

Project 03 Team 6

November 27, 2021

1 Project 03: Chevron Equipment Maintenance Data

1.1 Team 6 Members

- Alejandro Bañuelos A01244596
- Aline Martínez Briones A00826215
- Marco Ortiz A00823250
- Maximiliano Contreras Serrano A00825377

```
[ ]: import pandas as pd
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from collections import Counter
```

```
[ ]: fontT = {'family' : 'Arial',
            'weight' : 'bold',
            'size'   : 20}

fontL = {'family' : 'Arial',
        'weight' : 'bold',
        'size'   : 18}

fontL2 = {'family' : 'Arial',
        'weight' : 'bold',
        'size'   : 6}

fontAx = {'family' : 'Arial',
        'weight' : 'normal',
        'size'   : 10}

fontTix = {'family' : 'Arial',
        'weight' : 'normal',
        'size'   : 6}
```

```
[ ]: df = pd.read_csv("WorkOrders15_18 Corrective.csv")
df.head()
```

```
[ ]: WorkOrder FieldProductionTeam \
0 15516284 RANGELY
1 15506143 VAN
2 15476748 VAN
3 15476785 VAN
4 15477769 VAN
```

```
Comment EquipmentCode \
0 CS03; LPGAS LINE FROZEN/PLUGGED C35CS03
1 401-1D: NEW CHEMICAL PUMP & ELECTRICAL CQ3253FL1
2 D: 14-7: BELTS ARE STILL LOOSE 22B0359734
3 D: NEIL 115 SW CVWU 64-1 INJECTION WELL NEED ... VAN
4 D: SWAIN 14 SWD THE NORTH DISPOSAL DISCHARGE ... VAN
```

```
EquipmentRollupCode EquipmentDescription EquipmentType \
0 C35CS03 COLLECTION STATION; 03 System
1 CQ3253FL1 LINE; PRODUCTION FLOW FOR CVWU 4011 System
2 22B0359734 PUMPUNIT; SWAIN WF 14-7 + Asset
3 VAN VAN OPERATIONS System
4 VAN VAN OPERATIONS System
```

```
EquipmentClass EquipmentCriticality StatusCode ... \
0 SE - Battery NA - Not Applicable C ...
1 Line - Production Flowline 3 - Low C ...
2 Pumping Unit 3 - Low C ...
3 SE - Field NA - Not Applicable C ...
4 SE - Field NA - Not Applicable C ...
```

```
TargetDate SchEndDate ActualDuration ScheduleCompliant PMOverdue \
0 01/01/2015 01/01/2015 -1 Future Week None
1 01/01/2015 01/01/2015 3 Future Week None
2 01/01/2015 01/01/2015 19 Future Week None
3 01/01/2015 01/01/2015 19 Future Week None
4 01/01/2015 01/01/2015 19 Future Week None
```

```
CompletedDate CompletedMonth CompletedWeekDay TBF TBF_Equipment
0 01/01/2015 1 5 0 4.0
1 01/01/2015 1 5 0 NaN
2 01/01/2015 1 5 0 1.0
3 01/01/2015 1 5 0 0.0
4 01/01/2015 1 5 0 0.0
```

[5 rows x 51 columns]

```
[ ]: column_names = df.columns
      print(column_names)
```

```
Index(['WorkOrder', 'FieldProductionTeam', 'Comment', 'EquipmentCode',
      'EquipmentRollupCode', 'EquipmentDescription', 'EquipmentType',
      'EquipmentClass', 'EquipmentCriticality', 'StatusCode',
      'StatusDescription', 'StatusShortDescription', 'Priority', 'WOType',
      'JobType', 'Action', 'Requirement', 'Cause', 'FailureReason',
      'Duration', 'GrossProductionLoss', 'AffectedProduction',
      'IsAffectingProduction', 'LaborCost', 'MaterialCost', 'TotalCost',
      'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole', 'Manufacturer',
      'Model', 'Safety', 'Reopened', 'CreatedBy', 'UpdatedBy', 'CreatedDate',
      'UpdatedDate', 'ReportDate', 'ReportMonth', 'ReportWeekDay',
      'TargetDate', 'SchEndDate', 'ActualDuration', 'ScheduleCompliant',
      'PMOverdue', 'CompletedDate', 'CompletedMonth', 'CompletedWeekDay',
      'TBF', 'TBF_Equipment'],
      dtype='object')
```

```
[ ]: ## Check for unique Values to see if any column is not bringing any
      ↳ information##
      df_uniques = df.nunique()
      df_uniques = df_uniques.values.tolist()
      print(df_uniques)
```

```
[89480, 56, 87299, 45782, 45586, 43328, 4, 332, 8, 2, 2, 2, 6, 1, 1, 35, 21, 50,
580, 169, 463, 392, 2, 1357, 1346, 4059, 1140, 78, 4, 4, 914, 5366, 2, 2, 1726,
329, 1590, 1021, 1594, 12, 7, 1467, 1466, 280, 2, 1, 1235, 12, 7, 4, 1020]
```

```
[ ]: ## Removing columns with only one unique value##

      del_cols = []

      for i in range(0, len(column_names)):
          print(column_names[i] + " " + str(i) + " " + str(df_uniques[i]))
          if df_uniques[i] == 1:
              del_cols.append(column_names[i])
              df = df.drop([column_names[i]], axis=1)
              print(column_names[i] + " Se elimino la columna")
```

```
WorkOrder 0 89480
FieldProductionTeam 1 56
Comment 2 87299
EquipmentCode 3 45782
EquipmentRollupCode 4 45586
EquipmentDescription 5 43328
EquipmentType 6 4
EquipmentClass 7 332
EquipmentCriticality 8 8
```

StatusCode 9 2
 StatusDescription 10 2
 StatusShortDescription 11 2
 Priority 12 6
 WOType 13 1
 WOType Se elimino la columna
 JobType 14 1
 JobType Se elimino la columna
 Action 15 35
 Requirement 16 21
 Cause 17 50
 FailureReason 18 580
 Duration 19 169
 GrossProductionLoss 20 463
 AffectedProduction 21 392
 IsAffectingProduction 22 2
 LaborCost 23 1357
 MaterialCost 24 1346
 TotalCost 25 4059
 Assigned 26 1140
 Trade 27 78
 TradeGroup 28 4
 SupervisorRole 29 4
 Manufacturer 30 914
 Model 31 5366
 Safety 32 2
 Reopened 33 2
 CreatedBy 34 1726
 UpdatedBy 35 329
 CreatedDate 36 1590
 UpdatedDate 37 1021
 ReportDate 38 1594
 ReportMonth 39 12
 ReportWeekDay 40 7
 TargetDate 41 1467
 SchEndDate 42 1466
 ActualDuration 43 280
 ScheduleCompliant 44 2
 PMOverdue 45 1
 PMOverdue Se elimino la columna
 CompletedDate 46 1235
 CompletedMonth 47 12
 CompletedWeekDay 48 7
 TBF 49 4
 TBF_Equipment 50 1020

```
[ ]: del_cols
```

```
[ ]: ['WOType', 'JobType', 'PMOverdue']
```

```
[ ]: column_names = df.columns  
print(column_names)
```

```
Index(['WorkOrder', 'FieldProductionTeam', 'Comment', 'EquipmentCode',  
      'EquipmentRollupCode', 'EquipmentDescription', 'EquipmentType',  
      'EquipmentClass', 'EquipmentCriticality', 'StatusCode',  
      'StatusDescription', 'StatusShortDescription', 'Priority', 'Action',  
      'Requirement', 'Cause', 'FailureReason', 'Duration',  
      'GrossProductionLoss', 'AffectedProduction', 'IsAffectingProduction',  
      'LaborCost', 'MaterialCost', 'TotalCost', 'Assigned', 'Trade',  
      'TradeGroup', 'SupervisorRole', 'Manufacturer', 'Model', 'Safety',  
      'Reopened', 'CreatedBy', 'UpdatedBy', 'CreatedDate', 'UpdatedDate',  
      'ReportDate', 'ReportMonth', 'ReportWeekDay', 'TargetDate',  
      'SchEndDate', 'ActualDuration', 'ScheduleCompliant', 'CompletedDate',  
      'CompletedMonth', 'CompletedWeekDay', 'TBF', 'TBF_Equipment'],  
      dtype='object')
```

```
[ ]: ## The Following Variables where deleted ##  
print(del_cols)
```

```
['WOType', 'JobType', 'PMOverdue']
```

```
[ ]: ## In this case we deleted the following columns because they don't bring any  
      ↳meaningfull information ##  
columns2del =  
↳['Comment', 'StatusDescription', 'StatusShortDescription', 'CreatedBy', 'UpdatedBy', 'CreatedDate',  
columns2del2 =  
↳['LaborCost', 'EquipmentDescription', 'EquipmentRollupCode', 'CompletedMonth']
```

```
[ ]: ## In this case we deleted the following columns because they don't bring any  
      ↳meaningfull information ##  
  
for i in range(0, len(column_names)):  
    print(column_names[i] + " " + str(i) + " " + str(df_uniques[i]))  
    if df_uniques[i] == 1:  
        del_cols.append(column_names[i])  
        df = df.drop([column_names[i]], axis=1)  
        print(column_names[i] + " Se elimino la columna")
```

```
WorkOrder 0 89480  
FieldProductionTeam 1 56  
Comment 2 87299  
EquipmentCode 3 45782  
EquipmentRollupCode 4 45586  
EquipmentDescription 5 43328  
EquipmentType 6 4
```

