Pacific Airfields Data Analysis

Developing Airfield Functional Categories

Using K-Means Clustering

Segment 4 Deliverable

28 Sep 2022

Christopher Rice

Topic Selection

I consult with aerospace companies, investors, and other interested parties who need to understand business opportunities for new types of aircraft in Pacific Ocean regions.

This project semi-automates the ETL process for airfield data, then employs K-means clustering to classify airfields based on (1) the total length of their runway(s), (2) the surface material of their runway(s), (3) the number of radio frequencies and navigation aids, and (4) the number of airlines providing scheduled service.

Classification of airfields creates a useful reference dataset that can be browsed in a public Tableau dashboard. Automating the ETL process saves large amounts of work for future airfield analysis for other regions. The clustering analysis provides a classification schema for detailed transportation studies to inform aircraft development, aircraft basing, aircraft routing, upgrading existing airfields, and building new facilities.

Technologies, languages, tools, and algorithms:

This project experimented with a range of machine learning models including a neural network, supervised multinomial logistic regression, and the K-means clustering algorithms. Models were executed using the Python sklearns library; coding and model runs were performed in Jupyter Notebook. Model outputs were examined in 3D graphs generated with plotly.

Preliminary data exploration and model runs were conducted using Excel CSV files. Web scraping and transformation of downloaded data tables was performed with pandas in Jupyter Notebook. An entity relationship diagram was prepared using the QuickDBD website tools; final preparation of the PostgresSQL database was performed in pgAdmin 4. SQL files were examined and edited in VSCode.

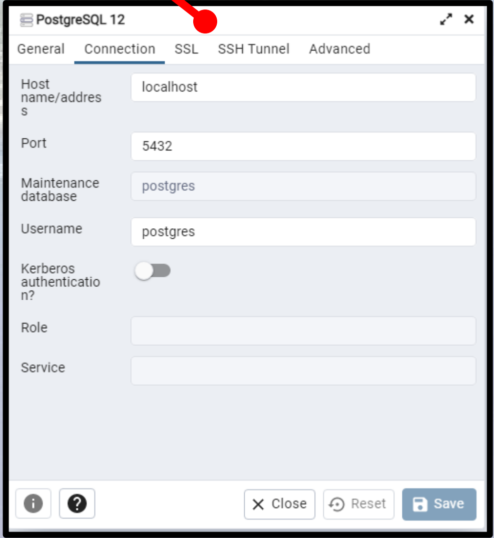
Data visualizations were prepared in Tableau.

Airfields.sql

```sql
CREATE TABLE airfields (
    GPS_Code VARCHAR(4) NOT NULL,
    af_name VARCHAR,
    latitude FLOAT(24),
    longitude FLOAT(24),
    runway_length SMALLINT,
    freq_count SMALLINT,
    airline_count SMALLINT,
    PRIMARY KEY (GPS_Code),
    UNIQUE (GPS_Code)
);
```

```python
# Show AF1
AF1
```

	ID	name	continent	country	region	af_type	latitude	longitude	elevation	runway_L	runway_W	freq_count	airline_count
0	YBAU	Badu Island Airport	Oceania (OC)	Australia (AU)	Queensland (AU-QLD)	small airport	-10.150000	142.173401	14	2788.0	0.0	4	0
1	YHID	Horn Island Airport	Oceania (OC)	Australia (AU)	Queensland (AU-QLD)	medium airport	-10.586400	142.289993	43	8609.0	173.0	3	0
2	YKUB	Kubin Airport	Oceania (OC)	Australia (AU)	Queensland (AU-QLD)	small airport	-10.225000	142.218002	15	3281.0	60.0	2	0
3	YSNF	Norfolk Island International Airport	Oceania (OC)	Norfolk Island (NF)	(unassigned) (NF-U-A)	medium airport	-29.041599	167.938995	371	11106.0	248.0	4	4
4	NFCI	Cicia Airport	Oceania (OC)	Fiji (FJ)	Eastern (FJ-E)	small airport	-17.743299	-179.341995	13	0.0	0.0	0	0



Raw Data -> ETL -> SQL Database

HTML table from Airportdatabase.net

	Unnamed: 0	Unnamed: 1
0	Honiara International Airport details and info...	Honiara International Airport details and info...
1	ident:	AGGH
2	type:	medium airport
3	NaN	NaN
4	latitude:	-9.42800045013428
5	longitude:	160.05499267578125
6	elevation:	28 ft.
7	continent:	Oceania (OC)
8	iso country:	Solomon Islands (SB)
9	ISO Region:	Capital Territory (Honiara) (SB-CT)
10	Municipality:	Honiara
11	Scheduled Service:	yes
12	GPS Code:	AGGH
13	IATA Code:	HIR
14	wikipedia link:	Honiara International Airport in Wikipedia
15	l Airport runways	Honiara International Airport runways
16	06/24	7218x148 ft.
17	Honiara International Airport frequencies	Honiara International Airport frequencies
18	AFIS (INFO)	118.1 Mhz
19	INFO (INFO)	342.5 Mhz
20	nal Airport navaids	Honiara International Airport navaids
21	Honiara (HN)	112600 Mhz
22	Honiara (HN)	348 Mhz
23	A ... ara International ...	Airlines flying from/to Honiara International ...
24	Pa ... n Blue (DJ) , Vir...	Pacific Blue (DJ) , Polynesian Blue (DJ) , Vir...

