

RBQM and Failure Prediction Using Python

Vidya Muthukumar, Senior Data Strategist/Clinical Data Manager – IQVIA Biotech

Disclaimer

The views and opinions expressed in the following PowerPoint slides are those of the individual presenter and should not be attributed to IQVIA, or any organization with which the presenter is employed or associated.

Table of Contents



Synopsis, Purpose and Model Setup



Data Structure



Problem Statement & Model Assumptions

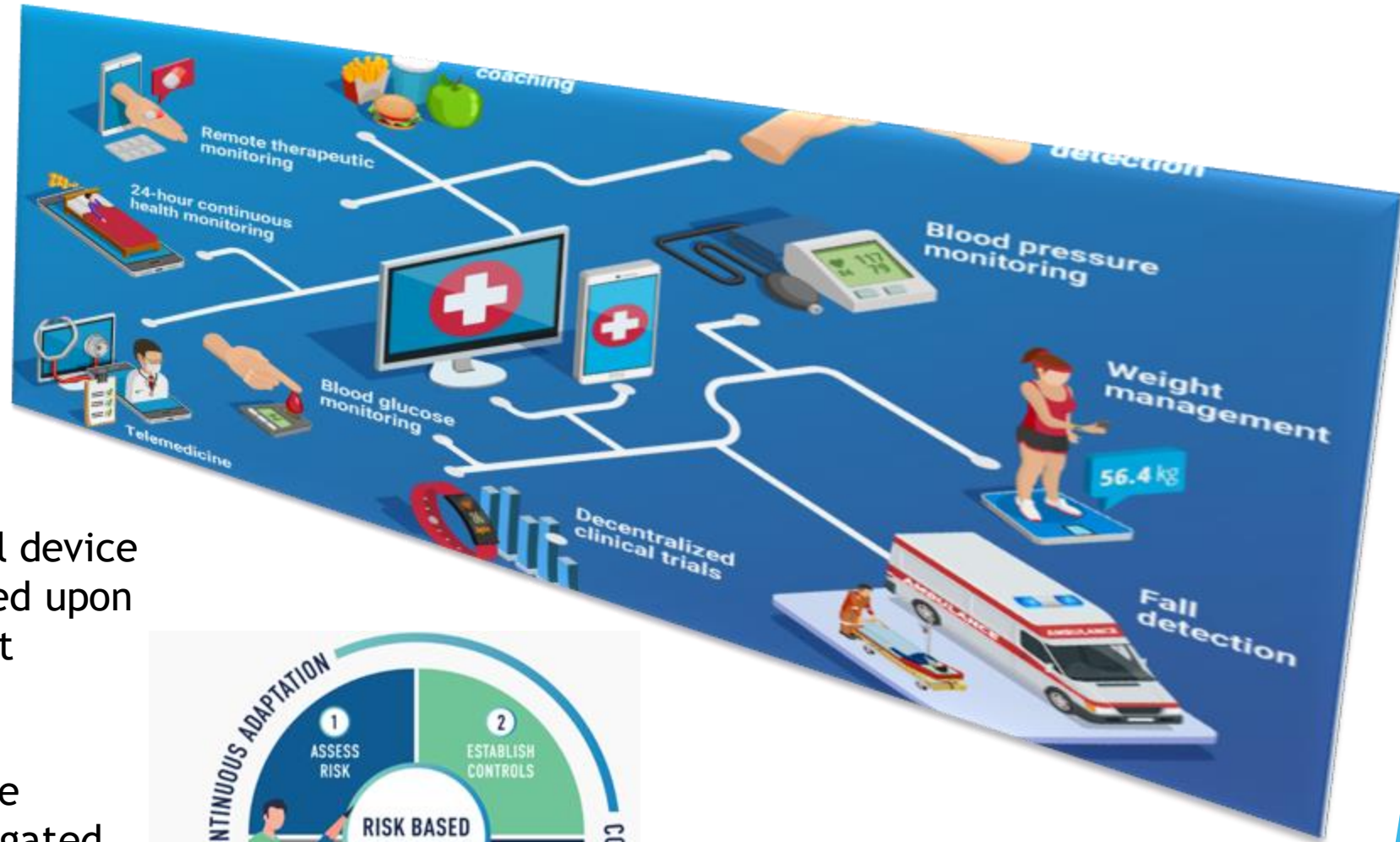
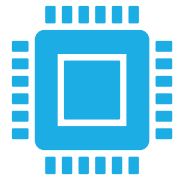
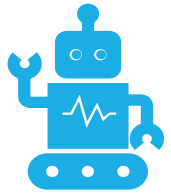