EquipmentClass 7 332
 EquipmentCriticality 8 8
 StatusCode 9 2
 StatusDescription 10 2
 StatusShortDescription 11 2
 Priority 12 6
 Action 13 1
 Action Se elimino la columna
 Requirement 14 1
 Requirement Se elimino la columna
 Cause 15 35
 FailureReason 16 21
 Duration 17 50
 GrossProductionLoss 18 580
 AffectedProduction 19 169
 IsAffectingProduction 20 463
 LaborCost 21 392
 MaterialCost 22 2
 TotalCost 23 1357
 Assigned 24 1346
 Trade 25 4059
 TradeGroup 26 1140
 SupervisorRole 27 78
 Manufacturer 28 4
 Model 29 4
 Safety 30 914
 Reopened 31 5366
 CreatedBy 32 2
 UpdatedBy 33 2
 CreatedDate 34 1726
 UpdatedDate 35 329
 ReportDate 36 1590
 ReportMonth 37 1021
 ReportWeekDay 38 1594
 TargetDate 39 12
 SchEndDate 40 7
 ActualDuration 41 1467
 ScheduleCompliant 42 1466
 CompletedDate 43 280
 CompletedMonth 44 2
 CompletedWeekDay 45 1
 CompletedWeekDay Se elimino la columna
 TBF 46 1235
 TBF_Equipment 47 12

```
[ ]: for i in range(0,len(columns2del)):
      del_cols.append(columns2del[i])
```

```

df = df.drop([columns2del[i]],axis=1)
print(columns2del[i] + " Se elimino la columna")

for i in range(0,len(columns2del2)):
    del_cols.append(columns2del2[i])
    df = df.drop([columns2del2[i]],axis=1)
    print(columns2del2[i] + " Se elimino la columna")

```

```

Comment Se elimino la columna
StatusDescription Se elimino la columna
StatusShortDescription Se elimino la columna
CreatedBy Se elimino la columna
UpdatedBy Se elimino la columna
CreatedDate Se elimino la columna
UpdatedDate Se elimino la columna
ReportDate Se elimino la columna
TargetDate Se elimino la columna
SchEndDate Se elimino la columna
CompletedDate Se elimino la columna
LaborCost Se elimino la columna
EquipmentDescription Se elimino la columna
EquipmentRollupCode Se elimino la columna
CompletedMonth Se elimino la columna

```

```
[ ]: print(del_cols)
```

```

['WOType', 'JobType', 'PMOverdue', 'Action', 'Requirement', 'CompletedWeekDay',
'Comment', 'StatusDescription', 'StatusShortDescription', 'CreatedBy',
'UpdatedBy', 'CreatedDate', 'UpdatedDate', 'ReportDate', 'TargetDate',
'SchEndDate', 'CompletedDate', 'LaborCost', 'EquipmentDescription',
'EquipmentRollupCode', 'CompletedMonth']

```

```
[ ]: column_names = df.columns
print(column_names)
```

```

Index(['WorkOrder', 'FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
      'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
      'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
      'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
      'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
      'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
      'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
      'TBF_Equipment'],
      dtype='object')

```

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 89480 entries, 0 to 89479

Data columns (total 30 columns):

#	Column	Non-Null Count	Dtype
0	WorkOrder	89480 non-null	int64
1	FieldProductionTeam	89480 non-null	object
2	EquipmentCode	89480 non-null	object
3	EquipmentType	89480 non-null	object
4	EquipmentClass	89480 non-null	object
5	EquipmentCriticality	89480 non-null	object
6	StatusCode	89480 non-null	object
7	Priority	89480 non-null	int64
8	Cause	89480 non-null	object
9	FailureReason	89480 non-null	object
10	Duration	89480 non-null	int64
11	GrossProductionLoss	89480 non-null	int64
12	AffectedProduction	89480 non-null	int64
13	IsAffectingProduction	89480 non-null	object
14	MaterialCost	89480 non-null	int64
15	TotalCost	89480 non-null	int64
16	Assigned	89480 non-null	object
17	Trade	89480 non-null	object
18	TradeGroup	89480 non-null	object
19	SupervisorRole	89480 non-null	object
20	Manufacturer	89480 non-null	object
21	Model	89480 non-null	object
22	Safety	89480 non-null	object
23	Reopened	89480 non-null	object
24	ReportMonth	89480 non-null	int64
25	ReportWeekDay	89480 non-null	int64
26	ActualDuration	89480 non-null	int64
27	ScheduleCompliant	89480 non-null	object
28	TBF	89480 non-null	int64
29	TBF_Equipment	43698 non-null	float64

dtypes: float64(1), int64(11), object(18)

memory usage: 20.5+ MB

```
[ ]: ## Graph Maker ##
      ## Equipment Class ## Histograms

def createGraphs(df):
    EquipmentClass = df["EquipmentClass"]
    plt.figure(figsize=(30,15))
    plt.title("Histogram of Equipment",**fontT)
    plt.xlabel("Number of times present in the data set",**fontL)
    plt.ylabel("Equipment Class",**fontL)
    plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
```



```
plt.yticks(fontsize = 12 , family = "Arial")
plt.gca().spines['top'].set_visible(False)
plt.hist(EquipmentClass,bins='auto', color="g")
```

```
[ ]: ## Function to standarize the data set ##
```

```
def EncodeData(df):
    column_names = df.columns
    data_types = df.dtypes
    data_types = data_types.values.tolist()
    le= LabelEncoder()
    for i in range(0,len(column_names)):
        name = column_names[i]
        if data_types[i] == "object":
            df[name]= le.fit_transform(df[name])
    df.fillna(df.mean())
    df = df.set_index('WorkOrder')
    return(df)
```

```
[ ]: def getCommonValues(df,n):
    Equipment_Class = df["EquipmentClass"].unique()

    common_values = []

    Count = Counter(df["EquipmentClass"])
    for i in range(0,len(Equipment_Class)):
        Equipclass = Equipment_Class[i]
        times= Count[Equipclass]
        if times > n:
            common_values.append(Equipclass)
    return(common_values)
```

```
[ ]: ## In this case we will separate the dataset by the type of equipment ##
```

```
Equipment_Types = df["EquipmentType"].unique()
Equipment_Types
```

```
[ ]: array(['System', 'Asset', 'Position', 'Location'], dtype=object)
```

```
[ ]: ## System Dataframe ##
dfSystem = df.loc[df["EquipmentType"] == 'System']
dfSystem.head()
```

```
[ ]:   WorkOrder FieldProductionTeam EquipmentCode EquipmentType \
0    15516284          RANGELY      C35CS03      System
1    15506143           VAN      CQ3253FL1      System
3    15476785           VAN           VAN      System
```

4	15477769	VAN	VAN	System
5	15463161	VAN	VAN	System

	EquipmentClass	EquipmentCriticality	StatusCode	Priority	\
0	SE - Battery	NA - Not Applicable	C	2	
1	Line - Production Flowline	3 - Low	C	4	
3	SE - Field	NA - Not Applicable	C	3	
4	SE - Field	NA - Not Applicable	C	3	
5	SE - Field	NA - Not Applicable	C	3	

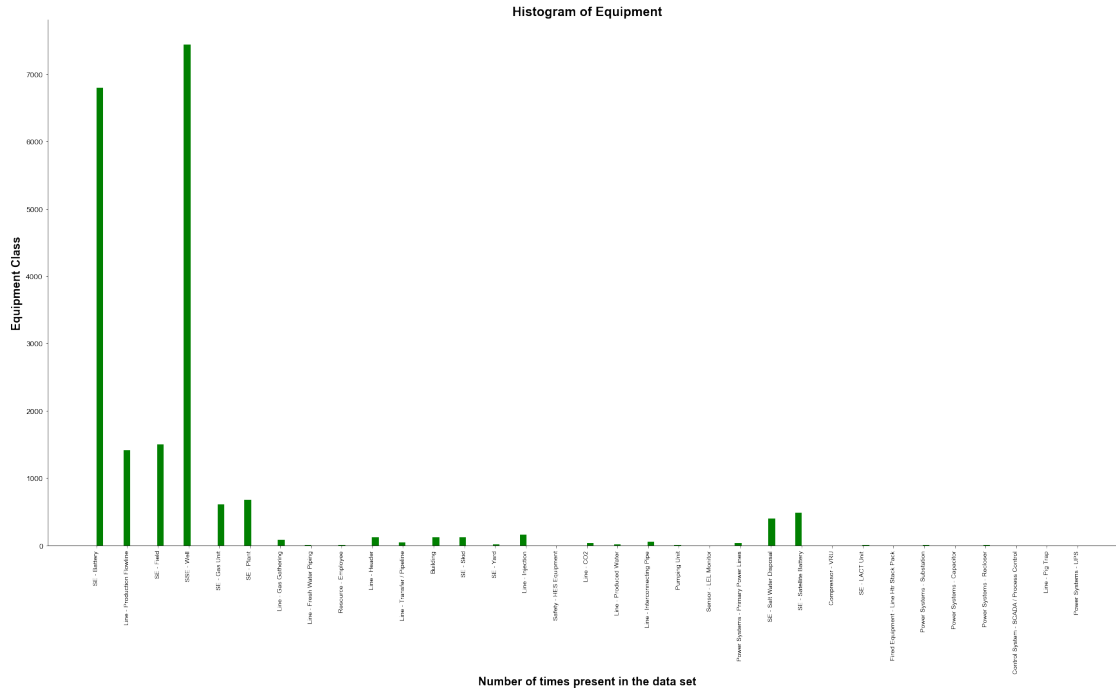
	Cause	FailureReason	...	Manufacturer	Model	Safety	\
0	Plugged	Pipe - Piping	...	NOTAPPLICABLE	NONE	N	
1	Missing	Assembly - Pump, Chemical	...	NOTAPPLICABLE	NONE	N	
3	Improper Fit	Pipe - Piping	...	NOTAPPLICABLE	NONE	N	
4	Fouling	Pump, other	...	NOTAPPLICABLE	NONE	N	
5	Distorted	Pump, other	...	NOTAPPLICABLE	NONE	N	

