INTEGRATED ORGANIZATIONAL MACHINE LEARNING FOR AVIATION FLIGHT DATA

MICHAEL J. PRITCHARD, PAULTHOMAS, ERICWEBB, JON MARTIN, & AUSTIN WALDEN NATIONAL TRAINING AIRCRAFT SYMPOSIUM OCTOBER, 2022 KANSAS STATE Salina Aerospace

UNIVERSITY

Campus

SITUATION

Major challenges face many flight organizations:

- I.Integration and automation of data collection frameworks
- 2. Data feature engineering, cleanup and preparation
- 3. Operationalizing embedded machine learning frameworks

CHALLENGES

While integration and automation of data collection efforts within many organizations is quite mature...

...there are special challenges for flight-based organizations (i.e., the automatic and efficient transmission of aircraft flight data to centralized analytical data processing systems).

OPPORTUNITY

- Constraints for implementing classical machine learning methods (i.e., clustering, classification, or prediction)
- This magnifies design challenges for novel 'prescriptive-based' architectures

Our research is focused on a design pattern for:

- a) The integration and automation of data collection for...
- b) ... an organizationally embedded ensemble machine learning method

APPLIED RESEARCH QUESTIONS

I.Identify challenges associated with the integration and automation of fleet data collection frameworks

2. Determine feature engineering, cleanup and preparation processes

3. Operationalizing embedded machine learning frameworks

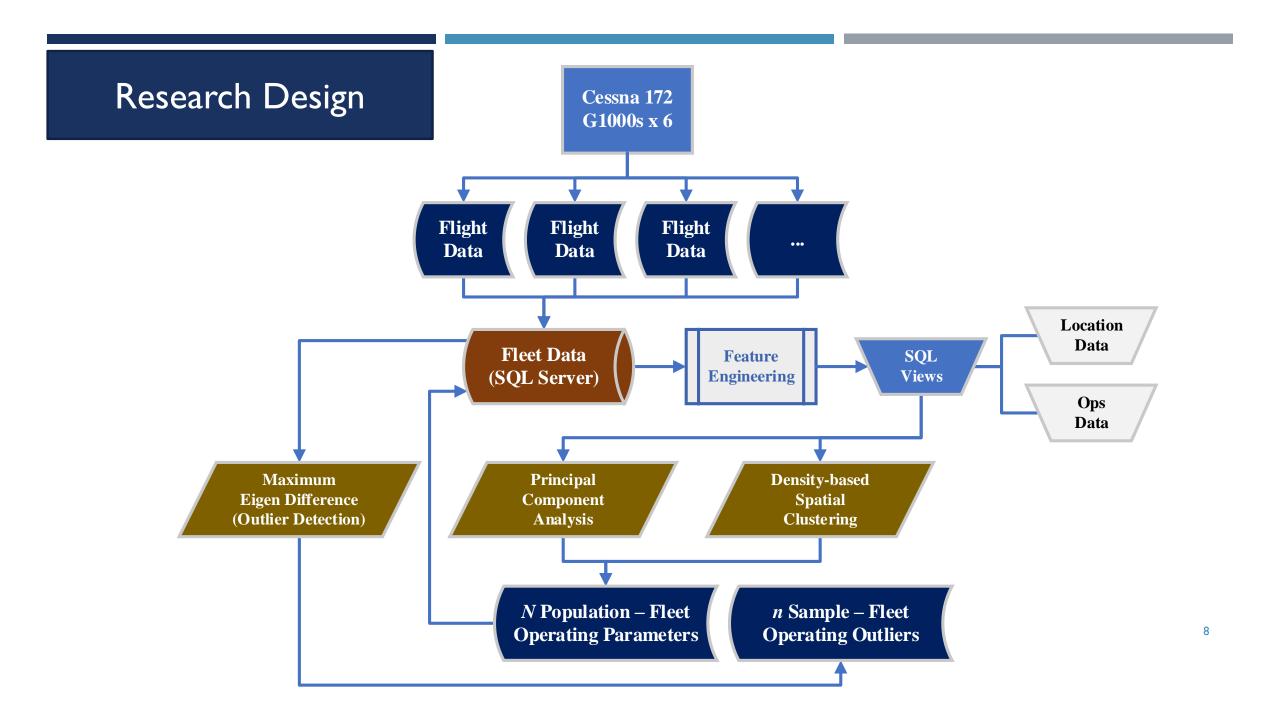
BACKGROUND RESEARCH, PART I

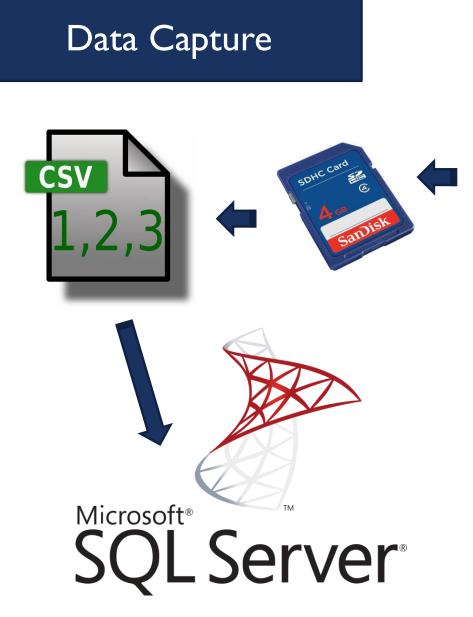
- Airplane monitoring systems have been around for several decades...
 - (Taylor, 1969; Milligan, Zhou, and Wilkerson, 1995).
- ...data from sensors for location, structure, engine, and cabin environment...
 - Gao et al., 2018).
- Image: monitoring systems are wired and wireless; and are used to enhance and predict maintenance...
 - (Zelenika et al., 2020).

