**Introduction to Python Programming**

**Assignment No 5**

**“Major Project”**

CSE-7702



Name of Candidate : Name of Advisor(s)

**Devender Sharma Dr Sandeep Nagar**

Email ID of Candidate : Department, Institute

**devender.sharma@gdgoenka.ac.in Mech Engg Dept**

Enrolment no. Date of Submission

**150020602002 30 Jan 2017**

**Project Report (Major Project)**

**Aim : To develop a simple regression analysis code for digital flight data recorded in an aircraft for possible anomaly detection of aircraft operation to safeguard against a potential safety hazard.**

**Introduction**

Modern aircraft are capable of recording hundreds of parameters during flight. This fact not only facilitates the investigation of an accident or a serious incident, but also provides the opportunity to use the recorded data to predict future aircraft behavior. It is believed that, by analyzing the recorded data, one can identify precursors to hazardous behavior and develop procedures to mitigate the problems before they actually occur. Because of the enormous amount of data collected during each flight, it becomes necessary to identify the segments of data that contain useful information.

**Regression Analysis**

Regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear or polynomial (2nd order or higher) regression model.

Before attempting to fit a model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable causes the other (for example, higher SAT scores do not cause higher college grades), but that there is some significant association between the two variables. In the instant case, analyzing the strength of the relationship between certain flight parameters recorded by the flight data recorder can be a helpful tool in determining the anomaly in aircraft performance with potential safety hazards.. Any deterioration in the association between the proposed explanatory and dependent variables can be utilized for preventibve flight safety measures. Fitting a regression model to the flight data therefore provides a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form Y = a + bX, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when x = 0).

Least-Squares Regression

The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values.

Outliers and Influential Observations

After a regression line has been computed for a group of data, a point which lies far from the line (and thus has a large residual value) is known as an outlier. In the instant case, such points may indicate a anomalies in the flight operations having potential hazards. If a point lies far from the other data in the horizontal direction, it is known as an influential observation. The reason for this distinction is that these points have may have a significant impact on the slope of the regression line.

Residuals

Once a regression model has been fit to a group of data, examination of the residuals (the deviations from the fitted line to the observed values) allows the modeler to investigate the deviation from the linear or second order relationship that . For example, the Lift and Drag characteristics have second order association with the velocity of the aircraft. Large number of residuals in a second order regression analysis shall indicate anomaly that may need investigation on the aero engine performance.

Lurking Variables

If non-linear trends are visible in the relationship between an explanatory and dependent variable, there may be other influential variables to consider. A lurking variable exists when the relationship between two variables is significantly affected by the presence of a third variable which has not been included in the modeling effort. Identification of such a variable shall give useful indication of the potential flight safety hazards (if any).

**Anomaly Detection Algorithm**

This project presents the python based regression analysis code for anomaly detection algorithm and terminologies used in describing the algorithms: cluster, outlier, anomalies, and abnormal flights. Multiple patterns exist in real - world operational data, however, the number of common patterns is finite because operations of commercial airline flights are highly standardized and a majority of flights share a few most common data patterns.

To facilitate regtession analysis, a prior step is to transform raw FDR data into excel / csv format was undertaken in the FDM software analysis package.

The transformation can be per formed in two ways: 1) convert the data of each flight for a specific phase and then convert the data file of the original FDR data into a csv format.

After data transformation, analysis was performed on limited data set points (four parameters) identified as correlation variables, which represent common data patterns in the dataset; Data points that do not belong to any clusters are detected as outliers, which indicate uncommon data patterns. In the last step, anomaly detection is performed

based on rregression analysis result. Flight, outliers identified in the individual attributes plot analysis are also the anomalies detected. Finally, the anomalies can be summarized by the specific abnormal flights that have relatively more or severer anomalies are further investigated by the domain experts.

**Flight Data Recording on Aircraft**

Flight data monitoring system collects and analyzes aircraft operational parameters that are recorded on board the aircraft using flight data recorders or quick access recorders (QARs). These can typically record a large number of aircraft flight parameters several times a second for several days at a time, and are downloaded periodically when the aircraft reaches a suitable station or maintenance base.