	Reopened	ReportMonth	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	\
0	N	12	4	-1	Future Week	0	
1	N	12	2	3	Future Week	0	
3	N	12	6	19	Future Week	0	
4	N	12	7	19	Future Week	0	
5	N	12	3	29	Future Week	0	

	TBF_Equipment
0	4.0
1	NaN
3	0.0
4	0.0
5	0.0

[5 rows x 30 columns]

```
[ ]: createGraphs(dfSystem)
```



```
[ ]: dfSystemCommonValues = getCommonValues(dfSystem,200)
print(dfSystemCommonValues)
```

```
['SE - Battery', 'Line - Production Flowline', 'SE - Field', 'SSE - Well', 'SE - Gas Unit', 'SE - Plant', 'SE - Salt Water Disposal', 'SE - Satellite Battery']
```

```
[ ]: dfSystem = dfSystem.loc[dfSystem["EquipmentClass"].isin(dfSystemCommonValues)]
dfSystem.head()
```

```
[ ]: WorkOrder FieldProductionTeam EquipmentCode EquipmentType \
0 15516284 RANGELY C35CS03 System
1 15506143 VAN CQ3253FL1 System
3 15476785 VAN VAN System
4 15477769 VAN VAN System
5 15463161 VAN VAN System
```

```
EquipmentClass EquipmentCriticality StatusCode Priority \
0 SE - Battery NA - Not Applicable C 2
1 Line - Production Flowline 3 - Low C 4
3 SE - Field NA - Not Applicable C 3
4 SE - Field NA - Not Applicable C 3
5 SE - Field NA - Not Applicable C 3
```

```
Cause FailureReason ... Manufacturer Model Safety \
0 Plugged Pipe - Piping ... NOTAPPLICABLE NONE N
```

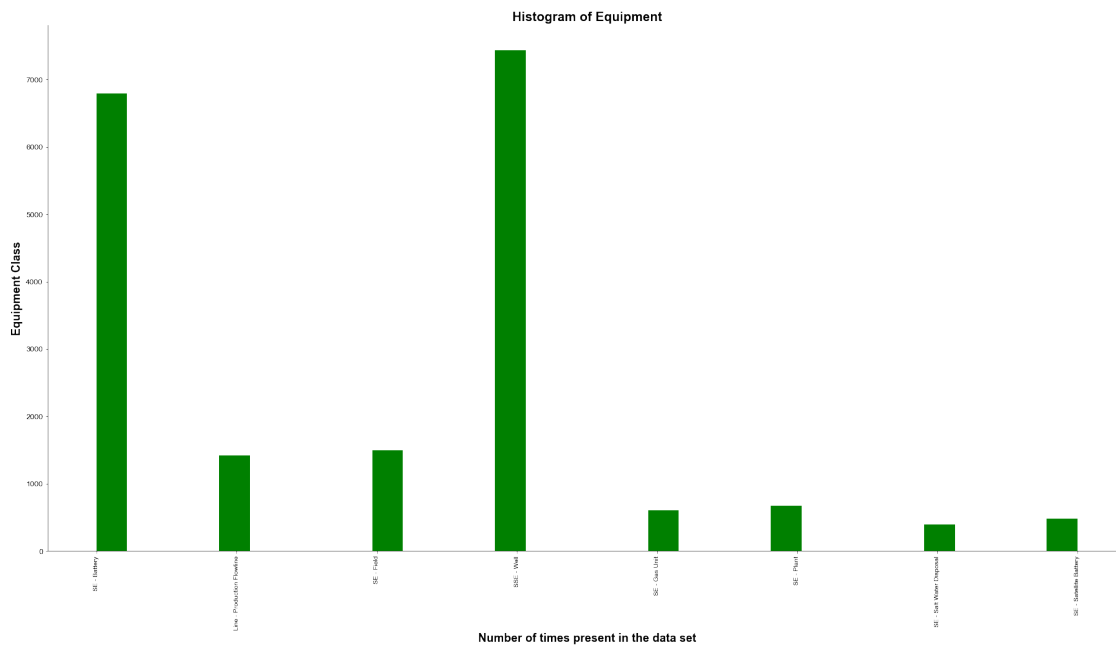
1	Missing	Assembly - Pump, Chemical	...	NOTAPPLICABLE	NONE	N
3	Improper Fit	Pipe - Piping	...	NOTAPPLICABLE	NONE	N
4	Fouling	Pump, other	...	NOTAPPLICABLE	NONE	N
5	Distorted	Pump, other	...	NOTAPPLICABLE	NONE	N

	Reopened	ReportMonth	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF \
0	N	12	4	-1	Future Week	0
1	N	12	2	3	Future Week	0
3	N	12	6	19	Future Week	0
4	N	12	7	19	Future Week	0
5	N	12	3	29	Future Week	0

	TBF_Equipment
0	4.0
1	NaN
3	0.0
4	0.0
5	0.0

[5 rows x 30 columns]

```
[ ]: createGraphs(dfSystem)
```



```
[ ]: ## Now that the data has been narrowed to the most common case we will encode
      ↳ it make some changes to it ##
dfSystem= EncodeData(dfSystem)
```

```
dfSystem.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
WorkOrder
15516284                23            1113                0                1
15506143                26            1269                0                0
15476785                26            6873                0                2
15477769                26            6873                0                2
15463161                26            6873                0                2
```

```
      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason  \
WorkOrder
15516284                2            0         2    36          153
15506143                1            0         4    25           7
15476785                2            0         3    19          153
15477769                2            0         3    13          162
15463161                2            0         3     8          162
```

```
      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth  \
WorkOrder      ...
15516284        1  ...           5    564        0          0          12
15506143        1  ...           5    564        0          0          12
15476785        1  ...           5    564        0          0          12
15477769        1  ...           5    564        0          0          12
15463161        1  ...           5    564        0          0          12
```

```
      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  \
WorkOrder
15516284           4             -1                   0    0
15506143           2              3                   0    0
15476785           6             19                   0    0
15477769           7             19                   0    0
15463161           3             29                   0    0
```

```
      TBF_Equipment
WorkOrder
15516284        4.0
15506143        NaN
15476785        0.0
15477769        0.0
15463161        0.0
```

```
[5 rows x 29 columns]
```

```
[ ]: ## We need to handle the datasets NA Values ##
dfSystem = dfSystem.fillna(dfSystem.mean())
dfSystem.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
WorkOrder
15516284      23      1113      0      1
15506143      26      1269      0      0
15476785      26      6873      0      2
15477769      26      6873      0      2
15463161      26      6873      0      2
```

```
      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason  \
WorkOrder
15516284      2      0      2      36      153
15506143      1      0      4      25      7
15476785      2      0      3      19      153
15477769      2      0      3      13      162
15463161      2      0      3      8      162
```

```
      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth  \
WorkOrder      ...
15516284      1  ...      5      564      0      0      12
15506143      1  ...      5      564      0      0      12
15476785      1  ...      5      564      0      0      12
15477769      1  ...      5      564      0      0      12
15463161      1  ...      5      564      0      0      12
```

```
      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  \
WorkOrder
15516284      4      -1      0      0
15506143      2      3      0      0
15476785      6      19      0      0
15477769      7      19      0      0
15463161      3      29      0      0
```

```
      TBF_Equipment
WorkOrder
15516284      4.000000
15506143      122.711546
15476785      0.000000
15477769      0.000000
15463161      0.000000
```

[5 rows x 29 columns]

```
[ ]:  ## For further analysis we know need to standarize the dataset##
      ## Scales ##
      scaling_procedure_1 = MinMaxScaler(feature_range= (0,1))
```

```
[ ]: ## Scaled Data ##
columnSystem = dfSystem.columns
dfSystem_scaled = scaling_procedure_1.fit_transform(dfSystem)
dfSystem_scaled = pd.DataFrame(dfSystem_scaled, columns = columnSystem)
dfSystem_scaled.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
0          0.766667      0.161445          0.0          0.142857
1          0.866667      0.184073          0.0          0.000000
2          0.866667      0.996954          0.0          0.285714
3          0.866667      0.996954          0.0          0.285714
4          0.866667      0.996954          0.0          0.285714

      EquipmentCriticality  StatusCode  Priority    Cause  FailureReason  \
0              0.4          0.0      0.4  0.765957      0.742718
1              0.2          0.0      0.8  0.531915      0.033981
2              0.4          0.0      0.6  0.404255      0.742718
3              0.4          0.0      0.6  0.276596      0.786408
4              0.4          0.0      0.6  0.170213      0.786408

      Duration  ...  Manufacturer      Model  Safety  Reopened  ReportMonth  \
0          0.0  ...      0.714286  0.996466      0.0          0.0          1.0
1          0.0  ...      0.714286  0.996466      0.0          0.0          1.0
2          0.0  ...      0.714286  0.996466      0.0          0.0          1.0
3          0.0  ...      0.714286  0.996466      0.0          0.0          1.0
4          0.0  ...      0.714286  0.996466      0.0          0.0          1.0

      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  TBF_Equipment
0          0.500000          0.002308                0.0  0.0          0.003223
1          0.166667          0.005385                0.0  0.0          0.098881
2          0.833333          0.017692                0.0  0.0          0.000000
3          1.000000          0.017692                0.0  0.0          0.000000
4          0.333333          0.025385                0.0  0.0          0.000000
```