BACKGROUND RESEARCH, PART II

- Prevalence of monitoring systems and the prompt analysis of data from collected from fleets can allow for more timely and effective maintenance activities which will reduce aircraft downtime while also reducing operational costs arising from maintenance (Dupuy, Wesely, and Jenkins, 2011).
- There has been a trend for applying statistical techniques to data collected from fleets of commercial aircraft to identify aircraft anomalies or abnormal trends (Gorinesky, Matthews, and Martin, 2012; Sumathi et al., 2017).











n = 65,525 (flight log entries)

Data	Framework
------	-----------

Flight Logs



SQLQuery9.sqlRS\mjp001 (64))*	<mark>3.sqlRS\mjp001 (68))* ≄ × </mark> SQLQuery6.sql
USE [FleetData]	
GO	
/****** Object: View [dbo].[v_flight	nt_log_v2] Script Date: 10/24/202
alter view [dbo].[v_flight_log_v2]	
as	
SELECT	
isnull logic on everything	
<pre>convert(int, isnull(year([Lcl_Date])</pre>	
<pre>convert(int, isnull(day([Lcl_Date]))</pre>	, , ,
<pre>convert(time, isnull([Lcl_Time], 0)</pre>) as lcl_time,
<pre>convert(float, isnull([Latitude], 0)</pre>	
<pre>convert(float, isnull([Longitude], 6</pre>	, , ,
<pre>convert(float, isnull([AltB], 0)</pre>) as altb,
<pre>convert(float, isnull([BaroA], 0)</pre>) <mark>as</mark> baro_a,
<pre>convert(float, isnull([AltMSL], 0)</pre>) as alt_msl,
<pre>convert(float, isnull([OAT], 0)</pre>) as oat,
<pre>convert(float, isnull([IAS], 0)</pre>) as ias,
<pre>convert(float, isnull([GndSpd], 0)</pre>) <mark>as</mark> gndspd,
<pre>convert(float, isnull([TAS], 0)</pre>) as taspd,
<pre>convert(float, isnull([VSpd], 0)</pre>) as vspd,
<pre>convert(float, isnull([WndSpd], 0)</pre>) as wndspd,
<pre>convert(float, isnull([Pitch], 0)</pre>) as pitch,
<pre>convert(float, isnull([Rol1], 0)</pre>) as roll,
<pre>convert(float, isnull([HDG], 0)</pre>) as hdg,
<pre>convert(float, isnull([volt1], 0)</pre>) as volt1,
<pre>convert(float, isnull([amp1], 0)</pre>) as amp1,
<pre>convert(float, isnull([E1_0ilT], 0)</pre>) as e1_oil_t,
<pre>convert(float, isnull([E1_RPM], 0)</pre>) as e1_rpm,
<pre>convert(float, isnull([HCDI], 0)</pre>) as hcdi,



■ dbo.v_flight_log □ dbo.v_flight_log_v2 🗏 🖉 Columns [■] target (int, null) [□] lcl_date_day (int, null) Icl_time (time(7), null) Iat (float, null) ^I long (float, null) [■] altb (float, null) baro_a (float, null) [■] alt_msl (float, null) I oat (float, null) I ias (float, null) [■] gndspd (float, null) [■] taspd (float, null) vspd (float, null) [■] wndspd (float, null) [■] pitch (float, null) [■] roll (float, null) ^I hdg (float, null) [■] volt1 (float, null) amp1 (float, null) [■] e1_oil_t (float, null) ^I e1_rpm (float, null) 🛚 hcdi (float, null) 🗉 vcdi (float, null) [■] mag_var (float, null) hal (float, null) 🗉 🖷 Triggers