The resulting data is stored in a large database and analyzed with special purpose software toidentify anomalous occurrences that exceed defined thresholds, often termed exceedance events, as well as long - term trends in operations. Once the data has been analyzed to identify any such events and trends, the raw data may or may not be preserved. Until recently, the data for each exceedance event was archived

Most FDM tools allow users to specify thresholds that define exceedances and then identify occurrences where the threshold was exceeded in the data. This project aims to archive all flight data and provide correlation analyses of limited no of flight parametrs. The algorithm can be further developed in to tools that allow data to be exported to sophisticated animation packages that provide a graphical or virtual representation of a flight or incident in question. This can even extend to an external view of the aircraft, showing the nominal and actual flight paths, an interior cockpit view showing the movement of the controls and current state of the instrument displays, and a tower view which can represent a viewpoint of the aircraft from any fixed location on the ground.

**Flight Parameters Considered**

As brought out above that the FDR records a large number of data over varying flight phases. A sample data for A 320 aircraft during trim flying conditions while cruising at 1200 Kph has been extracted for the purpose of this considered in the file named FDR.CSV. For the purpose of proving the concept / hypothesis outlined in the instant project, only four attributes no. 2,3,4 and 5 have been considered. The data set available in the csv file for flight reference time running along the flight as the first attribute is tabulated below for the remaining paramters:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P (deg/s) | | Q (deg/s) | R (deg/s) | PITCH  DEFLECTION (deg) | | ROLL  DEFLECTION(deg) | | | | | | | | | | | | | | | | | Yaw deflection (deg) | | | | |
| R\_{inertial} (deg/s) | | q bar (psf) | Reynolds Number | V\_{Total} (ft/s) | | V\_{Inertial} (ft/s) | | | | | | | | | | | | | | | | | UBody | | | | |
| Aero V\_{X Body} (ft/s) | | Aero V\_{Y Body} (ft/s) | Aero V\_{Z Body} (ft/s) | V\_{X\_{inertial}} (ft/s) | | V\_{Y\_{inertial}} (ft/s) | | | | | | | | | | | | | | | | | V\_{Z\_{inertial}} (ft/s) | | | | |
| V\_{Z\_{ecef}} (ft/s) | | V\_{North} (ft/s) | V\_{East} (ft/s) | V\_{Down} (ft/s) | | F\_{Drag} (lbs) | | | | | | | | | | | | | | | | | F\_{Side} (lbs) | | | | |
| F\_{Aero x} (lbs) | | F\_{Aero y} (lbs) | F\_{Aero z} (lbs) | Mass | | X\_{cg} | | | | | | | | | | | | | | | | | F\_{Gear x} (lbs) | | | | |
| F\_{Gear z} (lbs) | | I\_{xx} | I\_{xy} | I\_{xz} | | I\_{zz} | | | | | | | | | | | | | | | | | Y\_{cg} | | | | |
| F\_{Total z} (lbs) | | L\_{Aero} (ft-lbs) | M\_{Aero} ( ft-lbs) | N\_{Aero} (ft-lbs) | | L\_{Prop} (ft-lbs) | | | | | | | | | | | | | | | | |  | | | | |
| L\_{Total} (ft-lbs) | | M\_{Total} (ft-lbs) | N\_{Total} (ft-lbs) | Rho (slugs/ft^3) | | Absolute Viscosity | | | | | | | | | | | | | | | | | Kinematic Viscosity | | | | |