Airport data

Runway data

Signals data

Airline data

pandas ETL code

```python
# Create URL for HTML table
for index, row in url_surface.iterrows():
    url = row['url']

    # Read HTML table
    df1 = pd.read_html(url)[0]

    # find runway data rows
    pattern = '[0-9]x[0-9]'
    runway = df1['Col_2'].str.contains(pattern, na=False)

    # tag runway data rows in table w/"True"
    runway_B = pd.concat([df1, runway], axis=1)

    # Assign column name to runway data tag
    runway_B.columns = ['Col_1', 'Col_2', 'Col_3']

    # Select runway data rows
    runway_C = runway_B.loc[runway_B['Col_3'] == True]
    if runway_C.shape[0] == 0:
        runway_L = 0
        runway_W = 0
    else:
        runway_C['length'] = ''
        runway_C['width'] = ''
```

Wikipedia runway data

Maloelap Airport
Taroa Airfield

IATA: MAV · ICAO: *none* · FAA LID: 3N1

Summary

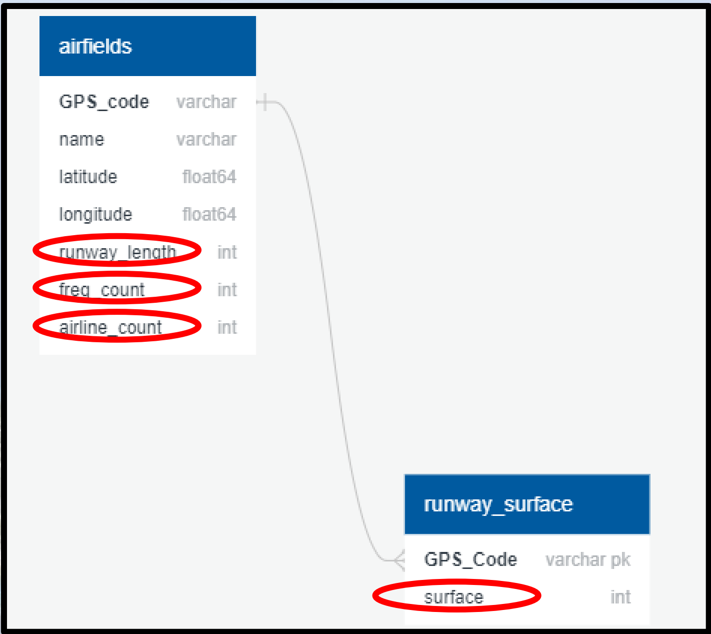
Elevation AMSL	4 ft / 1.2 m
Coordinates	8°42′18″N 171°13′50″E

Runways

Direction	Length		Surface
	ft	m	
04/22	3,500	1,067	turf

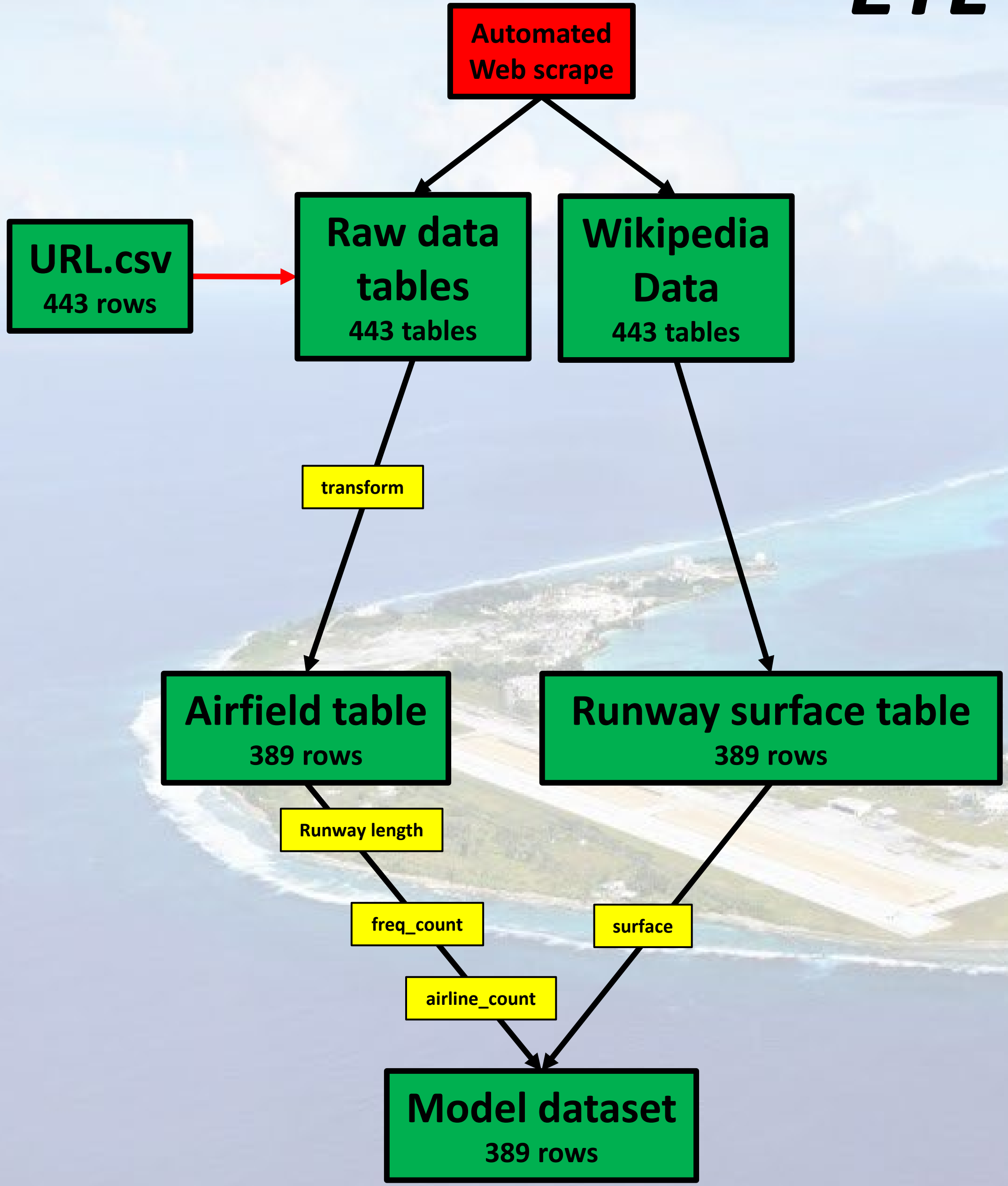
Source: Federal Aviation Administration[1]