Centralized Data View

SQLQuery6.sql - SLN-MJP001-LT\SQLEXPRES		Data (L	JSERS\mjp0	001 (79))* - Mic	rosoft SQ	L Server M	anager	ment St	tudio					Q	uick L	aunc	h (Ctr	+Q)	ş	- <	đ×
File Edit View Project Tools Window	00050500				1	1															
🖉 🔹 💿 🔤 👻 🔄 🖕 🏜 🖉 💭 New Query		DHT INLA		1 7 · C	- 8	- 2				- 5	۵ × ۵		÷								
I I I FleetData	= 1	20日			3 2		_														
		17.2 V. T.		The second s				41 - T- 14													
Object Explorer · · · · · · · · · · · · · · · · · ·	× sq	and the second second second second	and and a set of the particular the	njp001 (66))*	10.000	LQuery6.s	qlRS∖	mjp00'	1 (79))*	÷×											
Connect • ₩ ₩ = ▼ C →		se]	.ect * fr	om [v_fligh	t_log_v	2]															+
																					<u></u>
🗉 🎟 dbo.log_170130_155300_KMKC	121	% •																			Þ
	I R	esults e	Messages																		
⊞		target	lcl_date_day	Icl_time	lat	long	altb	baro_a	alt_msl	oat ias	gndspd	taspd	vspd	wndspd	pitch	roll	hdg	volt1	amp1	e1_oil_t	e1_rpm
⊞ m dbo.log 180919 002527 KJEF	1	2017	30	13:01:18.0000000	38.7970428	-97.6401596	1340.2	30.07	1246.6	19.8 0	0	0	-28.1	0	0	0	0	27.6	2.8	53.95	1082.7
⊞	2	2017	30	13:01:19.0000000	38.7970428	-97.6401596		30.07	1247.8	19.8 0	0	0		0	0	-	0		2.8	53.99	1087.5
STATES STATES STATES AND A STAT	3	2017	30	13:01:20.0000000	38.7970428	-97.6401596	1340.2		1248.5	19.8 0	0	0		0	0	-	0		2.8	54.03	1087.1
⊞ I dbo.log_191025_185233_KSLN	4	2017	30 30	13:01:21.0000000 13:01:22.0000000	38.7970428 38.7970428	-97.6401596 -97.6401596			1249.1 1249.8	19.5 0 19.5 0	0	0		0	0	-	0	27.6	2.7	54.07 54.11	1085
	6	2017	30	13:01:23.0000000	38.7970467	-97.640152	1340.2		1249.8	19.5 0	0	0		0	0	-	0	27.6		54.14	1089.2
■■ Views	7	2017	30	13:01:24.0000000	38,7970467	-97.640152	1342.2		1251.2	19.5 0	0	0		0	0	0	0	27.6		54.19	1085.1
	8	2017	30	13:01:25.0000000	38.7970467	-97.640152	1342.2	30.07	1251.8	19.5 0	0	0	36.5	0	0	0	0	27.6	2.6	54.23	1084.3
⊞ Ø dbo.v flight log	9	2017	30	13:01:26.0000000	38.7970467	-97.640152	1342.2	30.07	1252.4	19.5 0	0	0	19.8	0	0	0	0	27.6	2.6	54.27	1085.2
e= dbo.v_light_log v2	10	2017	30	13:01:27.0000000	38.7970467	-97.640152	1340.2		1252.9	19.5 0	0	0	-16.99		0	•	0	27.6		54.31	1088
	11	2017	30	13:01:28.0000000	38.7970467	-97.640152	1341.2		1253.3	19.5 0	0	0	-12.04		0	-	0	27.5		54.35	1092.3
■■ Columns	12	2017	30	13:01:29.0000000 13:01:30.0000000	38.7970467 38.7970467	-97.640152 -97.640152	1339.2 1341.2		1253.6 1253.6	19.5 0 19.5 0	0	0	-43.66	0	0	-	0	27.6	2.3	54.4 54.44	1095.8
I target (int, null)	14	2017	30	13:01:31.0000000	38.7970467	-97.640152	1341.2		1253.0	19.5 0	0	0		0	2.2	1. San Carton			2.3	54.5	1093.5
Icl_date_day (int, null)	15	2017	30	13:01:33.0000000	38.7970428	-97.640152	1341.2		1253.3	19.5 0	0	0		0	2.19		167.2			54.58	1088.8
Icl time (time(7), null)	16	2017	30	13:01:34.0000000	38.7970428	-97.640152	1340.2	30.07	1253	19.5 0	0	0	-2.09	0	2.2	-0.87	167.2	27.6	2.3	54.63	1083.6
I lat (float, null)	17	2017	30	13:01:35.0000000	38.7970428	-97.640152	1341.2	30.07	1252.9	19.5 0	0	0	-3.07	0	2.24	-0.83	167.2	27.6	2.2	54.68	1081.8
	18	2017	30		38.7970428	-97.640152	1340.2		1252.7	19.5 0		0	-17.95		2.24		167.2			54.73	1081.2
Iong (float, null)	19	2017	30	13:01:37.0000000	38.7970467	-97.640152	1342.2		1252.5	19.5 0	0	0		0	2.23				2.3	54.78	1086.6
🛙 altb (float, null)	20	2017 2017	30 30	13:01:38.0000000 13:01:39.0000000	38.7970467 38.7970467	-97.640152 -97.640152	1343.2 1344.2		1252.3 1252.2	19.5 0 19.5 0	0	0		0	2.18		167.2 167.2		2.2	54.84 54.89	1089.8 1091.6
🗉 baro_a (float, null)	21	2017	30	13:01:40.0000000	38.7970467	-97.640152	1344.2		1252.2	19.5 0	0	0		0	2.23		167.2		2.2	54.89	1091.6
alt msl (float, null)	23	2017	30	13:01:41.0000000	38,7970467	-97.640152	1342.2		1252.1	19.5 0	0	0		0	2.24				2.1	55	1096.6
I oat (float, null)	24	2017	30	13:01:42.0000000	38.7970467	-97.640152	1343.2	30.07	1252.2	19.5 0	0	0	-5.71	0	2.23	-0.81	167.2	27.6	2.2	55.05	1095.6
	25	2017	30	13:01:43.0000000	38.7970505	-97.6401596	1343.2	30.07	1252.4	19.5 0	0	0	10.36	0	2.2	-0.86	167.2	27.6	2.2	55.11	1096.6
🗉 ias (float, null)	26	2017	30	13:01:44.0000000	38.7970505	-97.6401596		30.07	1252.5	19.2 0	0	0		0	2.2				2.2	55.17	1098.9
gndspd (float, null)	27	2017	30	13:01:45.0000000	38.7970505	-97.6401596			1252.8	19.2 0	0	0	-16.07		2.23		167.2			55.23	1096.4
🗉 taspd (float, null)	28	2017	30	13:01:46.0000000 13:01:47.0000000	38.7970505 38.7970505	-97.6401596 -97.6401596		30.07	1253.1 1253.3	19.2 0 19.2 0	0	0		0	2.25			27.6	2.1	55.29 55.35	1100.9 1102.5
^{II} vspd (float, null)	30	2017	30	13:01:48.0000000	38.7970505	-97.6401596			1253.3	19.2 0		0	-26.96		2.23		167.2			55.41	1102.5
B wndspd (float, null)	31	2017	30	13:01:49.0000000	38.7970505	-97.6401596		30.07	1253.1	19.2 0	0	0		0	2.25				2.2	55.47	1109
I pitch (float, null)	32	2017	30	13:01:50.0000000	38.7970505	-97.6401596	1341.2	30.07	1252.8	19.2 0	0	0	-5.38	0	2.24	-0.78	167.2	27.6	2	55.53	1109.2
	33	2017	30	13:01:51.0000000	38.7970505	-97.6401596		30.07	1252.7	19.2 0	0	0		0	2.2				2	55.6	1114.8
I roll (float, null)	34	2017	30	13:01:52 0000000	38 7970467	-97 6401596	1340.2	30.07	1252 6	192 0	0	0	-30 64	0	2 24	-0.86	167 2	27.6	1.8	55 66	1120.4
hdg (float, null)	Y																				