| Theta (deg) | | Z\_{ECI} (ft) | Psi (deg) | Alpha (deg) | | Beta (deg) | | | | | | | | | | | | | | | | |  | | | | |
| Z\_{ECI} (ft) | | systems/mixture-cmd-norm | X\_{ECEF} (ft) | guidance/delta-lat-rad | | Earth Position Angle (deg) | | | | | | | | | | | | | | | | | guidance/wp- | | | | |
| ap/total-limited-roll-rate | | ap/roll-rate-saturation | ap/limited-roll-rate-error | ap/roll-  command-pid-control | | ap/roll-cmd-smoother | | | | | | | ap/roll-cmd-norm-output | | | | | ap/roll-command-selector-steering | | | | | |
| fcs/wing-leveler-ap-on-off | | fcs/roll-ap-error-pid | fcs/roll-ap-autoswitch | fcs/heading-true-degrees | | fcs/heading-error | | fcs/heading-error-bias-switch | | | | | | | | | | | | fcs/heading-corrected | | | | | | |
| fcs/heading-roll-error-lag | fcs/heading-roll-error | |  | fcs/altitude  -error | | | | | | | | | | fcs/alt-error-lag | | | | | | | | | fcs/hdot-command | | | | |
| Nose Gear stroke (ft) | | Nose Gear stroke velocity (ft/sec) | Nose Gear compress force (lbs) | Nose Gear wheel side force (lbs) | | Nose Gear wheel roll force (lbs) | | Nose Gear body X force (lbs) | | | Nose Gear body Y force (lbs) | | | | | Nose Gear wheel velocity vec X (ft/sec) | | | | | | | | |
| Nose Gear wheel velocity vec Y (ft/sec) | | Nose Gear wheel rolling velocity (ft/sec) | Nose Gear wheel side velocity (ft/sec) | Nose Gear wheel slip (deg) | | Left Main Gear WOW | | Left Main Gear stroke (ft) | | Left Main Gear stroke velocity (ft/sec) | | | | | Left Main Gear compress force (lbs) | | | | | | Left Main Gear wheel side force (lbs) | | | | |
| Left Main Gear wheel roll force (lbs) | | Left Main Gear body X force (lbs) | Left Main Gear body Y force (lbs) | Left Main Gear wheel velocity vec X (ft/sec) | | Left Main Gear wheel velocity vec Y (ft/sec) | | Left Main Gear wheel rolling velocityft/sec) | | Left Main Gear wheel side velocity (ft/sec) | | | | | Left Main Gear wheel slip (deg) | | | | |
| Right Main Gear WOW | | Right Main Gear stroke (ft) | Right Main Gear stroke velocity (ft/sec) | Right Main Gear compress force (lbs) | Right Main Gear wheel side force (lbs) | | Right Main Gear wheel roll force (lbs) | | | | Right Main Gear body X force (lbs) | | | | | Right Main Gear body Y force (lbs) | | | | |
| Right Main Gear wheel velocity vec X (ft/sec) | | Right Main Gear wheel velocity vec Y (ft/sec) | Right Main Gear wheel rolling velocity (ft/sec) | Right Main Gear wheel side velocity (ft/sec) | Right Main Gear wheel slip (deg) | | Total Gear Force\_X (lbs) | | | Total Gear Force\_Y (lbs) | | | | | Total Gear Force\_Z (lbs) | | | | |
| Total Gear Moment\_L (ft-lbs) | | Total Gear Moment\_M (ft-lbs) | Total Gear Moment\_N (ft-lbs) | IO320 Power Available (engine 0 in ft-lbs/sec) | IO320 HP (engine 0) | | | | IO320 equivalent ratio (engine 0) | | | | | | | | IO320 MAP (engine 0 in inHg) | | | | |
| Fixed-Pitch 75-inch Two-Blade Propeller PFactor Pitch (engine | | Fixed-Pitch 75-inch Two-Blade Propeller PFactor Yaw (engine 0) | Fixed-Pitch 75-inch Two-Blade Propeller Thrust (engine 0 in lbs) | Fixed-Pitch 75-inch Two-Blade Propeller RPM (engine 0) | vrp-gc-latitude deg | | | | | | | vrp-longitude deg | | | | | | | vrp-radius-ft | | | | | | | | |