[5 rows x 29 columns]

```
[ ]: ##Here is the correlation plot of the system##

corr_df = dfSystem_scaled

corr = dfSystem_scaled.corr()
corr.style.background_gradient(cmap="coolwarm")

plt.figure(figsize=(70, 10))
heatmap = sns.heatmap(corr, vmin=-1, vmax=1, annot=True)
```



```
[ ]: ## PCA ##
#Only O3 Dataframe

Xdat = dfSystem_scaled

header_row= dfSystem_scaled.columns.values

y=dfSystem_scaled["TBF"]
```

```
[ ]: # create the PCA instance
pca = PCA()

# fit on data
pca = pca.fit(Xdat)

#Q eigenvectors
Q=pca.components_.T

#Explained deviations
s= np.sqrt(pca.explained_variance_)
```

```
[ ]: # transform data obtain sample scores
F = pca.transform(Xdat)
print("Sample scores")

## Sample Scores ##
sample_scores = pd.DataFrame(F)
print(sample_scores)
```

```
Sample scores
          0          1          2          3          4          5          6  \
0   -0.579877  0.077823 -0.027538  0.051255  0.417339  0.044362  0.222892
1   -0.499227 -0.200578  0.156723 -0.106649  0.417238  0.703406 -0.036615
2   -0.527049 -0.032093  0.140532 -0.070417 -0.221779 -0.448005  0.322284
3   -0.535728 -0.018801  0.127992 -0.053992 -0.201225 -0.525898  0.422260
4   -0.516279 -0.025694  0.151367 -0.067108 -0.236529 -0.423622 -0.009162
...      ...      ...      ...      ...      ...      ...      ...
19300  0.624775  0.630449 -0.299916 -0.041822 -0.014483 -0.126359  0.129573
19301 -0.135414  0.175100 -0.770830  0.224554  0.164545 -0.294017 -0.117421
```



```

19302  0.001771 -0.182979 -0.457570 -0.018899 -0.356210  0.788594  0.442213
19303  0.635626  0.467988 -0.118068 -0.273918 -0.345894  0.593446  0.341009
19304  0.147060 -0.657255  0.271244  0.239615  0.394199 -0.179411  0.184281

```

```

          7          8          9  ...          19          20          21  \
0      -0.251686  1.098944  0.264984  ... -0.010781 -0.012197  0.000835
1      -0.295610  0.979616  0.545309  ...  0.001547 -0.011683  0.001846
2       0.072080  0.866731  0.457996  ...  0.001869 -0.009787 -0.002930
3       0.183406  0.805905  0.562950  ...  0.001823 -0.010082 -0.003107
4      -0.312710  0.730799  0.626446  ...  0.002906 -0.009498 -0.000369
...
19300  0.543490 -0.177609  0.198381  ...  0.004834 -0.001383 -0.008006
19301 -0.207806 -0.364319  0.158101  ... -0.004523  0.003368 -0.005214
19302 -0.106015 -0.163111 -0.106599  ...  0.016423  0.001322  0.007832
19303  0.165086 -0.097062  0.115100  ...  0.013647 -0.000015  0.009202
19304  0.280363 -0.008290 -0.265507  ... -0.004980 -0.001786 -0.005056

```

```

          22          23          24          25          26          27  \
0      -0.032691 -0.005139 -0.003369 -0.003132 -0.000526  0.004037
1      -0.031981 -0.004915  0.002049 -0.005904  0.000085 -0.000918
2      -0.019456 -0.004966 -0.000488  0.001762 -0.000357 -0.001469
3      -0.019435 -0.004985 -0.000762  0.001834 -0.000363 -0.001372
4      -0.012124 -0.004166 -0.000961  0.001778 -0.000484 -0.001069
...
19300  0.000762  0.000480 -0.005905  0.000527 -0.000850 -0.000750
19301 -0.001405  0.000416  0.001310  0.000394  0.000687 -0.000124
19302  0.000513  0.001697 -0.004452  0.000116  0.000556 -0.001324
19303  0.001382  0.001500  0.060111  0.000476 -0.008543  0.000639
19304 -0.002184 -0.000320  0.001068 -0.002740  0.001056 -0.001384

```

```

          28
0      -2.969047e-17
1       2.126569e-17
2      -1.202613e-16
3      -1.310632e-16
4      -1.144945e-16
...
19300 -1.175632e-16
19301 -1.960125e-16
19302  2.249297e-17
19303 -8.937506e-17
19304 -1.541705e-16

```

```
[19305 rows x 29 columns]
```

```
[ ]: # Obtain Factor Loadings
L = np.zeros(shape=(Q.shape))
```

```

for col in range(0,len(s)-1):
    L[:,col] = Q[:,col] * s[col]
print("Factor Loadings")

columns_factorLoading = []

for i in range(0, 29):
    nombre = "F" + str(i)
    columns_factorLoading.append(nombre)

## Factor Loadings Dataframe ##
factor_loadings = pd.DataFrame(L, columns= columns_factorLoading)
print(factor_loadings)

```

Factor Loadings

	F0	F1	F2	F3	F4 \
0	-2.251498e-02	-1.822834e-02	-1.847917e-02	2.751824e-02	3.203609e-02
1	-6.469392e-02	2.646159e-03	1.439105e-01	-9.347111e-02	-2.737765e-01
2	-5.326034e-17	-7.130236e-17	-6.613795e-17	1.066881e-17	4.442305e-17
3	2.452924e-01	-3.764365e-02	-2.987261e-01	-1.958994e-02	-1.212640e-01
4	7.735359e-03	2.717889e-02	1.392210e-02	-4.568015e-03	1.293404e-02
5	3.093580e-03	5.199895e-03	-1.984475e-03	3.107925e-03	-6.443601e-03
6	-1.941060e-02	-2.715260e-02	-1.558109e-02	2.936459e-03	-8.847779e-05
7	-1.217811e-02	-3.714084e-03	4.956751e-03	-1.146907e-02	-2.133014e-02
8	-8.484577e-02	1.002694e-01	-4.340206e-02	5.875685e-02	1.174451e-02
9	-5.356499e-04	-2.515442e-04	3.799527e-05	5.481615e-04	7.826019e-04
10	1.019142e-03	1.923245e-03	1.235749e-03	-3.120494e-04	1.756797e-04
11	4.197497e-04	6.233904e-04	5.726303e-04	-1.627485e-04	-4.888043e-05
12	3.021916e-01	2.948328e-01	1.263672e-01	-6.875994e-03	1.499842e-02
13	-3.014746e-05	7.089744e-04	6.237011e-05	9.258752e-05	4.024444e-04
14	-2.300412e-04	1.063397e-03	4.515325e-04	1.922399e-04	5.873497e-04
15	-6.155523e-02	3.703977e-02	-1.823345e-02	-4.976955e-02	-2.228381e-02
16	-6.358153e-02	9.999392e-02	-4.043505e-02	3.207608e-02	6.023191e-03
17	-4.677101e-02	7.483751e-02	-4.129437e-03	2.139753e-02	-6.097016e-03
18	-2.340344e-01	2.565782e-01	-1.591097e-01	-3.949051e-02	-2.902927e-02
19	7.049841e-04	-2.104840e-04	-6.509736e-04	3.679231e-04	7.824659e-05
20	-2.444371e-02	4.919432e-03	2.471581e-02	7.032531e-04	-3.251957e-03
21	-9.414646e-03	4.764879e-03	1.146128e-02	-8.261734e-03	9.766559e-04
22	-3.651081e-03	2.198386e-03	1.064839e-03	1.716260e-03	1.269392e-03
23	2.211933e-03	-1.291756e-02	-1.243835e-02	-2.697194e-01	9.429571e-02
24	-1.206577e-02	5.016113e-03	-1.470729e-02	8.354895e-03	2.055806e-02
25	-1.168981e-03	-7.670582e-04	6.231898e-04	2.364885e-03	-1.256613e-03
26	1.741269e-02	3.105909e-04	-2.127246e-02	-1.306268e-01	5.073266e-02
27	-4.389606e-04	-3.593622e-04	7.468278e-04	-6.351963e-05	3.632964e-04
28	1.469305e-02	3.216264e-03	-1.571656e-02	2.679892e-03	-6.876319e-03
	F5	F6	F7	F8	F9 ... \