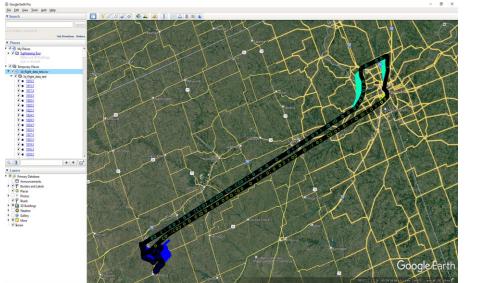
<u>**G1000 Data Definitions:**</u> Fala, N. (2019). *Data-driven safety feedback as part of debrief for General Aviation pilots* (Doctoral dissertation, Purdue University Graduate School).

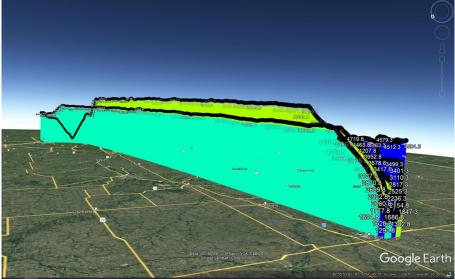
Analytical Framework

🕅 Spyder (Python 3.9) import pyodbc import numpy as np, pandas as pd **E** - 🔺 10 from sklearn.cluster import DBSCAN as dbs \Users\mjp001\OneDrive - Kansas State University\Documents\Research\NTAS 2022\pca_entropy_applied.py fleet_data_odbc_dbscan_v2.py × sql_odbc_python_corrplot.py × pca_entropy_applied.py × # data integration framework Kansas State University, Aerospace & Technology Campus Author: Michael J. Pritchard, PhD, <u>mjp00@ksu.edu</u> data = pd.read_sql(query, sql_conn) print(data.head()) print("Dataset shape:", data.shape) #PCA is good for taking data sets with higher dimensions print(data.isnull().any().any()) #and combining it into 2 dimensional scatter plots import pyodbc x = data.loc[:, ['amp1', # X-Axis import pandas as pd, numpy as np import matplotlib.pyplot as plt from sklearn.decomposition import FactorAnalysis, PCA from sklearn.preprocessing import StandardScaler sql conn = pyodbc.connect('DRIVER={SOL Server}; \ SERVER=SLN-MJP001-LT\SQLEXPRESS; \ DATABASE=FleetData: Trusted Connection=yes') # sql query plt.show() # showing the plot query = "select * from v_flight_log_v2" df = pd.read_sql(query, sql_conn) # cluster the data into five clusters print(df) # show me what you got! df.describe() print('>>>>>>------<<<<<<<') print(df.head()) features = ['altb', 'baro_a', 'alt_msl', 'oat', 'ias' # plot the clusters ,'vspd','pitch','roll','hdg','volt1' 45 plt.xlabel("X, Amps") # X-axis label ,'amp1','e1_oil_t','e1_rpm','mag_var','hal'] plt.show() # display plot X = df.loc[:, features].values y = df.loc[:,['target']].values X = StandardScaler().fit_transform(X) #Standardize the features #check for any missing value, this needs to return false n sample size print(np.any(np.isnan(X))) principal component 1 print('>>>>>>-----<<<<<<<') 39 principal component 2 65525 print(pd.DataFrame(data = X, columns = features).head()) target 65525 dtype: int64 $pca = PCA(n_components=2)$ principalComponents = pca.fit_transform(X) #this does a fit and transform in a single step principalDf = pd.DataFrame(data = principalComponents, columns = ['principal component 1', 'principal IPython console History