ERD Diagram for SQL database



- Model features

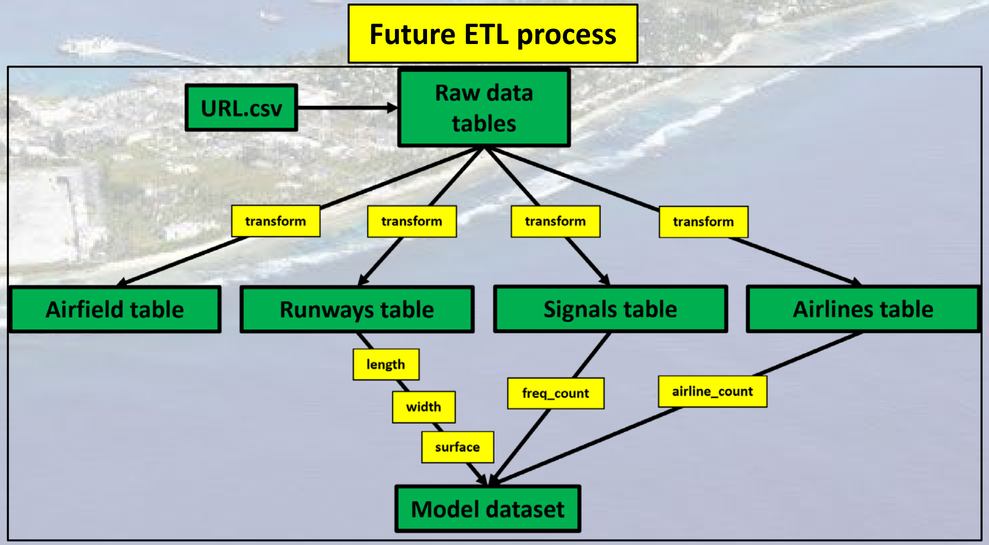
ETL Challenges



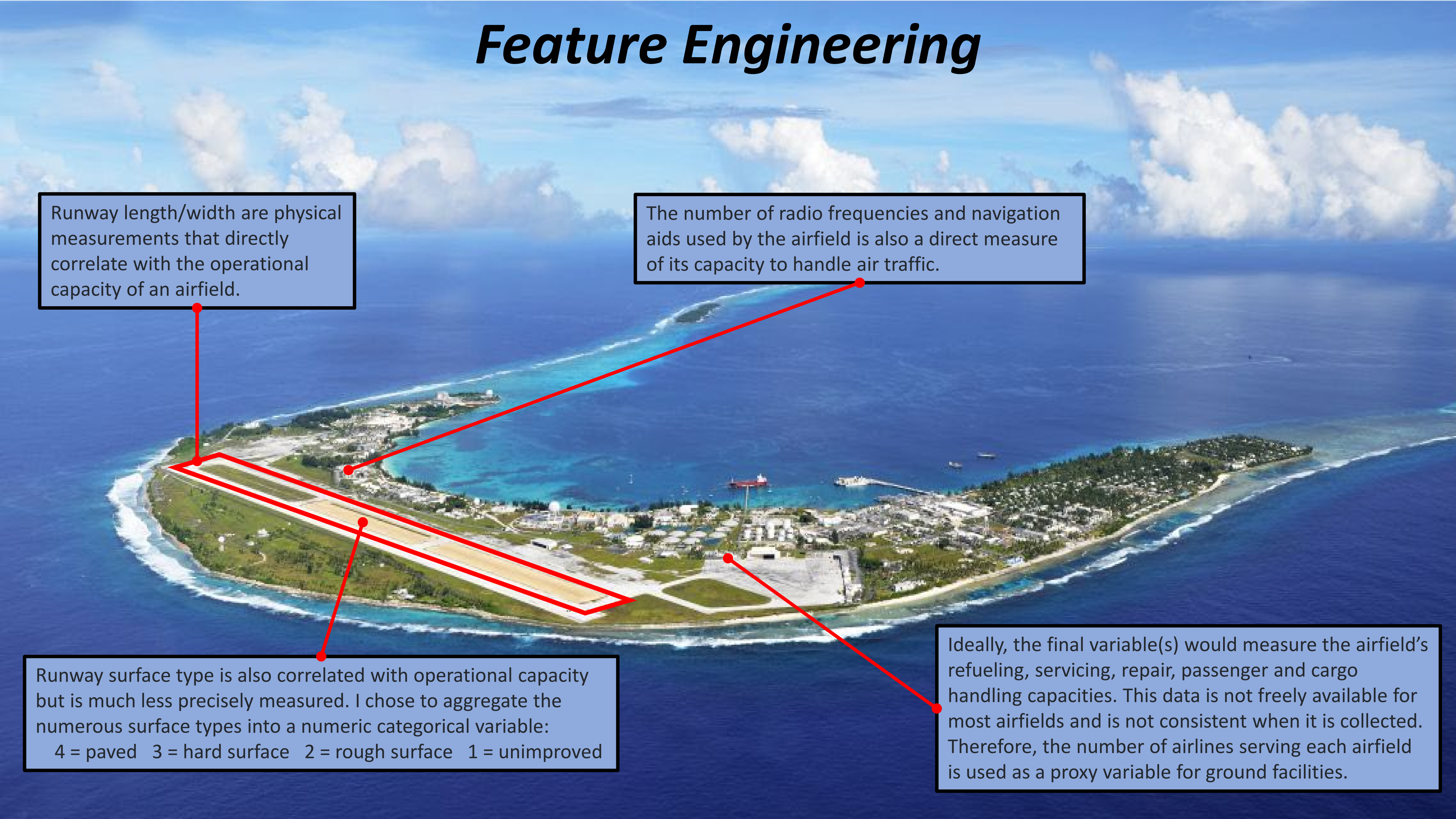
I created three model features (runway length, signals count, and airline count) during the transform process and stored them in the airfield table.

The runway surface feature was collected manually from Wikipedia and stored in a separate SQL table, then joined with the other model features to complete the input data for the model.

A future enhancement of this process would generate additional tables in the SQL table with reference data for runways, signals, and airlines.



Feature Engineering



Runway length/width are physical measurements that directly correlate with the operational capacity of an airfield.

The number of radio frequencies and navigation aids used by the airfield is also a direct measure of its capacity to handle air traffic.