0	5.422692e-02	0.181528	-1.377373e-01	-5.776392e-03	1.840482e-02	...
1	-4.427720e-02	0.031141	-7.917832e-03	-1.774510e-02	6.226823e-03	...
2	-4.092356e-17	0.000000	1.545116e-17	2.159976e-17	1.823657e-17	...
3	-2.041176e-02	0.011998	-1.011687e-02	4.326280e-03	3.975487e-03	...
4	-2.110760e-03	0.020836	-1.166287e-02	-1.774381e-02	-8.184505e-03	...
5	6.082215e-03	0.005557	-9.335765e-04	-1.426084e-02	-4.179078e-03	...
6	-5.040399e-03	0.001180	3.946853e-03	6.597126e-03	-6.894079e-03	...
7	4.281993e-02	0.025924	-3.172827e-03	1.398709e-01	-2.070252e-01	...
8	-2.019306e-01	0.039289	-9.213110e-02	1.342159e-02	-3.055158e-02	...
9	5.446937e-04	0.000113	-6.496143e-05	6.732519e-05	-6.480820e-05	...
10	-4.321243e-05	0.000171	9.701756e-05	-2.455317e-04	2.341759e-04	...
11	3.134552e-04	-0.000052	5.383115e-05	-1.313931e-04	1.848602e-04	...
12	1.958152e-02	0.016429	-3.771784e-03	1.045516e-02	1.867841e-03	...
13	-5.070631e-04	-0.000309	8.685461e-05	-1.247962e-04	5.255921e-05	...
14	-6.986564e-04	-0.000417	4.134641e-05	-2.244787e-04	1.317148e-04	...
15	1.234632e-01	0.103553	-6.146737e-02	2.783964e-03	3.998099e-02	...
16	-7.741503e-02	0.011197	-1.781488e-02	-2.076890e-02	-1.111003e-02	...
17	-4.984026e-02	0.005235	-8.695797e-03	-1.421330e-02	6.755559e-03	...
18	8.232250e-02	-0.045949	4.388971e-02	4.928856e-03	7.972523e-03	...
19	1.174795e-04	0.000124	-1.954433e-04	-1.742899e-04	-1.044971e-04	...
20	2.315840e-03	-0.006673	5.347026e-03	-6.742719e-04	-6.180010e-03	...
21	-5.136071e-03	-0.003104	1.842802e-03	-1.696938e-02	6.434500e-03	...
22	-3.109183e-03	-0.000272	6.482029e-05	-9.239284e-04	-1.902558e-03	...
23	-7.266765e-02	0.000106	-2.057097e-02	9.357951e-02	5.189368e-02	...
24	-5.208432e-02	0.183446	2.076417e-01	7.443560e-03	8.956865e-03	...
25	-7.163324e-04	-0.000711	-7.678352e-05	6.608778e-03	4.084978e-03	...
26	-1.586277e-03	0.017786	-2.290892e-03	-1.968970e-01	-1.229241e-01	...
27	1.889279e-04	0.000087	1.336379e-04	8.866294e-04	6.915009e-04	...
28	-6.603034e-03	-0.001755	6.670278e-04	9.029430e-03	5.735615e-03	...

	F19	F20	F21	F22	F23 \
0	5.989970e-04	-4.781394e-05	1.355726e-05	2.232189e-05	2.586812e-05
1	1.964890e-04	7.105420e-06	2.589875e-05	-1.976893e-05	-3.335702e-05
2	-1.239918e-17	1.567234e-17	8.466367e-19	6.961186e-18	1.030754e-17
3	5.067670e-04	1.264916e-04	4.253936e-05	3.757222e-06	5.733165e-05
4	5.053786e-04	1.376939e-04	-6.312236e-05	-3.231021e-05	1.070789e-05
5	6.630206e-04	2.585892e-04	-3.319128e-05	-2.367016e-05	1.758321e-05
6	-9.177087e-04	2.504381e-04	1.197397e-04	-1.520519e-04	3.098997e-05
7	-3.027691e-04	3.173967e-05	4.175466e-05	4.469253e-06	-4.948983e-06
8	-6.355697e-04	-5.114141e-05	-4.242740e-05	-1.124991e-05	-2.391897e-06
9	-1.462035e-05	3.615836e-05	6.003669e-04	-1.201347e-04	7.472680e-05
10	1.612489e-04	4.194322e-04	4.503960e-03	4.795034e-05	2.429357e-04
11	-5.882460e-06	-9.090752e-08	2.416403e-04	6.103288e-05	5.303021e-07
12	6.238270e-05	1.232116e-05	-5.222469e-05	3.401198e-05	6.264274e-06
13	1.546756e-04	-4.431475e-05	1.526583e-02	-2.159572e-04	1.503442e-04
14	3.110615e-04	3.981249e-06	1.653712e-02	-2.143297e-04	1.944596e-04
15	-1.414313e-04	9.263755e-06	7.904050e-05	-5.751168e-06	-1.929458e-06
16	-9.603926e-04	2.843438e-04	4.540222e-06	-9.089379e-06	-5.689263e-06

17	-1.709217e-03	3.185911e-05	-7.440664e-05	5.651868e-06	1.001140e-05
18	2.435551e-04	-3.019062e-05	-3.065149e-05	1.591812e-05	-1.379438e-05
19	6.010607e-06	-1.281963e-04	3.497653e-04	1.562923e-03	-2.025345e-02
20	7.330210e-04	8.775895e-05	-6.188366e-05	1.066054e-05	2.061173e-04
21	3.401489e-04	7.381635e-05	4.019639e-05	-2.369101e-05	-1.876703e-05
22	8.499082e-02	-2.514584e-04	-9.616817e-05	1.958543e-05	1.783054e-06
23	2.810067e-04	-7.016424e-05	-2.101350e-05	-1.265751e-04	-2.659648e-05
24	-8.906268e-05	-4.700307e-05	2.071835e-06	9.166930e-06	-5.673910e-06
25	-6.894368e-05	1.172269e-05	2.930947e-04	2.099292e-02	1.509550e-03
26	-3.370002e-04	1.324525e-04	2.672293e-06	6.512529e-04	5.189149e-05
27	4.639072e-04	4.792204e-02	-2.714026e-05	-2.267841e-06	-5.776908e-05
28	3.033360e-03	-2.256349e-04	1.160456e-05	-3.011285e-06	-5.152803e-05

	F24	F25	F26	F27	F28
0	3.409243e-06	-2.381041e-05	4.085344e-08	4.381972e-07	0.0
1	-1.530139e-05	3.159297e-05	-3.376767e-06	-7.308223e-08	0.0
2	1.083945e-17	-2.343904e-17	-4.988578e-19	1.107087e-18	0.0
3	1.763078e-05	1.940900e-05	4.589512e-06	2.120103e-06	0.0
4	-3.764411e-05	3.953143e-05	1.346077e-05	-2.875330e-06	0.0
5	-1.624418e-05	8.884324e-06	-1.254796e-07	-1.290252e-06	0.0
6	1.571047e-04	-6.909250e-05	4.337975e-05	5.903768e-07	0.0
7	1.263300e-05	-3.601280e-06	4.154771e-06	1.003456e-06	0.0
8	-4.149437e-06	6.601666e-06	5.082579e-06	-3.962272e-07	0.0
9	-2.271306e-04	1.217566e-02	-5.417792e-05	-4.652242e-05	0.0
10	1.431812e-02	1.085567e-04	-1.176391e-03	-1.507658e-06	0.0
11	2.090474e-03	9.986042e-05	8.251985e-03	-4.136668e-05	0.0
12	-6.386385e-05	1.921289e-06	-8.890401e-06	-6.600490e-07	0.0
13	-1.980034e-03	-3.702335e-04	2.201441e-05	-2.308260e-03	0.0
14	-2.097177e-03	-1.330192e-04	1.818293e-04	2.133425e-03	0.0
15	-8.504526e-06	-1.224099e-06	-1.695630e-06	6.273051e-07	0.0
16	2.153037e-05	3.990086e-05	3.415584e-06	-1.229668e-06	0.0
17	-2.041815e-06	-4.258584e-05	6.987154e-06	-1.881083e-06	0.0
18	-6.558474e-06	-5.333211e-06	-3.153896e-06	2.178441e-07	0.0
19	1.299149e-04	4.632492e-05	-1.349770e-05	2.983922e-06	0.0
20	-6.565403e-06	-2.082306e-05	-9.270032e-07	-9.653554e-07	0.0
21	2.107157e-05	1.079719e-05	1.603172e-06	4.646227e-07	0.0
22	-1.328190e-05	1.423367e-06	2.499596e-06	-3.637393e-06	0.0
23	-5.792122e-06	1.235541e-05	1.087316e-06	-1.010329e-07	0.0
24	-1.141861e-05	8.609729e-07	1.368522e-06	5.505282e-07	0.0
25	-9.008904e-05	6.026513e-05	-1.811650e-05	-2.335242e-06	0.0
26	-9.562820e-06	3.055561e-07	-1.734142e-06	2.602836e-07	0.0
27	-1.272678e-04	-1.034132e-05	1.006523e-05	-2.257594e-06	0.0
28	7.742133e-06	1.131573e-05	3.547363e-06	2.351989e-07	0.0

[29 rows x 29 columns]

```
[ ]: #Obtain squared cosines
COS2=L**2
print("Square Cosines")

## COS2 Dataframe ##
COS2_DF = pd.DataFrame(COS2)
print(COS2_DF)

## Create a table ##
```