Spyder (Python 3.9) n X File Edit Search Source Run Debug Consoles Projects Tools View Help 🚺 🤌 🛃 C:\Users\mjp001\OneDrive - Kansas State University\Documents\Research\NTAS 2022 /mjp001\OneDrive - Kansas State University\Documents\Research\NTAS 2022\fleet_data_odbc_dbscan_v2.py a l € ■ ★ + + + + -> 229% = fleet_data_odbc_dbscan_v2.py × sql_odbc_python_corrplot.py × pca_entropy_applied.py × Author: Michael J. Pritchard, PhD, mjp00@ksu.edu 1500 1000 import matplotlib.pyplot as plt, seaborn as sns 500 from sklearn.neighbors import NearestNeighbors -500 sql conn = pyodbc.connect('DRIVER={SQL Server}; \ -1000SERVER=SLN-MJP001-LT\SQLEXPRESS; \ -1500Trusted_Connection=yes') -2000 query = "select * from [dbo].[v_flight_log_v2]' -10 10 15 + -15 -5 0 5 X. Amps Help Variable Explorer Plots Files 'vspd']].values # Y-Axis Console 1/A X # other nn search algos: auto, ball tree, kd tree, brute Users/mjp001/OneDrive - Kansas State University/Documents/Research/ neighb = NearestNeighbors(n_neighbors=4, algorithm='ball_tree') # create NearestNeighbors object class NTAS 2022') nbrs = neighb.fit(x) # fit data to the object C:\Users\mjp001\Anaconda3\lib\site-packages\pandas\io\sql.py:761: distances, indices = nbrs.kneighbors(x) # finding the nearest neighbours UserWarning: pandas only support SQLAlchemy connectable(engine/ connection) ordatabase string URI or sqlite3 DBAPI2 connectionother distances = np.sort(distances, axis = 0) # sorting the distances DBAPI2 objects are not tested, please consider using SQLAlchemy distances = distances[:, 1] # taking the second column of the sorted distances warnings.warn(plt.rcParams['figure.figsize'] = (5,3) # setting the figure size target lcl_date_day lcl_time ... vcdi mag_var hal plt.plot(distances) # plotting the distances 13:01:18.0000000 ... 0.0 13:01:19.0000000 ... 0.0 3.9 3704.0 0 2017 30 2017 30 3.9 3704.0 30 13:01:20.0000000 ... 0.0 2017 3.9 3704.0 2017 30 13:01:21.0000000 ... 0.0 3.9 3704.0 dbscan = dbs(eps = 8, min_samples = 4).fit(x) # fitting the model 2017 30 13:01:22.0000000 ... 0.0 3.9 3704.0 labels = dbscan.labels_ # getting the labels [5 rows x 25 columns] plt.scatter(x[:, 0], x[:,1], c = labels, cmap= "plasma", alpha = 0.1) # plotting the clusters Dataset shape: (65525, 25) False plt.ylabel("Y, V-Speed (fpm)") # Y-axis label IPython console History USP Python: ready @ conda: base (Python 3.9.12) Line 45. Col 22 UTF-8 CRLF RW Me

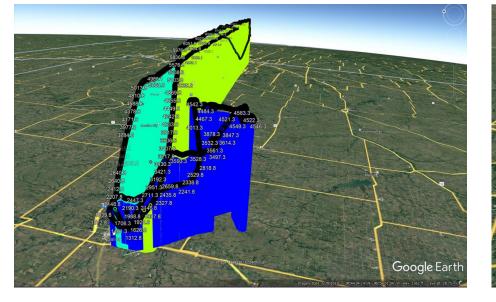
Python using Spyder, Integrated Development Environment (Scientific **PY**thon Development EnviRonment) Flight Data Analysis: *"Location Data"* Verification

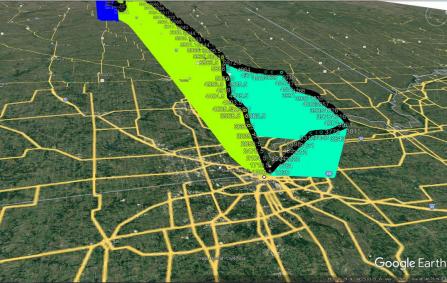




- Local Training (Blue)
- Solo Flight (Cyan/Lime)