Runway surface type is also correlated with operational capacity but is much less precisely measured. I chose to aggregate the numerous surface types into a numeric categorical variable:

4 = paved   3 = hard surface   2 = rough surface   1 = unimproved

Ideally, the final variable(s) would measure the airfield's refueling, servicing, repair, passenger and cargo handling capacities. This data is not freely available for most airfields and is not consistent when it is collected. Therefore, the number of airlines serving each airfield is used as a proxy variable for ground facilities.

Model Selection

Multinomial logistic regression results

```
#Use statsmodels to assess variables

logit_model=sm.MNLogit(y_train,sm.add_constant(X_train))
logit_model
result=logit_model.fit()
stats1=result.summary()
stats2=result.summary2()
print(stats1)
print(stats2)
```

```
Optimization terminated successfully.
         Current function value: nan
         Iterations 14
                          MNLogit Regression Results
==============================================================================
Dep. Variable:                  Class   No. Observations:                  131
Model:                        MNLogit   Df Residuals:                      104
Method:                           MLE   Df Model:                           24
Date:                Wed, 14 Sep 2022   Pseudo R-squ.:                     nan
Time:                        20:34:27   Log-Likelihood:                    nan
converged:                       True   LL-Null:                       -135.93
Covariance Type:            nonrobust   LLR p-value:                       nan
==================================================================================
  Class=Class_1       coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
const                  nan        nan        nan        nan         nan         nan
Runway_1               nan        nan        nan        nan         nan         nan
Runway_2               nan        nan        nan        nan         nan         nan
Type_Air taxi          nan        nan        nan        nan         nan         nan
```

```
#Create a confusion matrix
#y_test as first argument and the preds as second argument
confusion_matrix(y_test, preds)
```

```
array([[19,  1,  0,  0],
       [ 0,  6,  2,  0],
       [ 0,  0,  2,  0],
       [ 0,  0,  2,  1]], dtype=int64)
```

Multinomial logistic regression results

```
#transform confusion matrix into array
#the matrix is stored in a vaiable called confmtrx
confmtrx = np.array(confusion_matrix(y_test, preds))
#Create DataFrame from confmtrx array
#rows for test: Male, Female, Infant designation as index
#columns for preds: male, predicted_female, predicted_infant as column

pd.DataFrame(confmtrx, index=['Class_0','Class_1','Class_2','Class_3'],
columns=['predicted_Class_0','predicted_Class_1','predicted_Class_2','predicted_Class_3'])
```

	predicted_Class_0	predicted_Class_1	predicted_Class_2	predicted_Class_3
Class_0	19	1	0	0
Class_1	0	6	2	0
Class_2	0	0	2	0
Class_3	0	0	2	1

```
#Accuracy statistics

print('Accuracy Score:', metrics.accuracy_score(y_test, preds))
```

```
Accuracy Score: 0.8484848484848485
```

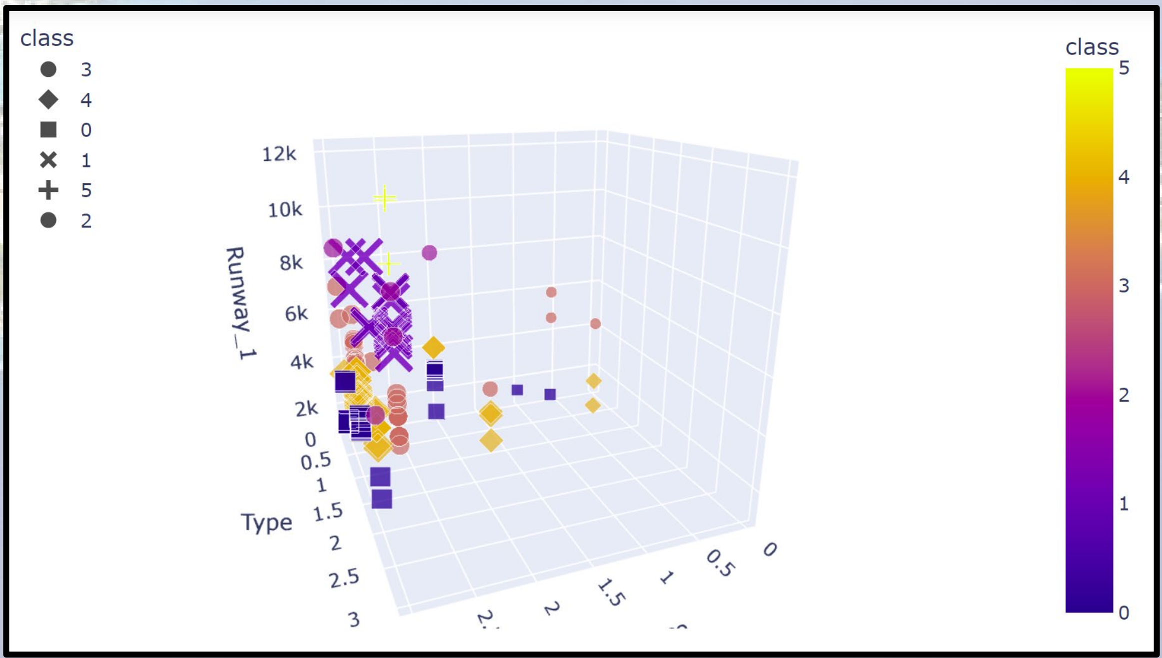
Initial modeling experiments explored using the provisional dataset as training data for a neural network, then for a multinomial logistic regression. After further analysis -- particularly the implications of finding additional data fields for radio stations, navigation aids, and airline service in airplanedatabase.net -- both approaches were discarded. The best use of this data was to group airfields into functional categories with an unsupervised learning model.