Square Cosines

	0	1	2	3	4	\
0	5.069243e-04	3.322725e-04	3.414798e-04	7.572535e-04	1.026311e-03	
1	4.185303e-03	7.002160e-06	2.071024e-02	8.736848e-03	7.495357e-02	
2	2.836664e-33	5.084027e-33	4.374229e-33	1.138235e-34	1.973407e-33	
3	6.016836e-02	1.417045e-03	8.923728e-02	3.837657e-04	1.470496e-02	
4	5.983578e-05	7.386921e-04	1.938250e-04	2.086676e-05	1.672893e-04	
5	9.570235e-06	2.703891e-05	3.938142e-06	9.659195e-06	4.152000e-05	
6	3.767713e-04	7.372636e-04	2.427705e-04	8.622794e-06	7.828320e-09	
7	1.483065e-04	1.379442e-05	2.456938e-05	1.315396e-04	4.549747e-04	
8	7.198805e-03	1.005395e-02	1.883739e-03	3.452368e-03	1.379336e-04	
9	2.869208e-07	6.327446e-08	1.443641e-09	3.004810e-07	6.124657e-07	
10	1.038650e-06	3.698872e-06	1.527076e-06	9.737484e-08	3.086334e-08	
11	1.761898e-07	3.886156e-07	3.279054e-07	2.648708e-08	2.389297e-09	
12	9.131975e-02	8.692637e-02	1.596867e-02	4.727929e-05	2.249525e-04	
13	9.088694e-10	5.026447e-07	3.890030e-09	8.572448e-09	1.619615e-07	
14	5.291897e-08	1.130814e-06	2.038816e-07	3.695616e-08	3.449797e-07	
15	3.789047e-03	1.371944e-03	3.324588e-04	2.477008e-03	4.965681e-04	
16	4.042611e-03	9.998784e-03	1.634993e-03	1.028875e-03	3.627883e-05	
17	2.187527e-03	5.600652e-03	1.705225e-05	4.578543e-04	3.717360e-05	
18	5.477211e-02	6.583235e-02	2.531591e-02	1.559501e-03	8.426984e-04	
19	4.970026e-07	4.430350e-08	4.237667e-07	1.353674e-07	6.122528e-09	
20	5.974949e-04	2.420081e-05	6.108710e-04	4.945649e-07	1.057522e-05	
21	8.863557e-05	2.270407e-05	1.313610e-04	6.825624e-05	9.538567e-07	
22	1.333039e-05	4.832903e-06	1.133882e-06	2.945548e-06	1.611357e-06	
23	4.892646e-06	1.668633e-04	1.547126e-04	7.274856e-02	8.891682e-03	
24	1.455829e-04	2.516139e-05	2.163043e-04	6.980427e-05	4.226337e-04	
25	1.366516e-06	5.883783e-07	3.883655e-07	5.592682e-06	1.579077e-06	
26	3.032017e-04	9.646668e-08	4.525177e-04	1.706336e-02	2.573803e-03	
27	1.926864e-07	1.291412e-07	5.577518e-07	4.034744e-09	1.319843e-07	
28	2.158858e-04	1.034436e-05	2.470104e-04	7.181823e-06	4.728376e-05	
	5	6	7	8	9	...
0	2.940559e-03	3.295240e-02	1.897158e-02	3.336671e-05	3.387376e-04	...
1	1.960470e-03	9.697309e-04	6.269206e-05	3.148886e-04	3.877333e-05	...
2	1.674738e-33	0.000000e+00	2.387385e-34	4.665498e-34	3.325724e-34	...
3	4.166398e-04	1.439467e-04	1.023511e-04	1.871670e-05	1.580450e-05	...
4	4.455309e-06	4.341399e-04	1.360226e-04	3.148429e-04	6.698612e-05	...

5	3.699334e-05	3.088495e-05	8.715651e-07	2.033715e-04	1.746470e-05	...
6	2.540562e-05	1.391677e-06	1.557765e-05	4.352207e-05	4.752833e-05	...
7	1.833547e-03	6.720589e-04	1.006683e-05	1.956387e-02	4.285943e-02	...
8	4.077598e-02	1.543648e-03	8.488140e-03	1.801391e-04	9.333992e-04	...
9	2.966912e-07	1.283411e-08	4.219988e-09	4.532681e-09	4.200103e-09	...
10	1.867314e-09	2.918788e-08	9.412407e-09	6.028579e-08	5.483837e-08	...
11	9.825419e-08	2.724811e-09	2.897792e-09	1.726415e-08	3.417330e-08	...
12	3.834360e-04	2.699102e-04	1.422635e-05	1.093105e-04	3.488831e-06	...
13	2.571130e-07	9.531628e-08	7.543724e-09	1.557409e-08	2.762470e-09	...
14	4.881208e-07	1.738524e-07	1.709526e-09	5.039068e-08	1.734880e-08	...
15	1.524316e-02	1.072330e-02	3.778237e-03	7.750457e-06	1.598479e-03	...
16	5.993086e-03	1.253663e-04	3.173701e-04	4.313472e-04	1.234327e-04	...
17	2.484052e-03	2.740212e-05	7.561689e-05	2.020180e-04	4.563757e-05	...
18	6.776994e-03	2.111312e-03	1.926307e-03	2.429362e-05	6.356113e-05	...
19	1.380144e-08	1.525823e-08	3.819810e-08	3.037696e-08	1.091964e-08	...
20	5.363115e-06	4.453381e-05	2.859069e-05	4.546426e-07	3.819252e-05	...
21	2.637923e-05	9.635511e-06	3.395919e-06	2.879600e-04	4.140279e-05	...
22	9.667019e-06	7.387091e-08	4.201670e-09	8.536437e-07	3.619727e-06	...
23	5.280587e-03	1.126831e-08	4.231650e-04	8.757124e-03	2.692954e-03	...
24	2.712777e-03	3.365236e-02	4.311507e-02	5.540658e-05	8.022543e-05	...
25	5.131320e-07	5.054483e-07	5.895709e-09	4.367595e-05	1.668704e-05	...
26	2.516273e-06	3.163417e-04	5.248186e-06	3.876844e-02	1.511033e-02	...
27	3.569374e-08	7.634742e-09	1.785909e-08	7.861117e-07	4.781735e-07	...
28	4.360006e-05	3.078921e-06	4.449261e-07	8.153061e-05	3.289728e-05	...

	19	20	21	22	23	\
0	3.587974e-07	2.286173e-09	1.837994e-10	4.982670e-10	6.691597e-10	
1	3.860792e-08	5.048700e-11	6.707453e-10	3.908107e-10	1.112691e-09	
2	1.537397e-34	2.456221e-34	7.167937e-37	4.845811e-35	1.062454e-34	
3	2.568128e-07	1.600013e-08	1.809597e-09	1.411671e-11	3.286918e-09	
4	2.554075e-07	1.895961e-08	3.984433e-09	1.043950e-09	1.146590e-10	
5	4.395964e-07	6.686836e-08	1.101661e-09	5.602763e-10	3.091691e-10	
6	8.421893e-07	6.271923e-08	1.433760e-08	2.311977e-08	9.603779e-10	
7	9.166915e-08	1.007407e-09	1.743452e-09	1.997422e-11	2.449243e-11	
8	4.039488e-07	2.615444e-09	1.800084e-09	1.265606e-10	5.721171e-12	
9	2.137547e-10	1.307427e-09	3.604404e-07	1.443235e-08	5.584095e-09	
10	2.600122e-08	1.759234e-07	2.028565e-05	2.299235e-09	5.901774e-08	
11	3.460333e-11	8.264178e-15	5.839003e-08	3.725013e-09	2.812203e-13	
12	3.891601e-09	1.518109e-10	2.727418e-09	1.156815e-09	3.924113e-11	
13	2.392455e-08	1.963797e-09	2.330456e-04	4.663750e-08	2.260337e-08	
14	9.675923e-08	1.585034e-11	2.734764e-04	4.593721e-08	3.781452e-08	
15	2.000282e-08	8.581716e-11	6.247400e-09	3.307593e-11	3.722809e-12	
16	9.223539e-07	8.085137e-08	2.061362e-11	8.261680e-11	3.236771e-11	
17	2.921422e-06	1.015003e-09	5.536349e-09	3.194361e-11	1.002282e-10	
18	5.931910e-08	9.114735e-10	9.395137e-10	2.533866e-10	1.902849e-10	
19	3.612739e-11	1.643428e-08	1.223357e-07	2.442727e-06	4.102024e-04	
20	5.373197e-07	7.701633e-09	3.829587e-09	1.136470e-10	4.248434e-08	
21	1.157013e-07	5.448854e-09	1.615749e-09	5.612639e-10	3.522013e-10	

22	7.223439e-03	6.323132e-08	9.248317e-09	3.835889e-10	3.179280e-12
23	7.896474e-08	4.923021e-09	4.415670e-10	1.602125e-08	7.073727e-10
24	7.932160e-09	2.209288e-09	4.292500e-12	8.403260e-11	3.219326e-11
25	4.753232e-09	1.374214e-10	8.590450e-08	4.407028e-04	2.278740e-06
26	1.135691e-07	1.754367e-08	7.141149e-12	4.241304e-07	2.692727e-09
27	2.152099e-07	2.296521e-03	7.365937e-10	5.143101e-12	3.337266e-09
28	9.201274e-06	5.091112e-08	1.346659e-10	9.067836e-12	2.655138e-09

	24	25	26	27	28
0	1.162294e-11	5.669358e-10	1.669004e-15	1.920167e-13	0.0
1	2.341324e-10	9.981160e-10	1.140256e-11	5.341012e-15	0.0
2	1.174937e-34	5.493885e-34	2.488591e-37	1.225642e-36	0.0
3	3.108443e-10	3.767093e-10	2.106362e-11	4.494838e-12	0.0
4	1.417079e-09	1.562734e-09	1.811923e-10	8.267520e-12	0.0
5	2.638732e-10	7.893121e-11	1.574514e-14	1.664749e-12	0.0
6	2.468190e-08	4.773774e-09	1.881803e-09	3.485447e-13	0.0
7	1.595928e-10	1.296922e-11	1.726212e-11	1.006924e-12	0.0
8	1.721783e-11	4.358200e-11	2.583261e-11	1.569960e-13	0.0
9	5.158830e-08	1.482466e-04	2.935247e-09	2.164336e-09	0.0
10	2.050086e-04	1.178457e-08	1.383896e-06	2.273034e-12	0.0
11	4.370083e-06	9.972104e-09	6.809525e-05	1.711203e-09	0.0
12	4.078592e-09	3.691350e-12	7.903924e-11	4.356646e-13	0.0
13	3.920536e-06	1.370728e-07	4.846343e-10	5.328066e-06	0.0
14	4.398152e-06	1.769411e-08	3.306190e-08	4.551504e-06	0.0
15	7.232696e-11	1.498419e-12	2.875161e-12	3.935117e-13	0.0
16	4.635570e-10	1.592078e-09	1.166622e-11	1.512085e-12	0.0
17	4.169009e-12	1.813553e-09	4.882033e-11	3.538474e-12	0.0
18	4.301358e-11	2.844314e-11	9.947057e-12	4.745606e-14	0.0
19	1.687789e-08	2.145998e-09	1.821880e-10	8.903792e-12	0.0
20	4.310452e-11	4.335999e-10	8.593350e-13	9.319111e-13	0.0
21	4.440111e-10	1.165793e-10	2.570160e-12	2.158742e-13	0.0
22	1.764089e-10	2.025972e-12	6.247978e-12	1.323063e-11	0.0
23	3.354867e-11	1.526561e-10	1.182255e-12	1.020765e-14	0.0
24	1.303846e-10	7.412743e-13	1.872853e-12	3.030813e-13	0.0
25	8.116035e-09	3.631886e-09	3.282077e-10	5.453356e-12	0.0
26	9.144753e-11	9.336454e-14	3.007248e-12	6.774755e-14	0.0
27	1.619710e-08	1.069428e-10	1.013088e-10	5.096730e-12	0.0
28	5.994063e-11	1.280458e-10	1.258378e-11	5.531853e-14	0.0