Data Analysis for Model



Conclusion and Suggestions for Future Analysis

Synopsis, Purpose and Model Setup



- ❑ Remote health monitoring device trials and optimizing error detection in wearable devices
- ❑ Using logistic regression to model device failure (dependent variable) based upon the given attributes (independent variables)
- ❑ Purpose is to test a sample device monitoring telemetry daily aggregated log data using logistic regression even before setting up SDV/tSDV modules in EDC for RBQM



Data Structure Specifications At-a-Glance

date: YYYY-MM-DD

Device_id: device_id

Failure 0 - no , 1 - yes

Attribute1/attribute9: daily aggregated
telemetry log data

Test data: device_failure.csv.zip

	A	B	C	D	E	F	G	H	I	J	K	L
1	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6	attribute7	attribute8	attribute9
2	1/1/2015	S1F01085	0	215630672	56	0	52	6	407438	0	0	7
3	1/1/2015	S1F0166B	0	61370680	0	3	0	6	403174	0	0	0
4	1/1/2015	S1F01E6Y	0	173295968	0	0	0	12	237394	0	0	0
5	1/1/2015	S1F01JE0	0	79694024	0	0	0	6	410186	0	0	0
6	1/1/2015	S1F01R2B	0	135970480	0	0	0	15	313173	0	0	3
7	1/1/2015	S1F01TD5	0	68837488	0	0	41	6	413535	0	0	1
8	1/1/2015	S1F01XDJ	0	227721632	0	0	0	8	402525	0	0	0
9	1/1/2015	S1F023H2	0	141503600	0	0	1	19	494462	16	16	3
10	1/1/2015	S1F02A0J	0	8217840	0	1	0	14	311869	0	0	0
11	1/1/2015	S1F02DZ2	0	116440096	0	323	9	9	407905	0	0	164
12	1/1/2015	S1F02EVN	0	112348104	0	0	0	7	388146	0	0	1
13	1/1/2015	S1F02L38	0	223938928	0	0	0	2	215169	0	0	8
14	1/1/2015	S1F02MGA	0	44399688	0	266	1	6	399286	0	0	2269
15	1/1/2015	S1F02P76	0	104131304	1536	0	175	11	301679	0	0	0
16	1/1/2015	S1F02VAX	0	61019512	168	2	521	3	380496	0	0	3
17	1/1/2015	S1F02WFT	0	44348552	5160	14	1074	11	249515	0	0	21
18	1/1/2015	S1F0318A	0	35018688	0	0	0	9	394890	0	0	5
19	1/1/2015	S1F0322R	0	34540712	0	0	0	9	411399	0	0	0
20	1/1/2015	S1F0330P	0	125539768	0	0	12	14	297284	0	0	5
21	1/1/2015	S1F035SJ	0	220392160	0	0	0	9	389730	0	0	0