Flight Data Analysis: "Ops Data" Verification

✓ AltB Positive Correlation:

- E1_Oil_T
- E1_RPM
- \circ IAS

Alt_MSL Positive Correlation:

- E1_Oil_T
- **E1_RPM**
- **IAS**
- (Note: Alt_MSL is more strongly correlated than AltB)

✓ IAS Positive Correlation:

- E1_Oil_T
- E1_RPM
- \circ **AltB**

rank_order																		
lat		1	0.92	0.17	-0.46	0.17	-0.024	0.26	-0.014	-0.11	-0.015	-0.24	-0.063		0.28	0.26	-0.39	-0.47
bug		0.92	1	0.29	-0.53	0.29	-0.19	0.34	0.032	-0.066	0.0027	-0.12	0.019		0.37	0.35	-0.58	
altb		0.17	0.29	1	-0.22	1	-0.88	0.81	0.0037	-0.26	-0.082	0.059	0.2		0.7	0.82		0.12
baro_a		-0.46	-0.53	-0.22	1	-0.26	0.2	-0.36	-0.0011	0.057	0.021	0.14	0.07	0.4	-0.56	-0.32	0.33	0.25
alt_msl		0.17	0.29	1	-0.26	1	-0.87	0.81	0.01	-0.25	-0.082	0.051	0.2	-0.37	0.72	0.83	-0.42	0.11
oat		-0.024	-0.19	-0.88	0.2	-0.87	1	-0.75	0.085	0.32	0.086	-0.3	-0.25	0.21	-0.64	-0.71	0.31	-0.2
as		0.26	0.34	0.81	-0.36	0.81	-0.75	1	-0.14	-0.39	-0.087	-0.038	0.25	-0.34	0.79	0.94	-0.26	-0.098
pdsv		-0.014	0.032	0.0037	-0.0011	0.01	0.085	-0.14	1	0.83	0.17	-0.091	-0.0079	0.051	-0.091	0.081	-0.2	0.057
pitch		-0.11	-0.066	-0.26	0.057	-0.25	0.32	-0.39	0.83	1	0.2	-0.071	-0.072	0.15	-0.29	-0.19	-0.096	0.02
Iloi		-0.015	0.0027	-0.082	0.021	-0.082	0.086	-0.087	0.17	0.2	1	0.044	0.0073	0.04	-0.087	-0.048	-0.023	-0.032
pdg		-0.24	-0.12	0.059	0.14	0.051	-0.3	-0.038	-0.091	-0.071	0.044	1	0.13	0.024	-0.18	-0.07	0.05	0.034
volt1		-0.063	0.019	0.2	0.07	0.2	-0.25	0.25	-0.0079	-0.072	0.0073	0.13	1	0.59	-0.0028	0.35	-0.081	0.14
amp1		-0.42	-0.39	-0.36	0.4	-0.37	0.21	-0.34	0.051	0.15	0.04	0.024	0.59	1	-0.39	-0.27	0.27	0.29
e1_oil_t		0.28	0.37	0.7	-0.56	0.72	-0.64	0.79	-0.091	-0.29	-0.087	-0.18	-0.0028	-0.39	1	0.69	-0.22	-0.13
e1_rpm		0.26	0.35	0.82	-0.32	0.83	-0.71	0.94	0.081	-0.19	-0.048	-0.07	0.35	-0.27	0.69	1	-0.34	-0.063
mag_var		-0.39	-0.58	-0.41	0.33	-0.42	0.31	-0.26	-0.2	-0.096	-0.023	0.05	-0.081	0.27	-0.22	-0.34	1	-0.15
R		-0.47	-0.37	0.12	0.25	0.11	-0.2	-0.098	0.057	0.02	-0.032	0.034	0.14	0.29	-0.13	-0.063	-0.15	1
	rank_order	lat	long	altb	baro_a	alt_msl	oat	ias	vspd	pitch	roll	hdg	volt1	amp1	e1_oil_t	e1_rpm	mag_var	hal