K-Means Model Selection

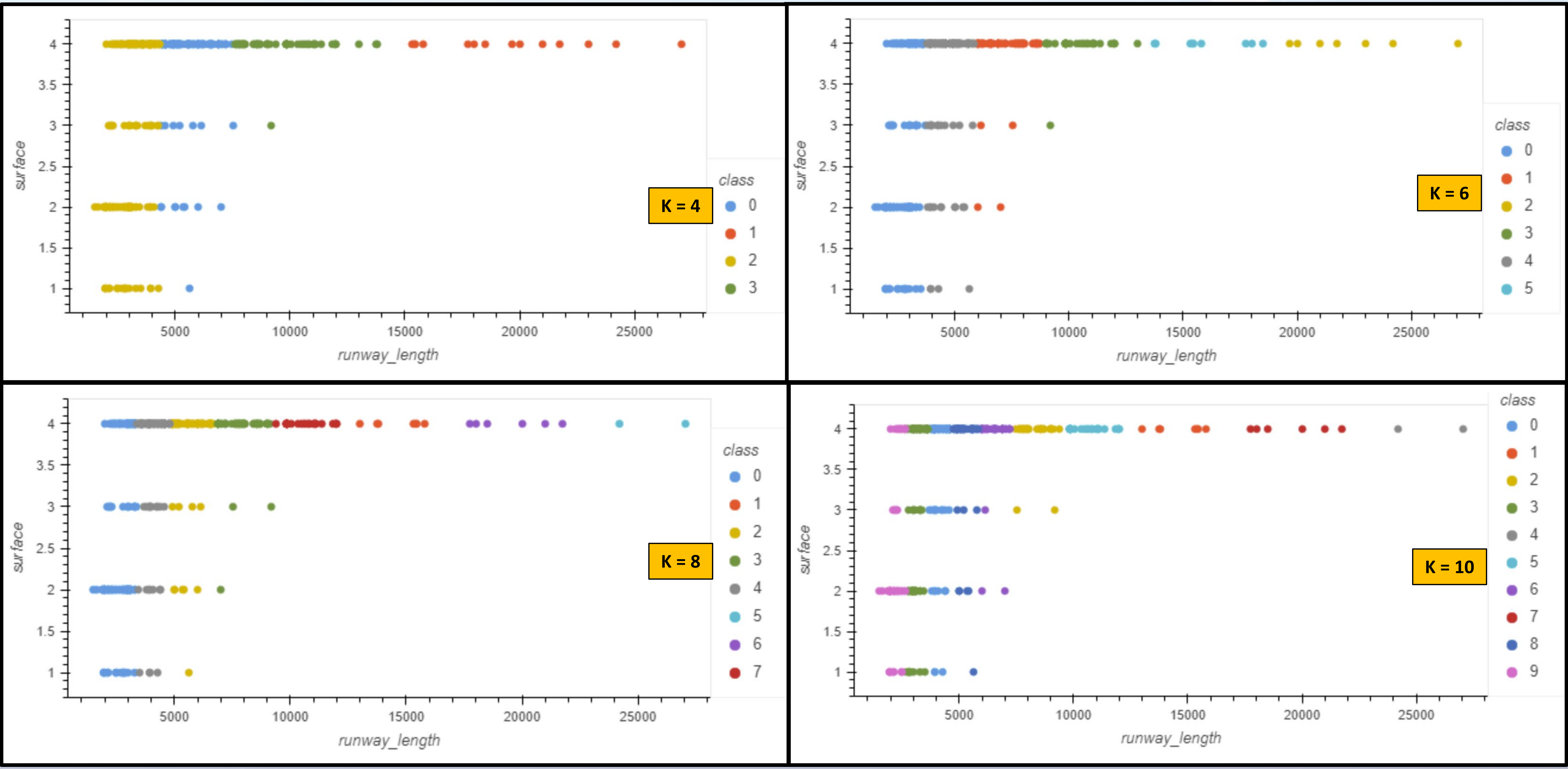
- K-means clustering creates more finely-grained categories of airfields that can be mapped and analyzed as part of market research and mobility studies. The principal limitation is that these categories are not direct measurements of operational capacity and cannot be linked to any explanatory variables in a quantified fashion.

- The K-means algorithm does not generate predictions, so accuracy scores were not useful for this model. What K-means does accomplish is to group observations based on their proximity to centroids in the n-dimensional model space (where n equals the number of model features; n=4 in this analysis).

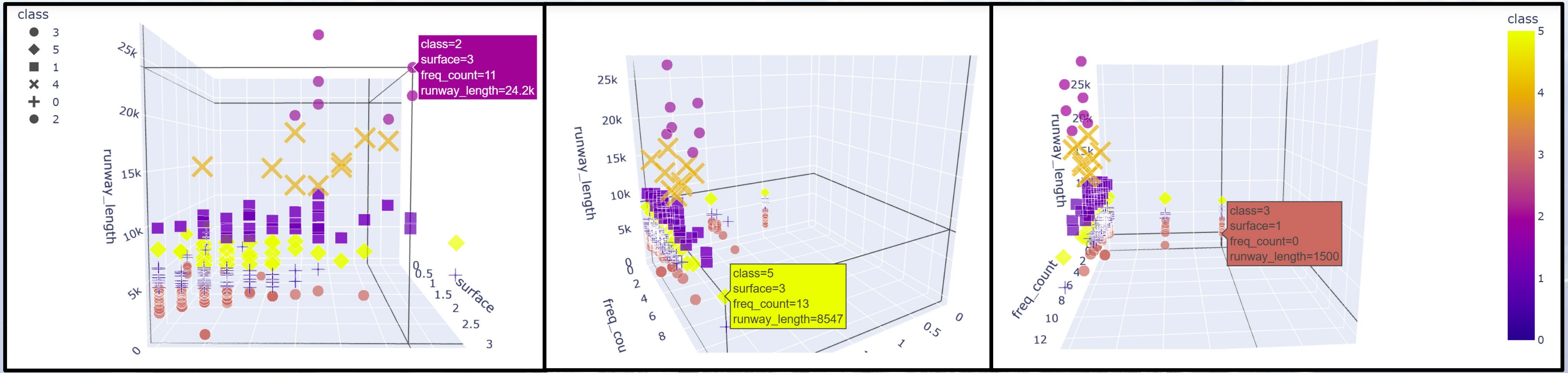
- The value of the model comes from visualization of the model output. In this analysis the output is placing airfields into one of six classes, based on how they cluster in the 4-dimensional space defined by total runway length, runway surface, # of EM signals, and # of airlines providing scheduled service.