Analysis

Model selection and effectiveness

Technology: Python

Problem Statement and Model Assumptions

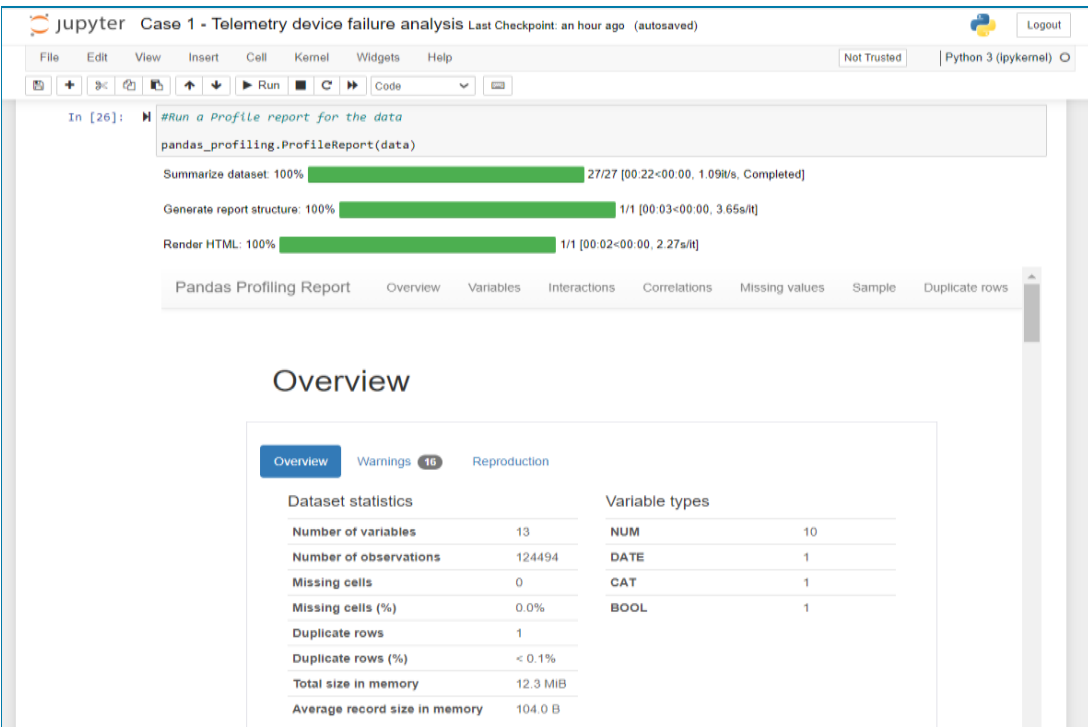
❑ Problem Statement

- Predict device failure given the daily aggregated device telemetry attribute log readings

❑ Assumptions

- Each device is being used by one person only
- Device can be for repeat use at different dates by same person

Data Analysis for Model



The screenshot shows a JupyterLab interface with a notebook titled 'Case 1 - Telemetry device failure analysis'. The notebook contains a cell with the following code:

```
In [1]: #####Case 1 uses aggregated telemetry data for 9 attributes with date, device and failure

#####Objective is to predict failure by device given the current 9 attributes, deviceID, date and Boolean failure data.

#####Import key packages and Load Case 1 data
import pandas as pd
import os
import seaborn as sns
import numpy as np
import pandas_profiling
from pandas_profiling import ProfileReport
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report
%matplotlib inline

#####Quick read of the head/tail data
data=pd.read_csv('Case_1_Data_device_failure.csv')
data.head(5)
```

The output of the code is a table of data:

	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6	attribute7	attribute8	attribute9
0	1/1/2015	S1F01085	0	215630672	56	0	52	6	407438	0	0	7
1	1/1/2015	S1F01668	0	61370680	0	3	0	6	403174	0	0	0
2	1/1/2015	S1F01E6Y	0	173295968	0	0	0	12	237394	0	0	0
3	1/1/2015	S1F01JE0	0	79694024	0	0	0	6	410186	0	0	0
4	1/1/2015	S1F01R2B	0	135970480	0	0	0	15	313173	0	0	3

- ❑ Analysis is about device failure
- ❑ Exploratory and preliminary data analysis was done using Python tools such as ProfileReport, Box plots