[29 rows x 29 columns]

```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
plt.grid()
```

```

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
↪05),L[row,1]+random.uniform(-0.05,0.05)),**fontL2)

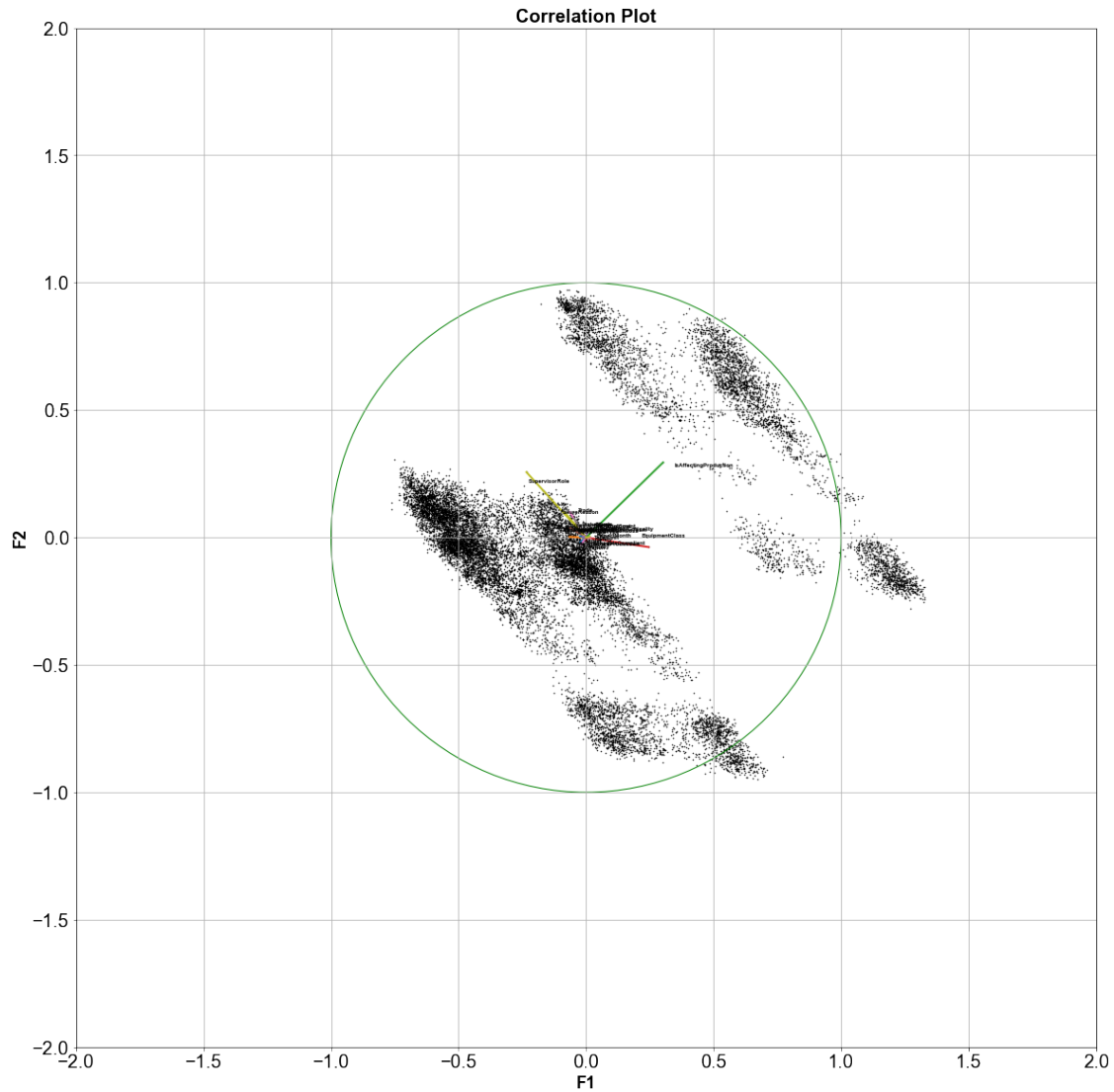
plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-2,2)
plt.ylim(-2,2)

plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)
plt.show()

```

```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
plt.grid()

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)
```

```

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
→05),L[row,1]+random.uniform(-0.05,0.05)),**fontL)

plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-0.3,0.3)
plt.ylim(-0.3,0.3)

plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)

```



```
[ ]: # Se convierte el array a dataframe para añadir nombres a los ejes.
```

```
pca_table = pd.DataFrame(
    data = modelo_pca.components_,
    columns = dfSystem.columns,
    index = PCNames
)
```

```
print(pca_table)
```

	FieldProductionTeam	EquipmentCode	EquipmentType	EquipmentClass	\
PC 0	5.059670e-03	4.465665e-02	9.714451e-17	-1.833574e-01	
PC 1	-6.481632e-02	-7.113495e-02	0.000000e+00	1.586745e-01	
PC 2	-3.444370e-02	-7.335182e-02	-9.714451e-17	1.724039e-01	
PC 3	2.775157e-02	3.845713e-01	3.469447e-17	-4.962724e-01	
PC 4	-2.180246e-01	1.777387e-01	-1.110223e-16	-7.222114e-02	
PC 5	5.917400e-01	-1.048288e-01	-6.245005e-17	5.273194e-02	
PC 6	-1.057825e-01	8.137301e-02	1.457168e-16	1.009564e-01	
PC 7	9.622293e-02	-3.212047e-01	2.012279e-16	-5.033336e-02	
PC 8	-1.148235e-01	-1.189903e-01	-1.387779e-16	-3.417031e-02	
PC 9	-1.748925e-01	1.019788e-01	4.163336e-17	1.647911e-01	
PC 10	-1.433773e-01	2.056416e-01	5.551115e-17	8.559888e-03	
PC 11	-4.341509e-02	2.986406e-02	3.469447e-18	-9.090380e-02	
PC 12	-3.434731e-02	-5.059356e-02	-8.326673e-17	-6.109194e-02	
PC 13	-1.100184e-01	1.586154e-01	9.714451e-17	7.435250e-02	
PC 14	-1.942002e-02	2.578665e-03	1.439820e-16	-5.817658e-02	
PC 15	-2.497981e-01	1.346583e-01	2.498002e-16	-2.538438e-02	
PC 16	-1.740267e-01	-1.824284e-01	-3.469447e-16	1.482085e-02	
PC 17	3.129393e-01	-1.221896e-01	-1.387779e-16	-2.438346e-02	
PC 18	2.317763e-01	5.117175e-01	8.326673e-17	3.474948e-02	
PC 19	-2.605430e-01	-3.671038e-01	-1.387779e-17	-2.285182e-02	
PC 20	1.810167e-01	-4.109476e-03	1.387779e-17	4.291267e-02	
PC 21	-2.662326e-01	9.335877e-02	-9.714451e-17	-7.092738e-02	
PC 22	5.206474e-02	2.852116e-01	-7.199102e-17	6.629399e-01	
PC 23	9.170960e-03	1.860665e-01	-7.502679e-17	-3.883373e-02	
PC 24	-1.654611e-01	-6.315847e-02	-2.498002e-16	2.773347e-01	
PC 25	7.160982e-02	-1.034470e-02	-1.526557e-16	-2.426429e-01	
PC 26	-2.112607e-01	8.564457e-03	-2.220446e-16	-8.944681e-02	
PC 27	2.422774e-03	-3.452591e-04	1.387779e-17	1.779728e-02	
PC 28	-2.102209e-17	-9.495866e-17	1.000000e+00	4.675931e-17	

	EquipmentCriticality	StatusCode	Priority	Cause	\
PC 0	1.152691e-01	1.649120e-02	-5.300962e-02	-3.546546e-02	
PC 1	-2.480116e-02	1.067163e-02	-1.056665e-01	-3.582295e-02	
PC 2	2.720170e-01	1.179449e-01	-3.777446e-01	-1.310537e-01	
PC 3	6.131679e-02	2.792854e-03	-1.581165e-01	9.055269e-02	
PC 4	-6.581573e-02	-4.340160e-02	-2.914810e-01	-5.072438e-02	
PC 5	2.200846e-01	1.890552e-01	-4.798228e-02	1.485607e-01	
PC 6	-2.552829e-01	-3.011173e-01	-6.333311e-02	5.465557e-02	