Inputs from the recorded data (FDR.csv) file with following attributes have been considered for running the cvode:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attr No** | | **Attr Name** | |
| 0 | | Reference time | |
| 1 | | P (deg/s) | |
| 2 | | q (deg/s) | |
| 3 | | r (deg/s) | |
| 4 | | Pitch deflection | |
| 5 | | Roll deflection | |
| 6  **Coding.**  A total of two codes were developed, the first code plots the graph of the individual six attributes selected while the second one undertakes the Linear regression analysis. Details are as under : | | Yaw deflection | |
|  | | **Code 1**  import csv  import random as rm  import matplotlib.pyplot as pt  import numpy as nm  %matplotlib inline  from sklearn import linear\_model  t=0  x=[]  y=[]  ch=int(input("enter a number between 1 to 5 \n")) ####ch to store attributes data  with open('FDR.csv') as h: #######load csv file  r=csv.reader(h)  for i in r:  for j in i:  if j=='NA' or j=='n/a': #####load rows except NA & n/a  t=1  if t==0:  x.append(i)  else:  t=0  for i in range(1,len(x)):  y.append(x[i][3]) ######y has all data of interest###  y\_new=[]  for i in y:  y\_new.append(float(i)) #####y\_new has data of y in float as list###  x\_new=[]  #####################choose the attribute b/w 1-5 and add data to x\_new #################  if ch==1:  print (x[0][1])  for i in range(1,len(x)):  x\_new.append(float(x[i][1]))  if ch==2:  print (x[0][2])  for i in range(1,len(x)):  x\_new.append(float(x[i][2]))  if ch==3:  print (x[0][3])  for i in range(1,len(x)):  x\_new.append(float(x[i][3]))  if ch==4:  print (x[0][4])  for i in range(1,len(x)):  x\_new.append(float(x[i][4]))  if ch==5:  print (x[0][5])  for i in range(1,len(x)):  x\_new.append(float(x[i][5]))  ################# changing x\_new in matrix#############  x\_new=nm.matrix(x\_new)  x\_new=x\_new.transpose()  ############ last 300 is used for testing ######  x\_train=x\_new[:-300]  x\_test=x\_new[-300:]  ############ last 300 is used for testing ######  y\_train=y\_new[:-300]  y\_test=y\_new[-300:]  #############initializing the reg\_model #######  reg\_model = linear\_model.LinearRegression()  ############### training of data using fit ##########  reg\_model.fit(x\_train,y\_train)  ############### pred is used to predict the points of test data #####  pred = reg\_model.predict(x\_test)  pos,neg=0,0  ans=0.0  for i in range (0,len(pred)):  pos+=pow(pred[i]-y\_new[i],2)  ans=float(pos)/len(pred)  ################# scatter is used to plot points of test data################  pt.scatter(x\_test,y\_test,color='blue')  ############## plot is used to plot line of test data #############  pt.plot(x\_test,pred,color='red')  print ("--> error is : "+str(ans))  print ("--> accuracy is :"+str(100-ans)) | |

**Graph of individual attributes**

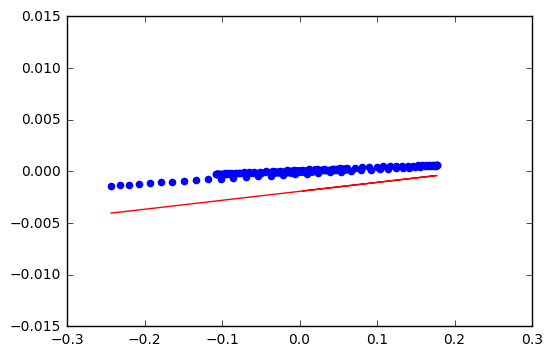
enter a number between 1 to 5

1

P (deg/s)

--> error is : 2.0819830147932782e-05

--> accuracy is :99.99997918016984



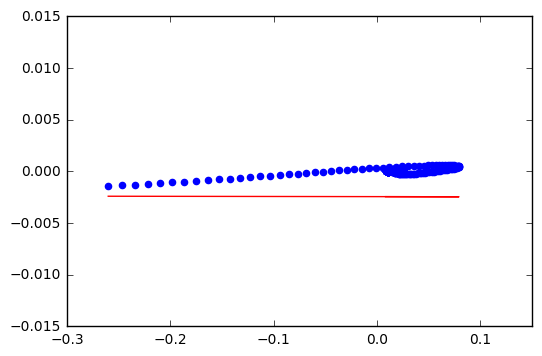
enter a number between 1 to 5

2

Q (deg/s)