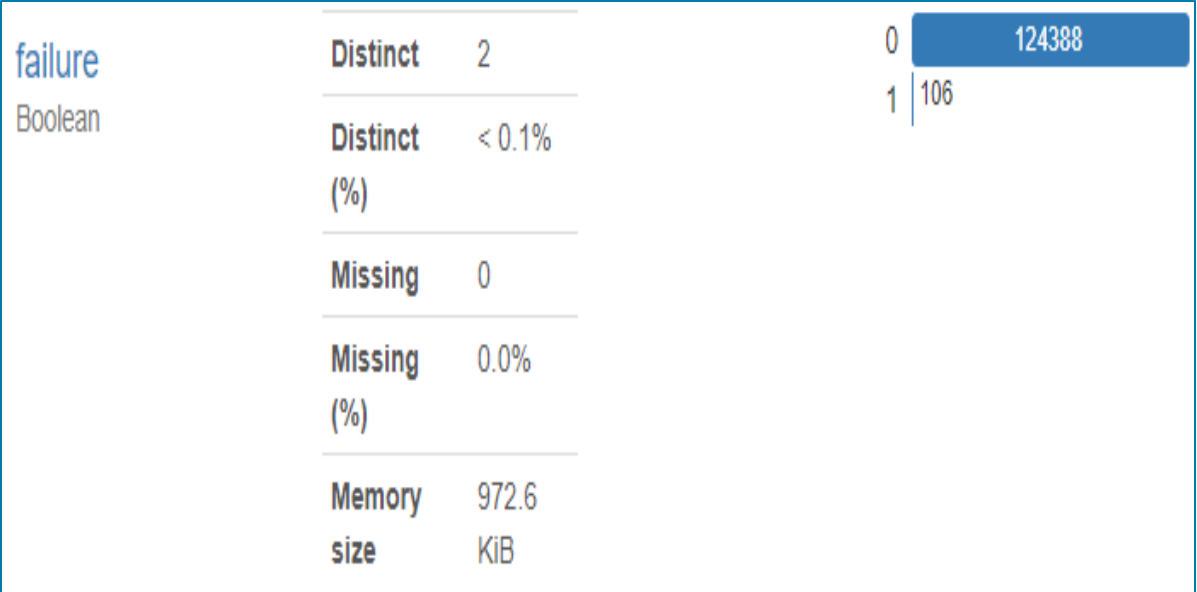
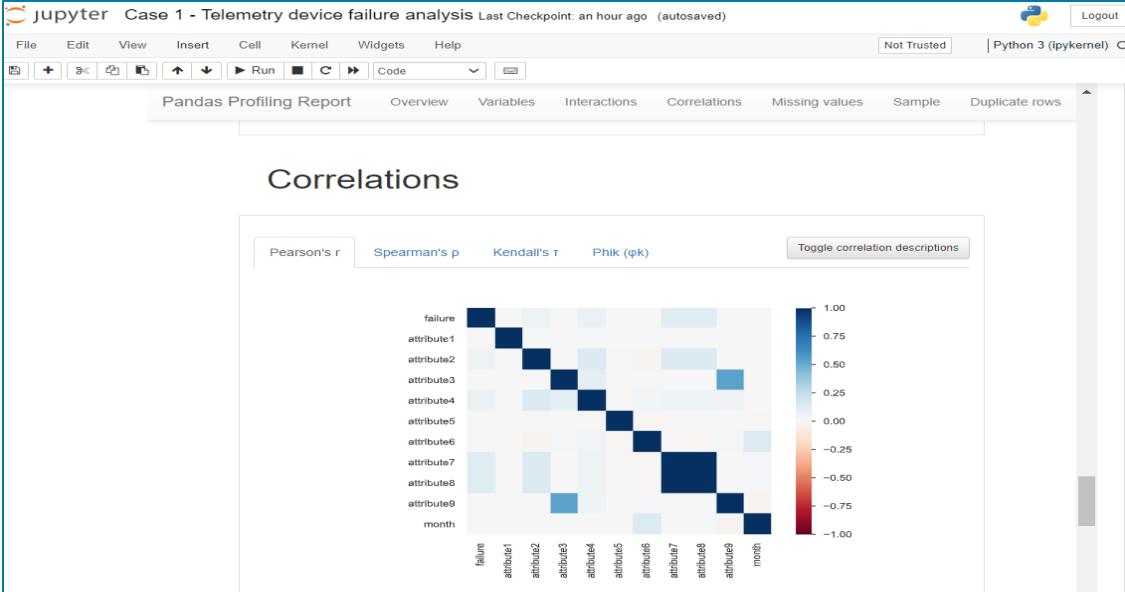
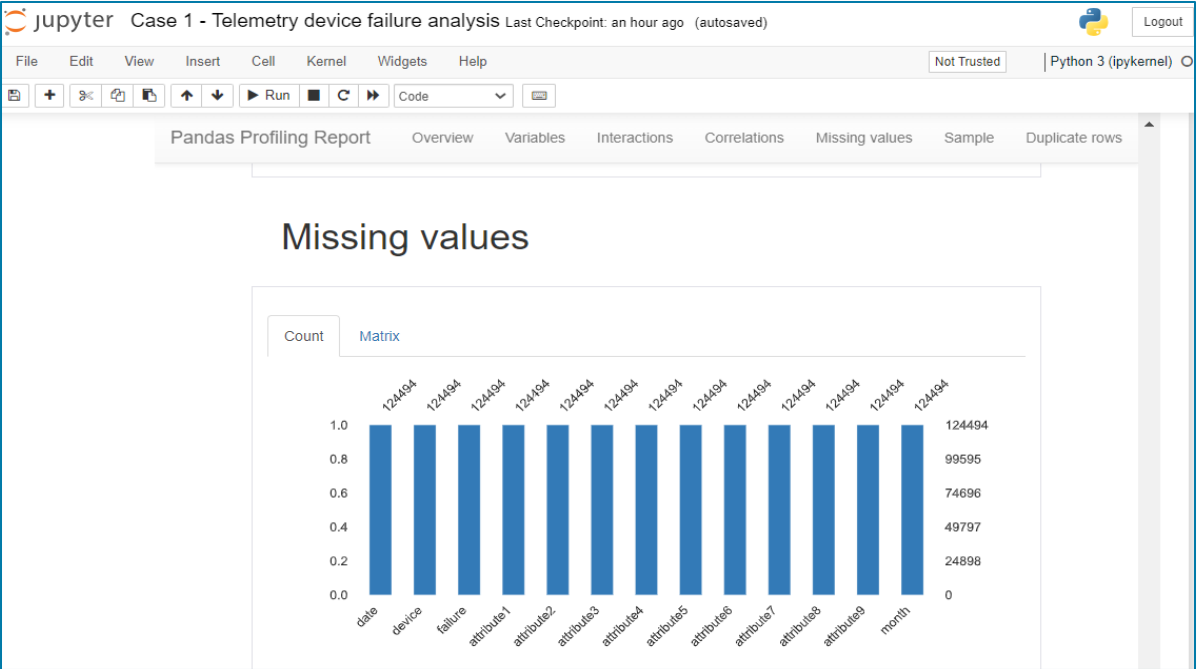
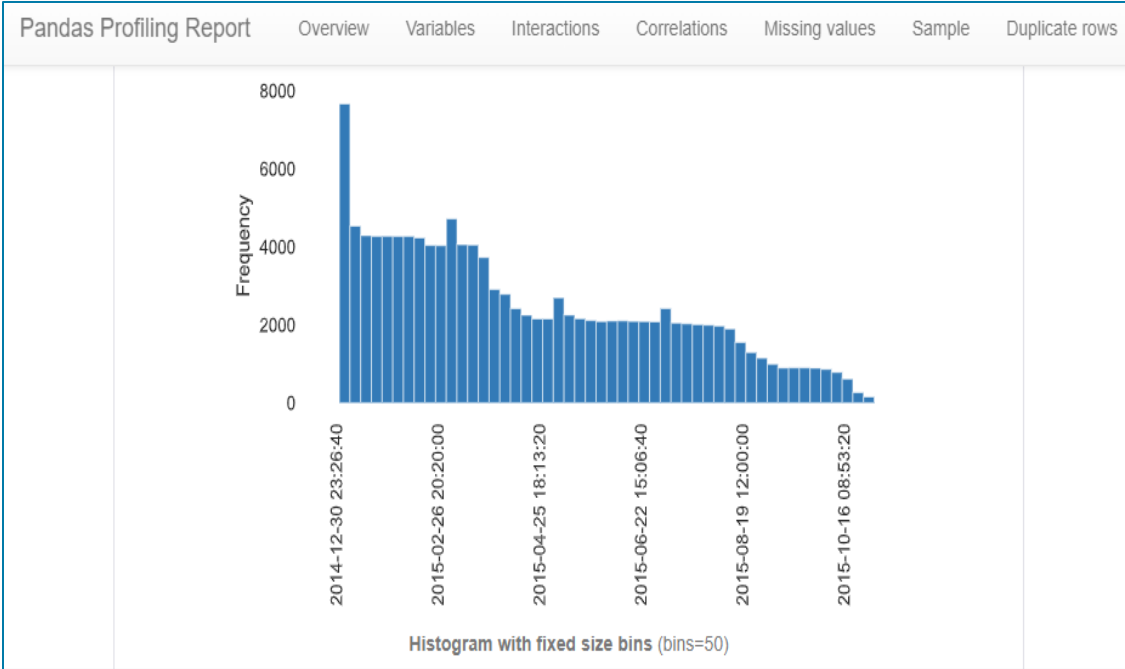
The screenshot shows a JupyterLab interface with a notebook titled 'Case 1 - Telemetry device failure analysis'. The notebook contains a cell with the following code:

```
In [2]: data.tail()
```