- 0.75

- 0.50

- 0.25

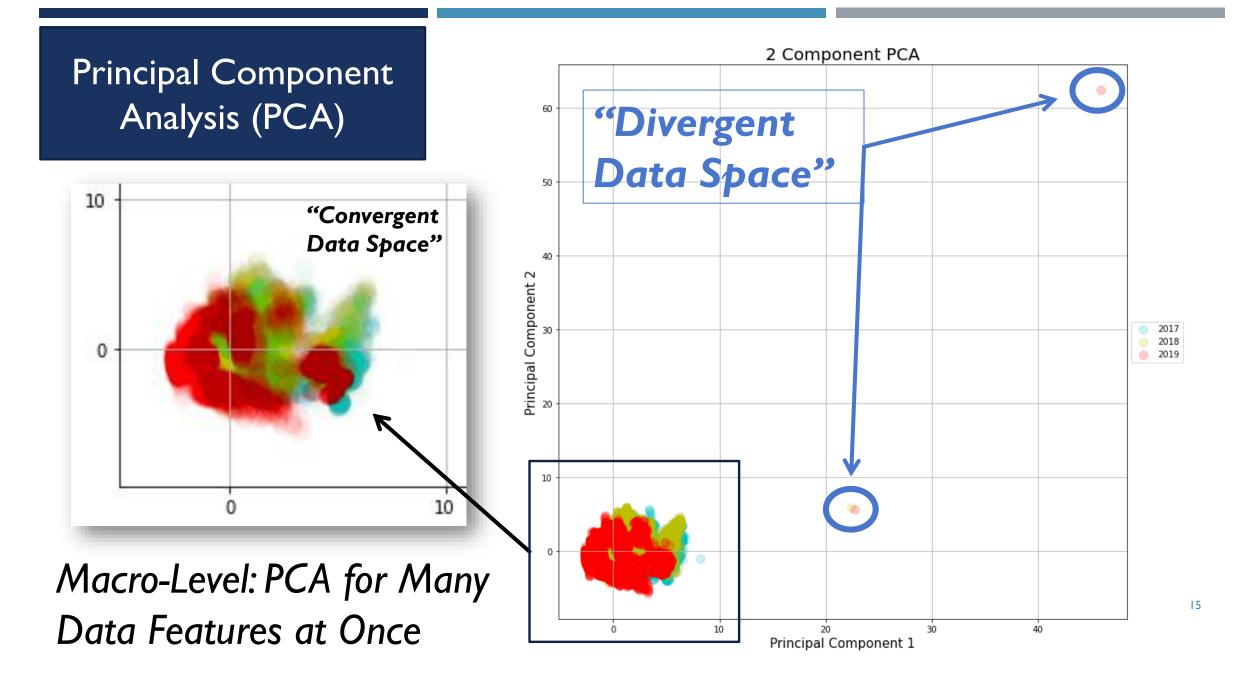
- 0.00

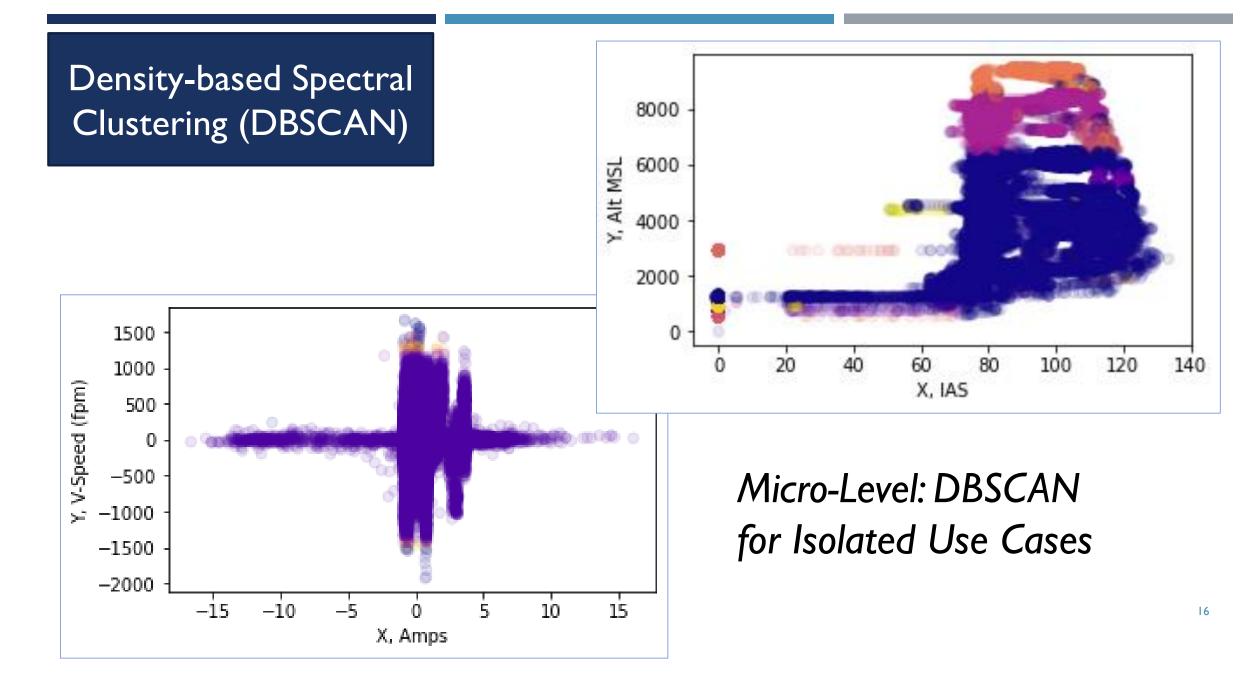
- -0.5

- -0.25

- -0.75

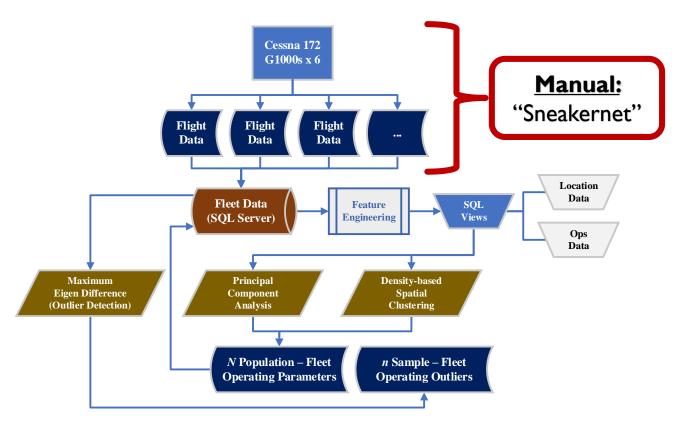
14





LIMITATIONS

- Operating Parameters (Limited to Cessna 172)
- Aircraft within a Training Environment
- Portions of Data Automation Systems are Manual (Sneakernet)
- Scalar Eigenvalue
 Integration



LESSONS LEARNED

- Use a prototype method, spread your initial data capture across a few aircraft
- Data feature engineering was the most time consuming (reworking and massaging the data)
- Leverage SQL Server Views for your Analytical Data Structure (i.e., SQL Server Views are Virtual Table structures)
- Null Logic Implementation (i.e., ISNULL() function required on all data features)
- Python made for an excellent backend system-level programing language
- PCA: High Level Feature Analysis | DBSCAN: Low Level Feature Analysis

CONCLUSIONS....AND...

Conclusions

- GI000 Flight Data is remarkably structured and easy to work with...
- Interpretation of the second secon
- Machine Learning Model development was easier than...
- ...automating the upfront "sneakernet" system.
- For larger datasets consider using Indexed Views to improve analytical data performance.

....FUTURE DIRECTIONS

Areas for Future Research

- Design Science Wireless transmission of fleet data via flight line access point system.
 - Future testing of onboard microcontroller/communication structure
 - Isolating the Scalar Eigenvalues at time interval level (not just at the day level)
- Additional Categorical Data Capture
 - Maintenance Records
 - Type of Training Flight Performed (IFR, VFR, MVFR, LIFR, etc...)
- SQL Server Engine Optimization
 - Support for Inline Data Predictions (i.e., native T-SQL)



Michael J. Pritchard, PHD Associate Dean for Research & Graduate Studies Assistant Professor, Machine Learning & Autonomous Systems <u>mjp001@ksu.edu</u> | (785) 979-0706

THANK YOU!

SELECTED REFERENCES

- Aziz, N. (2012). Detection of Outliers in Multivariate Data: A Method Based on Influence Eigen. In Proceedings of the World Congress on Engineering 2012 (Vol. 1).