--> error is : 1.8996364801857344e-05

--> accuracy is :99.9999810036352



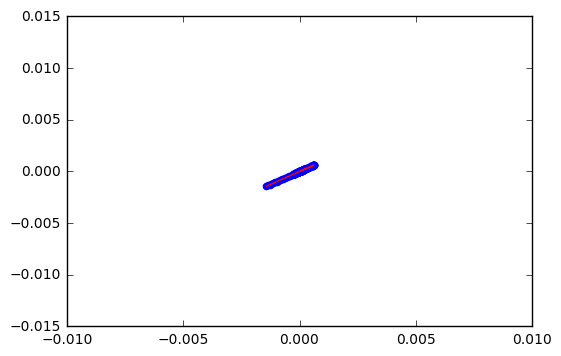
enter a number between 1 to 5

3

R (deg/s)

--> error is : 2.5434122233641768e-05

--> accuracy is :99.99997456587776



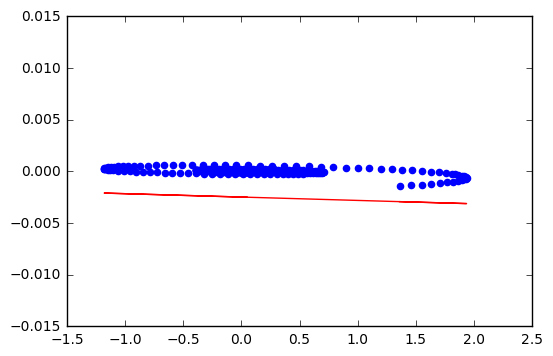
enter a number between 1 to 5

4

Pitch deflection (deg)

--> error is : 1.8738551340698437e-05

--> accuracy is :99.99998126144865



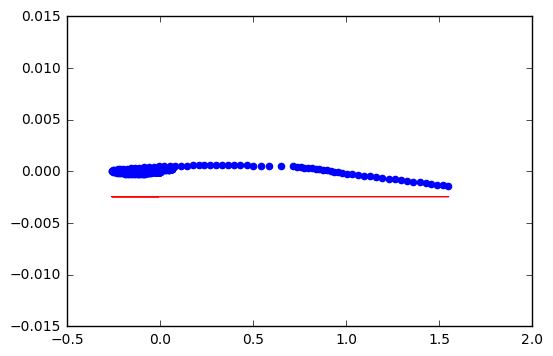
enter a number between 1 to 5

5

Roll deflection (deg)

--> error is : 1.8987791607340194e-05

--> accuracy is :99.9999810122084



**Code2**

import csv

import numpy as nm

import matplotlib.pyplot as pt

import itertools

%matplotlib inline

from sklearn import linear\_model

t=0

x=[]

a=[]

b=[]

#ch=int(input("enter a number between 2 to 5 except \n")) ####ch to store attributes data

with open('FDR.csv') as h: #######load csv file

r=csv.reader(h)

for i in r:

for j in i:

if j=='NA' or j=='n/a': #####load rows except NA & n/a

t=1

if t==0:

x.append(i)

else:

t=0

x.remove(x[0])

temp=[]

for i in range(0,len(x)):

temp.append((float(x[i][2]), float(x[i][3]), float(x[i][4]), float(x[i][5])))

for i in temp:

a.append([i[0],i[1],i[2],i[3]])

b.append(float(i[2]))

lis=[]

stuff=[2,3,4,5]

for l in range(0,len(stuff)+1):

for subset in itertools.combinations(stuff,l):

lis.append(list(subset))

lis.remove(lis[0])

an=[]

p=[ ]

error=[]

for k in lis:

for i in range(0,len(x)):

for j in k:

if j==2:

p.append(a[i][0])

if j==3:

p.append(a[i][1])

if j==4:

p.append(a[i][2])

if j==5:

p.append(a[i][3])

an.append(p)

p=[]

an=nm.matrix(an)

reg\_model = linear\_model.LinearRegression()

############### training of data using fit ##########

reg\_model.fit(an,b)