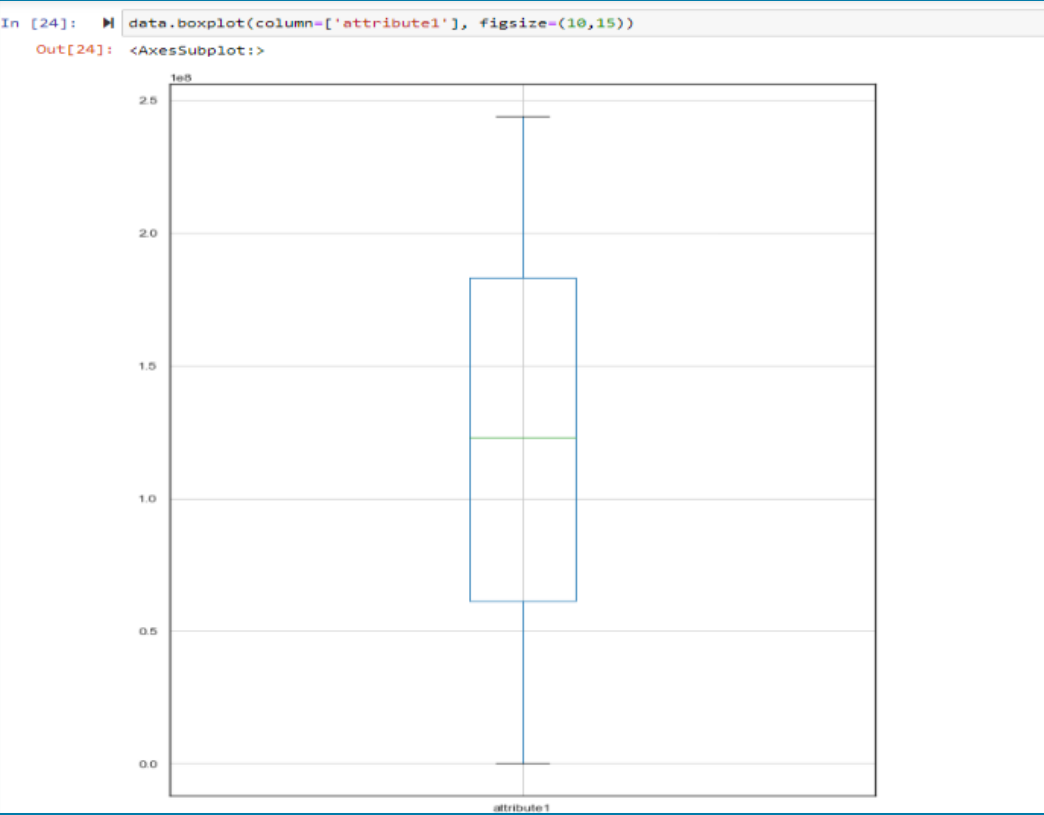
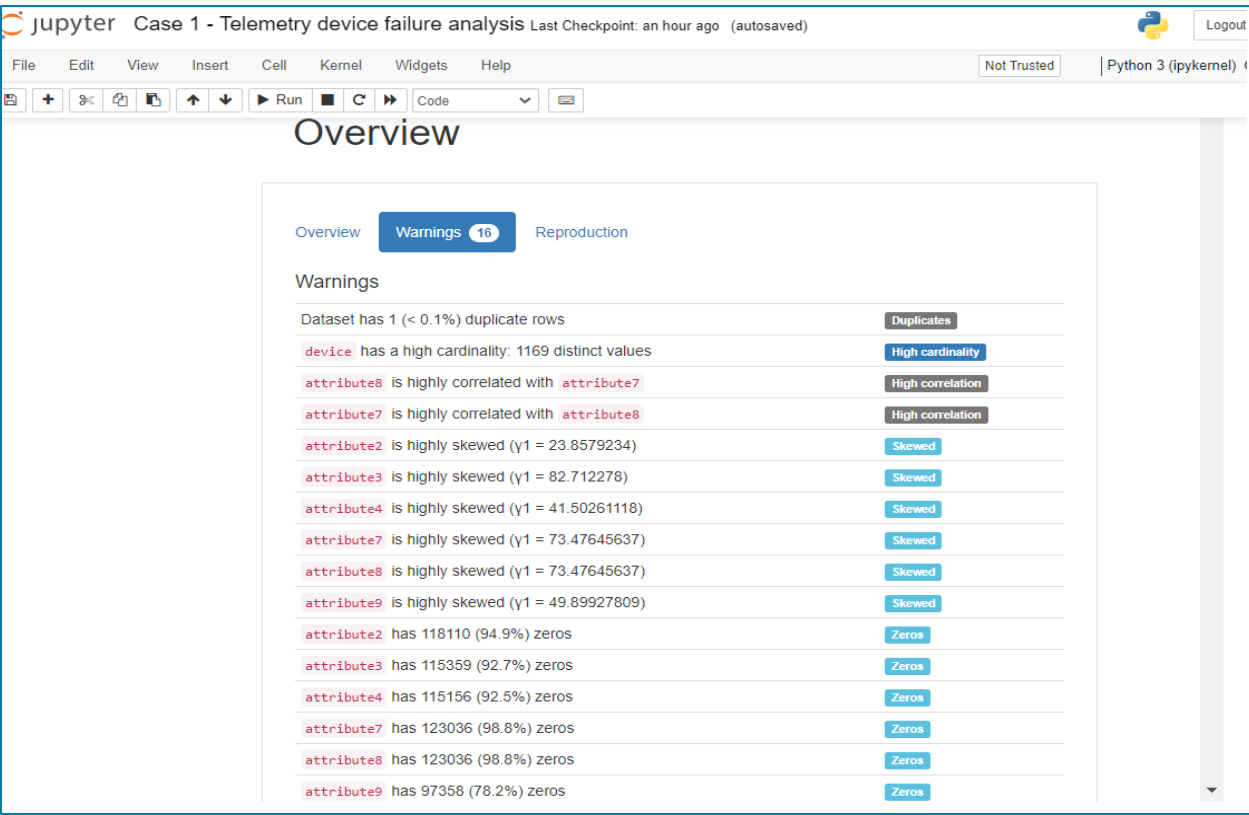
The output of the code is a table of data:

	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6	attribute7	attribute8	attribute9
124489	11/2/2015	Z1F0MA1S	0	18310224	0	0	0	10	353705	8	8	0
124490	11/2/2015	Z1F0Q8RT	0	172566880	96	107	4	11	332792	0	0	13
124491	11/2/2015	Z1F0QK05	0	19029120	4832	0	0	11	350410	0	0	0
124492	11/2/2015	Z1F0QL3N	0	226953408	0	0	0	12	358980	0	0	0
124493	11/2/2015	Z1F0QLC1	0	17572840	0	0	0	10	351431	0	0	0

Data Analysis for Model



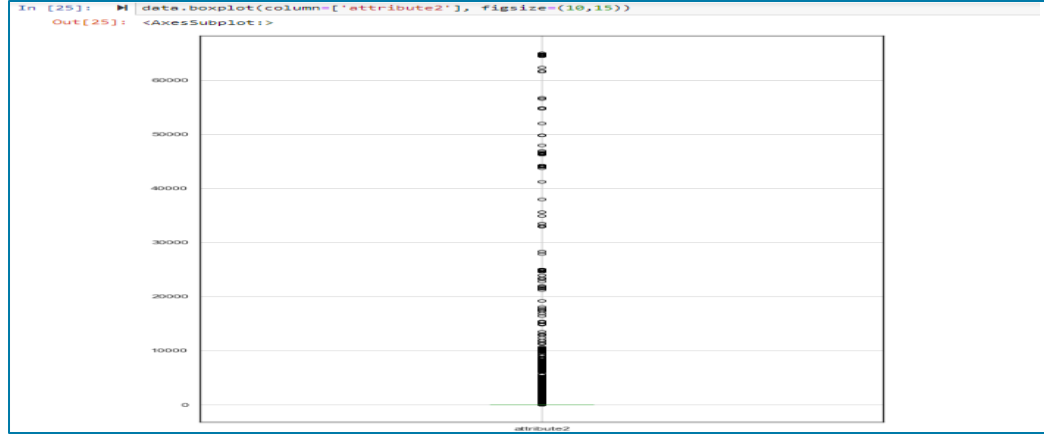
Data Analysis for Model



Duplicate rows

Most frequent

	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attr
0	2015-07-10	S1F0R4Q8	0	192721392	0	0	0	8	213



Data Analysis for Model

```
In [95]: # Step 3: Create a model and train it
model = LogisticRegression(fit_intercept = True, C = 1e12)
model.fit(device_means2, np.ravel(max_result, order='C'))
model.fit(device_means2, max_result)
```

```
In [115]: print(dev_fail.summary())
```

```
=====
                        Logit Regression Results
=====
Dep. Variable:          failure    No. Observations:          1169
Model:                  Logit      Df Residuals:              1160
Method:                  MLE        Df Model:                 8
Date:                   Wed, 17 Aug 2022    Pseudo R-squ.:         0.02954
Time:                   11:50:30      Log-Likelihood:        -344.99
converged:               True         LL-Null:               -355.49
Covariance Type:        nonrobust      LLR p-value:           0.007134
=====
```

	coef	std err	z	P> z	[0.025	0.975]
Intercept	-2.5583	0.686	-3.727	0.000	-3.904	-1.213
attribute1	1.641e-09	4.97e-09	0.330	0.741	-8.1e-09	1.14e-08
attribute2	2.806e-05	1.61e-05	1.744	0.081	-3.47e-06	5.96e-05
attribute3	-0.0006	0.002	-0.330	0.741	-0.004	0.003
attribute4	0.0023	0.001	1.609	0.108	-0.000	0.005
attribute5	0.0159	0.006	2.558	0.011	0.004	0.028
attribute6	-8.942e-07	1.07e-06	-0.838	0.402	-2.98e-06	1.2e-06
attribute7	0.0154	0.009	1.803	0.071	-0.001	0.032
attribute9	-6.39e-05	0.000	-0.229	0.819	-0.001	0.000

```
=====
```

```
In [119]: dev_fail.pred_table()
```

```
Out[119]: array([[1060.,   3.],
                 [ 104.,   2.]])
```

```
In [96]: print(model.coef_)
# print(model.intercept)
```

```
[[-1.45465653e-08  9.79205536e-08 -1.01467323e-09  1.02154132e-09
  2.09212390e-10 -2.03640257e-06  2.61251056e-10 -1.44239983e-09]]
```

```
In [114]: dev_fail = logit("failure ~ attribute1 + attribute2 + attribute3 + attribute4 + attribute5 + attribute6 + attribute7 + attribute9",
                           device_means3).fit()
```

```
Optimization terminated successfully.
Current function value: 0.295114
Iterations 10
```

- Used mean values of the attributes for each device as the independent variables
- Null Hypothesis: The 9 device attributes don't affect device failure
- Alternate Hypothesis: The 9 device attributes affect device failure; [if p-value < 0.1, t-stat > 1.65 or so]
- Device failure = $\alpha + \beta_1 * \text{Attribute1} + \beta_2 * \text{Attribute2} + \beta_3 * \text{Attribute3} + \dots + \beta_9 * \text{Attribute9}$
- P-values of < 0.1 => imply significant effect on device failure

Conclusion and Suggestions for Further Analysis

- ❑ The logistic regression model seems to perform well in identifying true negatives but does not do as well in predicting true positives. The overall explanatory power of the model seems to be low (Low R^2).
- ❑ Possible reasons:
 - Multiple people using same device
 - Telemetry data could be after device failure if device failed, was repaired, and reused by a patient.

Questions?