K-Means Results – Stratification based on Runway Length

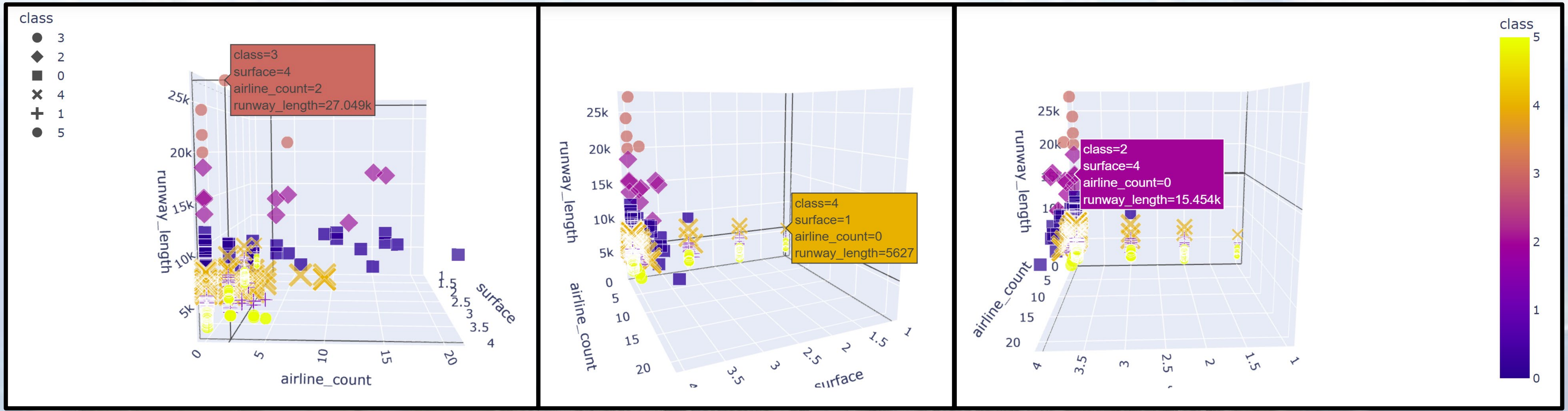


v3 Model Results



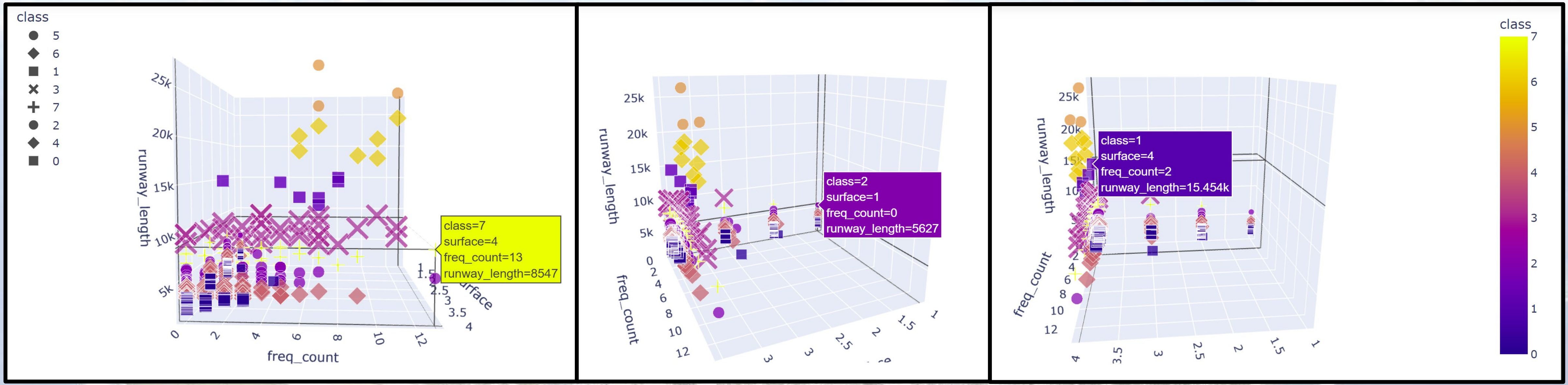
The base model results show that airfields are clustered in six distinct bands largely defined by runway length.