############### pred is used to predict the points of test data #####

pred = reg\_model.predict(an)

pos,neg=0,0

ans=0.0

for i in range (0,len(pred)):

pos+=pow(pred[i]-b[i],2)

ans=float(pos)/len(pred)

print (ans,k)

error.append(ans)

#print reg\_model.coef\_

an=[]

for i in range(0,len(lis)):

pt.bar(i,error[i])

pt.xlabel("Subset Combinations")

pt.ylabel("Anomaly")

pt.title("Histogram Plot")

import csv

import numpy as nm

import matplotlib.pyplot as pt

import itertools

%matplotlib inline

from sklearn import linear\_model

t=0

x=[]

a=[]

b=[]

#ch=int(input("enter a number between 2 to 5 except \n")) ####ch to store attributes data

with open('FDR.csv') as h: #######load csv file

r=csv.reader(h)

for i in r:

for j in i:

if j=='NA' or j=='n/a': #####load rows except NA & n/a

t=1

if t==0:

x.append(i)

else:

t=0

x.remove(x[0])

temp=[]

for i in range(0,len(x)):

temp.append((float(x[i][2]), float(x[i][3]), float(x[i][4]), float(x[i][5])))

for i in temp:

a.append([i[0],i[1],i[2],i[3]])

b.append(float(i[2]))

lis=[]

stuff=[2,3,4,5]

for l in range(0,len(stuff)+1):

for subset in itertools.combinations(stuff,l):

lis.append(list(subset))

lis.remove(lis[0])

an=[]

p=[ ]

error=[]

for k in lis:

for i in range(0,len(x)):

for j in k:

if j==2:

p.append(a[i][0])

if j==3:

p.append(a[i][1])

if j==4:

p.append(a[i][2])

if j==5:

p.append(a[i][3])

an.append(p)

p=[]

an=nm.matrix(an)

​

reg\_model = linear\_model.LinearRegression()

############### training of data using fit ##########

reg\_model.fit(an,b)

############### pred is used to predict the points of test data #####

pred = reg\_model.predict(an)

​

pos,neg=0,0

ans=0.0

for i in range (0,len(pred)):

pos+=pow(pred[i]-b[i],2)

ans=float(pos)/len(pred)

​

print (ans,k)

error.append(ans)

#print reg\_model.coef\_

an=[]

for i in range(0,len(lis)):

pt.bar(i,error[i])

pt.xlabel("Subset Combinations")

pt.ylabel("Anomaly")

pt.title("Histogram Plot")

2.3980509630066646 [2]

2.008175186585198 [3]

4.942908731766664e-31 [4]

2.4031665729902127 [5]

2.0040955117891635 [2, 3]

0.0 [2, 4]

2.389790641544934 [2, 5]

1.8296632006258803e-31 [3, 4]

1.9940805033176174 [3, 5]

4.962558936301083e-31 [4, 5]

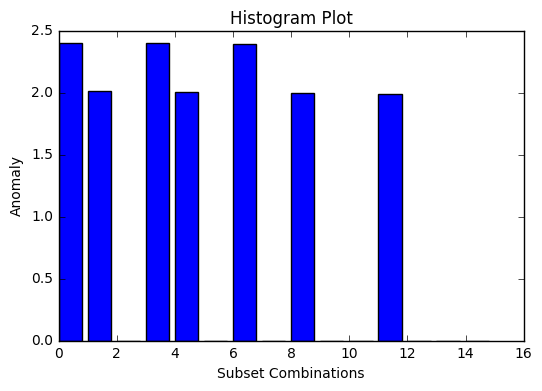
9.956873170585943e-31 [2, 3, 4]

1.9893730255359905 [2, 3, 5]

3.3510200758474564e-29 [2, 4, 5]

9.580258989911409e-29 [3, 4, 5]

5.162916677380705e-29 [2, 3, 4, 5]



Error corresponding to subset of selected attributes

Results

For best result we consider only 4 attribute i.e. attribute number 2,3, 4 &5. The attributes are already described in the above table. The subset of these attributes and their corresponding errors are shown in the above histogram. From above graph we are able to select attributes which give us highest anomalies that could be investigated further. SAs seen above that the anomalies of pitch rate reflect highest in the individual and subset score.