- Cartocci, N., Costante, G., Napolitano, M. R., Valigi, P., Crocetti, F., & Fravolini, M. L. (2020, September). PCA methods and evidence based filtering for robust aircraft sensor fault diagnosis. In 2020 28th Mediterranean Conference on Control and Automation (MED) (pp. 550-555). IEEE.
- Dufrenois, F., & Noyer, J. C. (2015, July). Generalized eigenvalue proximal support vector machines for outlier description. In 2015 International Joint Conference on Neural Networks (IJCNN) (pp. 1-9). IEEE.
- Fala, N. (2019). Data-driven safety feedback as part of debrief for General Aviation pilots (Doctoral dissertation, Purdue University Graduate School).
- Jajo, N. K., & Matawie, K. M. (2009, July). Eigenvalues Application in Robust outlier Detection. In 24th International Workshop on Statistical Modelling (p. 182).
- Li, L., Gariel, M., Hansman, R. J., & Palacios, R. (2011, October). Anomaly detection in onboard-recorded flight data using cluster analysis. In 2011 IEEE/AIAA 30th Digital Avionics Systems Conference (pp. 4A4-1). IEEE.
- Puranik, T. G., & Mavris, D. N. (2018). Anomaly detection in general-aviation operations using energy metrics and flight-data records. *Journal of Aerospace Information Systems*, 15(1), 22-36.
- Reddy, K. K., Sarkar, S., Venugopalan, V., & Giering, M. (2016). Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach. In Annual Conference of the PHM Society (Vol. 8, No. 1).
- Sheridan, K., Puranik, T. G., Mangortey, E., Pinon-Fischer, O. J., Kirby, M., & Mavris, D. N. (2020). An application of dbscan clustering for flight anomaly detection during the approach phase. In AIAA Scitech 2020 Forum (p. 1851).
- Smart, E., Brown, D., & Denman, J. (2012). A two-phase method of detecting abnormalities in aircraft flight data and ranking their impact on individual flights. IEEE Transactions on Intelligent Transportation Systems, 13(3), 1253-1265.
- Wang, Y., Wang, X., & Wang, X. L. (2016, July). A spectral clustering based outlier detection technique. In International Conference on Machine Learning and Data Mining in Pattern Recognition (pp. 15-27). Springer, Cham.