v4 Model Results – Adding More Small Airfields



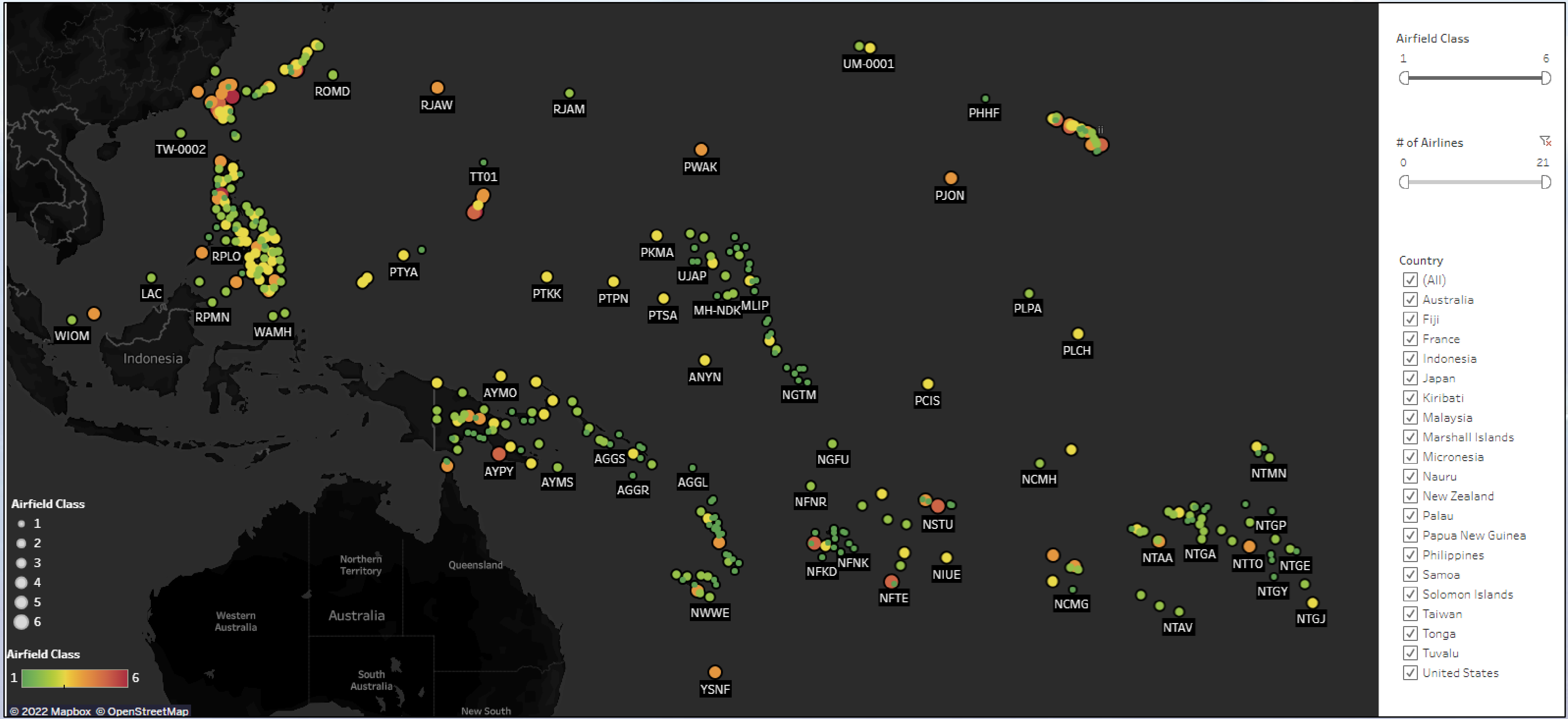
Fixing the ETL process allowed rapid addition of 111 additional airfields, most of them with small runways and unimproved surfaces. These additions did not change the clustering of airfields based on runway length but did show that this stratification extends across runway surface types.

v4 Model Results – K=8



Additional model runs were conducted with K=4, K=8, and K=10. The results for the K=8 run are shown here. The takeaway is that clustering is still based on runway length.

Tableau Dashboard - Map

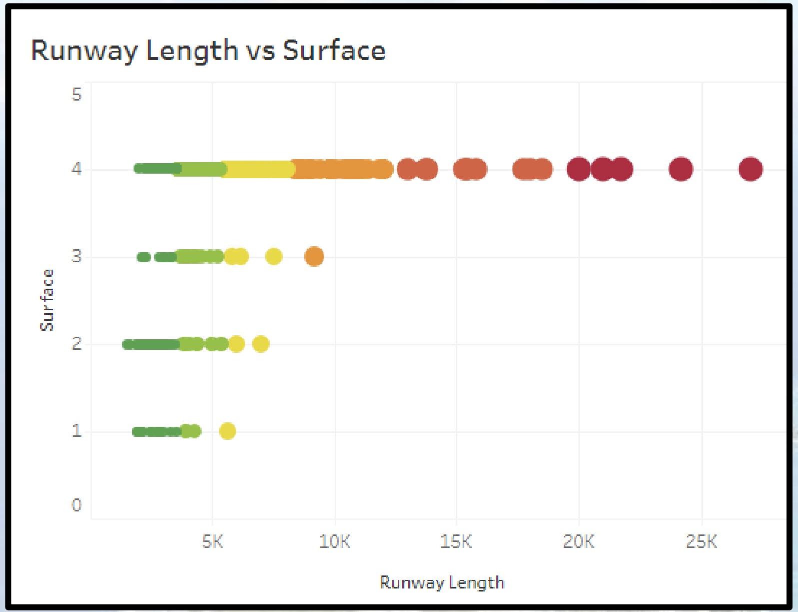


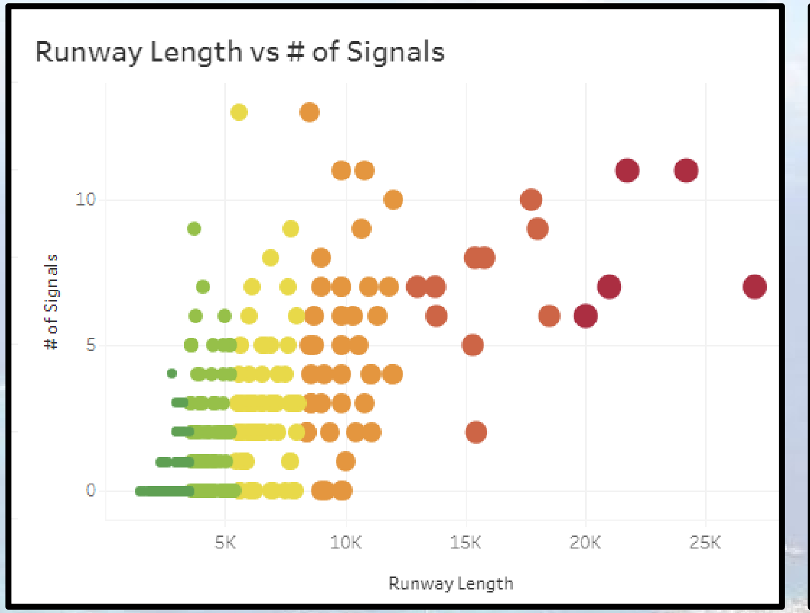
The first section of the dashboard is a geographic view that displays airfields by location. This view displays airfields as circles with their size and color determined by the output of the machine learning model.

