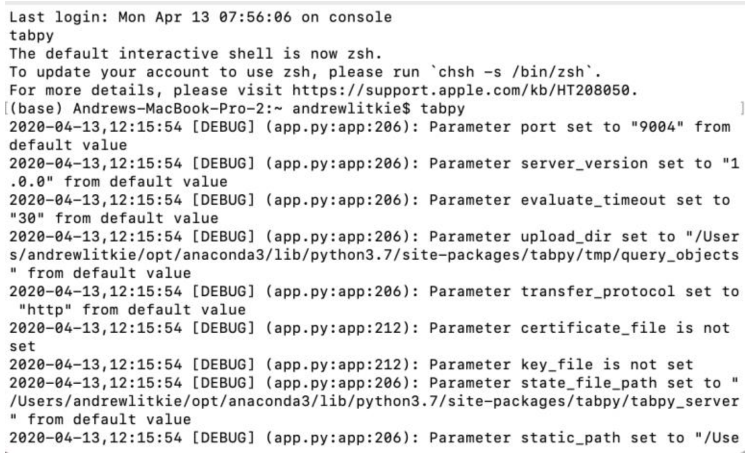
How the integration was so smooth

1 To integrate the recommender system into TripActions front end, we decided to leverage the power of TabPy and Tableau.

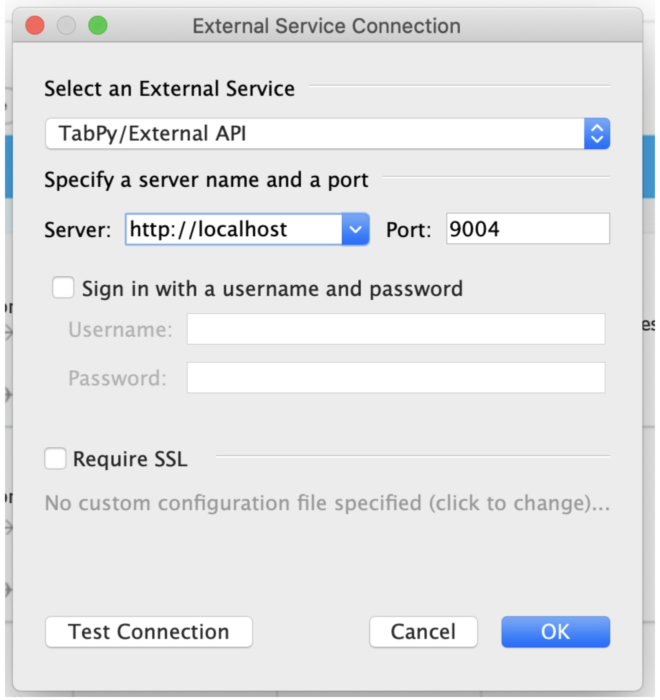
2 The following needed to be installed:
- TabPy (pip install tabpy)
- Virtual Environment (pip install virtualenv)

3 Create a virtual environment for TabPy to be able to connect to Tableau:
- virtualenv my-tabpy-env

4 To establish the Taby and Tableau connection you need to open up the TabPy and get the port number, as seen below:



5 Then on Tableau you need to manage the connection show that Tableau is listening to the port that was just opened up:



6 Then you need to open up the recommender system on Jupyter notebook through the virtual environment that was created:

Model Implementation

How the integration was so smooth

7

The recommender system needs to be deployed into the TabPy environment. This is done on the right, which uses the recommender system to create an endpoint in the localhost called FlightPredictor.

```python
# Connect to TabPy server using the client library
connection = tabpy_client.Client('http://localhost:9004/')


# Create reccomender system:

def recommend(arrival_time, origin, destination, airline, threshold, rec_data = data, model = model):


# Publish the function to TabPy server so it can be used from Tableau
# Using the name Flight Predictor it can be called on Tableau
connection.deploy('FlightPredictor',
                  recommend,
                  'Returns probability of delay of flights in the US', override = True)
```

8

By searching the following URL: http://localhost:9004/endpoints, you can observe all the deployed functions. Where you can see that the the FlightPredictor function has been successfully deployed.

localhost:9004/endpoints

{"GermanCreditCheck": {"description": "Classifies bad or good according to the model trained by relevant dataset", "type": "model", "version": 6, "dependencies": [], "target": null, "creation_time": 1585140925, "last_modified_time": 1586270551, "schema": null, "docstring": "-- no docstring found in query function --"}, "DiagnosticsDemo": {"description": "Returns diagnosis suggestion based on ensemble model trained using Wisconsin Breast Cancer dataset", "type": "model", "version": 12, "dependencies": [], "target": null, "creation_time": 1585143498, "last_modified_time": 1586078702, "schema": null, "docstring": "-- no docstring found in query function --"}, "DiagnosticsDemo1": {"description": "Returns diagnosis suggestion based on ensemble model trained using Wisconsin Breast Cancer dataset", "type": "model", "version": 1, "dependencies": [], "target": null, "creation_time": 1585315750, "last_modified_time": 1585315750, "schema": null, "docstring": "-- no docstring found in query function --"}, "FlightPredictor": {"description": "Returns probability of delay of flights in the US", "type": "model", "version": 13, "dependencies": [], "target": null, "creation_time": 1585841412, "last_modified_time": 1586270353, "schema": null, "docstring": "-- no docstring found in query function

9

The last step is to call the function in Tableau. To do this you need to create a calculated field to query the endpoint that was just created.

Flight Predictor

Results are computed along Table (across).

```
SCRIPT_STR("return tabpy.query('FlightPredictor',_arg1,_arg2,_arg3,_arg4,_arg5,_arg6)['response']",
[date], [Time] ,[origin], [destination], [airline], [threshold])
```

Default Table Calculation

The calculation is valid.

Apply OK

How do we deliver?

I have a dinner with a client at 9pm in Denver, but I can only check in at my hotel from 6:30pm

Origin: DFW
Destination: DEN

	Date	Departure Time	Arrival Time	Airline	Probability of Delay
6957952	28/12/2020	18:56	20:12	AA	31.07%
6906869	28/12/2020	17:25	18:32	AA	35.93%
6996924	28/12/2020	17:52	18:58	AA	35.93%
6996931	28/12/2020	17:21	18:33	AA	35.93%
7255138	28/12/2020	17:55	19:01	NK	53.41%
7251505	28/12/2020	18:15	19:25	NK	60.63%
7255143	28/12/2020	18:00	19:07	NK	60.63%

… also, most of my miles are with American Airlines!!!

Origin: DFW
Destination: DEN
Airline: AA

	Date	Departure Time	Arrival Time	Probability of Delay
6957952	28/12/2020	18:56	20:12	31.07%
6906869	28/12/2020	17:25	18:32	35.93%
6996924	28/12/2020	17:52	18:58	35.93%
6996931	28/12/2020	17:21	18:33	35.93%