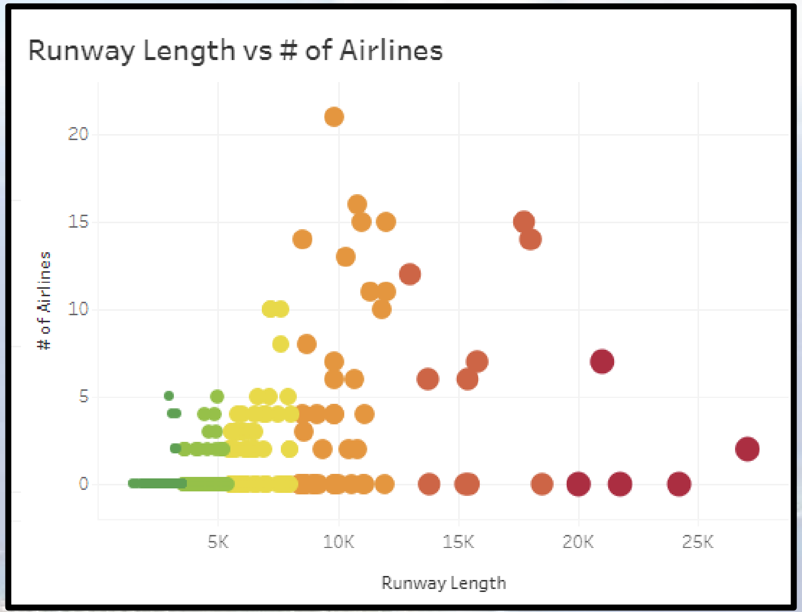
The interactive elements for the map include:

1) Slider bars to filter airfields based on airfield class and # of airlines.
2) A filter list allowing selection of one or more countries' airfields.

Tableau Dashboard - Analytics







The analytic view is comprised of 3 scatter plots showing the relationship between runway length and three other model variables:

(1) runway surface type.
(2) # of signals
(3) # of airlines.

The interactive elements for the graphs are linked to the map display and are filtered based on airfield class and countries. Airport class is displayed in the same manner as the map (size/color of the circles on the scatter plot).

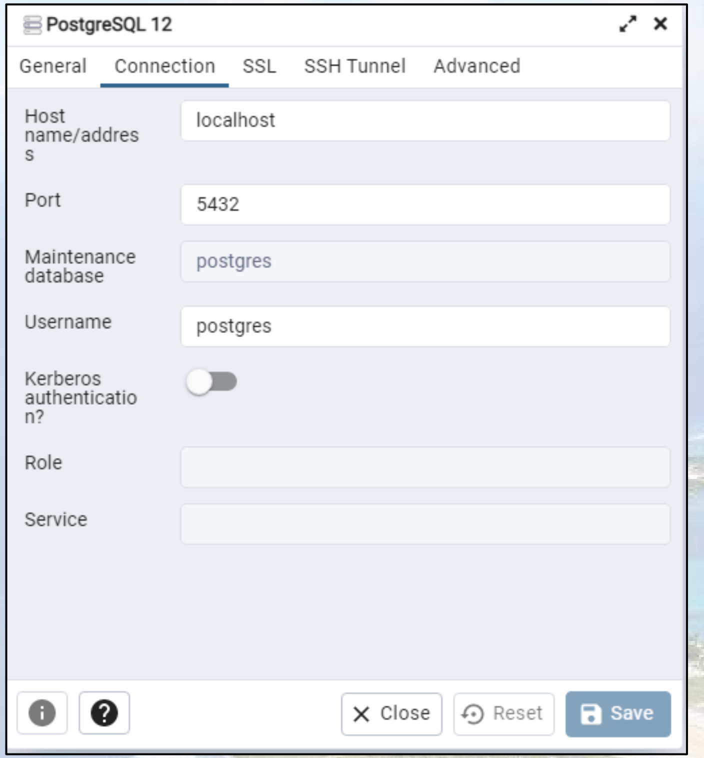
Lessons Learned

- Spend more time looking for data sources and evaluating them. Try to anticipate the challenges of the ETL process (based on experience).
- Build web scraping skills (but have a fallback plan if it doesn't work).
- More SQL practice. With this small of a dataset, I could always fall back on CSVs to handle data. That would not work with big data.
- Understand what each type of machine learning model does for you. I wasted time thinking about models that weren't appropriate for my problem; clustering was always the obvious solution.
- There is a critical phase of post-processing model outputs into visualization parameters; there are many nuances in Tableau that apply to this.
- Future analysis would benefit from more model features; facilities and air traffic data would go a long way in providing finer-grained results.

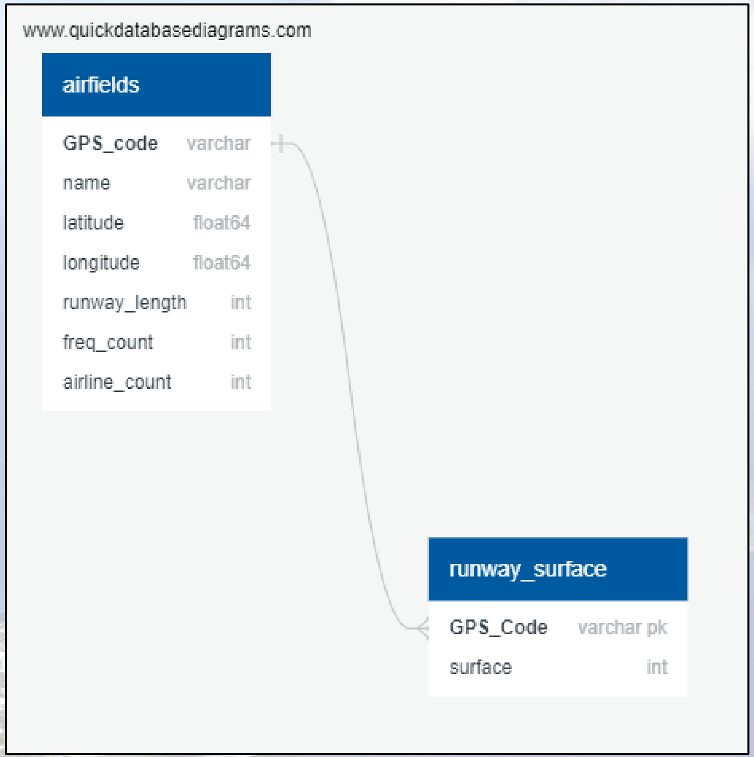
Backups

SQL Database

SQL connection information



ERD



The "Airfields" SQL database contains two tables:

"airfields" contains the data scraped from airportdatabase.net

"runway_surface" contains the data collected manually from Wikipedia and Google Earth.

Required database functionality is documented in the *.sql files and *.ipynb model files.