PC 7	-4.148792e-01	-9.173334e-02	-1.245458e-01	-3.342738e-01
PC 8	6.054005e-02	-3.923812e-01	1.701640e-01	3.225224e-01
PC 9	-3.481961e-01	1.897530e-01	-3.063669e-02	2.246514e-01
PC 10	-9.338316e-02	3.418630e-01	1.166208e-01	2.560964e-01
PC 11	1.849611e-01	-5.679378e-02	1.873851e-02	-2.545945e-02
PC 12	3.251103e-02	-6.047089e-02	1.107267e-01	-6.943445e-02
PC 13	-2.555401e-01	2.705677e-01	9.045927e-02	-6.250348e-02
PC 14	5.269300e-02	2.880302e-01	1.948437e-01	-3.224712e-01
PC 15	2.065373e-01	3.210826e-01	2.677549e-01	-3.670673e-01
PC 16	-1.468307e-02	3.528830e-01	-4.738388e-02	5.487375e-01
PC 17	5.697777e-02	2.391048e-01	8.493164e-02	1.333433e-01
PC 18	-1.266770e-02	-2.041127e-01	2.766501e-01	1.338779e-01
PC 19	3.050801e-01	-1.551983e-01	4.241781e-01	1.042290e-01
PC 20	-3.619065e-02	2.479652e-02	-4.310305e-04	8.078045e-03
PC 21	3.038151e-01	-2.888079e-02	-8.999152e-02	1.071671e-02
PC 22	9.688263e-02	-1.160677e-01	1.706249e-01	-8.558208e-02
PC 23	2.917776e-01	-9.592639e-02	-2.939042e-01	1.762156e-02
PC 24	1.073215e-01	3.447362e-03	-3.756121e-01	9.191355e-02
PC 25	-1.951001e-01	-6.575183e-02	5.419236e-02	8.985967e-03
PC 26	-6.451057e-02	-8.975121e-03	6.375361e-02	-2.204861e-02
PC 27	-8.922429e-03	-3.846708e-03	1.651925e-03	5.295900e-03
PC 28	6.318465e-17	-1.113952e-17	-3.248189e-16	1.666135e-16

	FailureReason	Duration	...	Manufacturer	Model \
PC 0	2.988558e-01	2.590319e-02	...	-1.236097e-02	9.918289e-02
PC 1	-2.530293e-01	5.869191e-03	...	1.913905e-02	-7.768025e-02
PC 2	2.941330e-02	-1.563210e-01	...	-6.049839e-04	-1.257121e-01
PC 3	-1.655751e-01	7.880062e-03	...	-4.008336e-02	3.918603e-01
PC 4	1.173956e-02	-1.138772e-01	...	-2.137938e-02	9.986038e-02
PC 5	-3.636108e-02	1.106897e-01	...	3.488285e-02	-1.670279e-01
PC 6	1.526532e-02	-1.365153e-01	...	-2.898709e-01	-2.279444e-01
PC 7	3.172721e-02	4.657823e-01	...	-9.470048e-02	-1.739982e-01
PC 8	3.305205e-02	2.598314e-01	...	1.533475e-01	2.391073e-01
PC 9	-9.856460e-02	1.421633e-01	...	1.869895e-01	7.509045e-02
PC 10	-3.814970e-02	-7.550530e-02	...	-6.060884e-01	-2.480750e-01
PC 11	-3.956941e-02	3.888571e-01	...	-5.137795e-01	1.308288e-01
PC 12	-8.970844e-03	1.595287e-01	...	-1.739862e-01	-7.972016e-03
PC 13	2.558807e-03	-1.712877e-01	...	2.201654e-01	9.543416e-02
PC 14	1.025904e-01	-1.550737e-01	...	-4.132183e-02	-7.497323e-02
PC 15	-2.033210e-01	4.007651e-01	...	1.698736e-01	1.456044e-02
PC 16	3.306744e-03	2.851171e-01	...	1.324865e-01	-1.186580e-01
PC 17	2.575144e-01	-2.832762e-02	...	-1.298338e-01	2.785975e-01
PC 18	7.852532e-02	1.060516e-01	...	1.815950e-01	-3.993377e-01
PC 19	-2.406469e-01	-2.902496e-01	...	-4.230312e-02	-6.967469e-02
PC 20	-4.534304e-01	-5.106671e-02	...	-7.418911e-02	1.366647e-01
PC 21	4.859351e-01	1.083119e-01	...	7.647123e-02	-2.421800e-01
PC 22	8.065281e-02	1.266343e-01	...	-1.219635e-01	3.004842e-01
PC 23	-4.084846e-01	1.235631e-01	...	8.026884e-02	-2.933929e-01

PC 24	5.735831e-02	5.053174e-02	...	-4.546665e-02	1.436621e-01
PC 25	3.212869e-02	-3.765601e-02	...	1.085782e-02	-7.680297e-02
PC 26	8.702018e-03	5.081450e-02	...	8.525856e-03	-4.393420e-02
PC 27	-2.315320e-03	-1.097842e-02	...	1.168957e-03	-2.448248e-03
PC 28	1.211122e-16	-2.457581e-17	...	1.625863e-16	6.028464e-17

	Safety	Reopened	ReportMonth	ReportWeekDay	\
PC 0	4.669377e-02	7.141042e-02	-3.592592e-02	3.629783e-02	
PC 1	-3.852175e-02	-4.584429e-02	5.218880e-02	-4.850229e-02	
PC 2	1.024433e-01	-1.594807e-02	8.675562e-02	1.895248e-03	
PC 3	1.774161e-01	6.499864e-03	8.708543e-02	-5.275333e-02	
PC 4	5.507209e-02	-2.077010e-02	-2.272842e-01	-9.437453e-02	
PC 5	-1.953847e-01	-1.038557e-01	-1.451279e-01	1.273028e-01	
PC 6	1.302182e-01	-8.328700e-02	5.424299e-01	-1.235747e-02	
PC 7	1.823196e-01	1.472871e-01	-1.773818e-01	-1.287282e-02	
PC 8	-5.224254e-01	1.750361e-01	1.078603e-01	7.727436e-02	
PC 9	-8.999918e-02	-2.042883e-01	-2.409387e-01	-4.542188e-01	
PC 10	-2.265766e-01	4.158445e-01	-1.760530e-01	9.724920e-02	
PC 11	1.529620e-02	-5.444432e-01	-1.712387e-01	-4.068962e-03	
PC 12	-3.348527e-02	-2.415534e-01	1.987427e-01	3.178252e-01	
PC 13	2.851224e-03	-2.885188e-01	-1.316387e-01	6.536875e-01	
PC 14	-4.293225e-01	-3.312199e-01	2.209493e-01	-3.941933e-01	
PC 15	7.788521e-02	2.935103e-01	1.725955e-01	7.325904e-03	
PC 16	2.700067e-01	-1.558072e-01	2.815460e-01	3.244409e-02	
PC 17	3.000179e-01	1.425872e-01	1.013562e-01	-1.884248e-01	
PC 18	1.806780e-01	-1.235483e-01	-4.255070e-02	-1.240623e-01	
PC 19	3.134660e-01	-7.995963e-02	-3.046255e-01	-8.573964e-02	
PC 20	-4.870686e-02	2.865741e-02	1.119926e-01	2.687424e-02	
PC 21	-2.617100e-02	4.367329e-03	-9.729838e-02	4.962849e-02	
PC 22	1.530429e-01	6.193570e-02	-8.814220e-02	1.446779e-03	
PC 23	-3.950860e-02	3.775188e-02	-6.816347e-02	2.232045e-03	
PC 24	-1.200779e-01	-2.961448e-02	-9.892753e-02	-1.356424e-03	
PC 25	-3.474013e-02	-1.242730e-02	-2.971745e-01	8.304596e-03	
PC 26	3.685290e-02	-7.387818e-03	-1.869972e-02	-4.208134e-03	
PC 27	1.743400e-03	-6.316637e-03	-7.052844e-04	3.207016e-03	
PC 28	5.590540e-18	-1.344202e-16	-3.694993e-17	-1.852903e-17	

	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
PC 0	8.123776e-03	-2.273816e-02	-6.571425e-03	-2.485740e-02
PC 1	-5.317654e-02	7.055977e-02	1.193539e-02	2.125276e-02
PC 2	-3.493632e-01	3.722361e-01	-2.377436e-02	4.097532e-02
PC 3	-1.223152e-01	1.638597e-01	3.577824e-02	-3.668576e-01
PC 4	4.255400e-01	-5.233234e-01	4.956398e-02	1.026687e-01
PC 5	1.818022e-01	-2.009330e-01	4.011739e-02	-9.622480e-03
PC 6	1.616259e-01	2.220291e-02	1.685961e-01	2.560408e-01
PC 7	-1.227413e-01	-2.397920e-04	1.571160e-01	-2.439458e-01
PC 8	-1.070639e-01	6.706067e-02	1.754042e-01	1.058998e-01
PC 9	-2.098831e-01	3.410723e-02	-3.255859e-01	2.017917e-01

PC 10	-1.030406e-01	1.941035e-02	-1.943656e-02	-8.156427e-02
PC 11	-1.423352e-01	8.007327e-04	2.300873e-01	2.419529e-01
PC 12	1.374148e-01	6.504638e-03	-7.809072e-01	-9.127683e-02
PC 13	-1.314894e-01	4.825957e-02	2.485421e-01	1.510736e-01
PC 14	1.292355e-01	5.973602e-02	1.848201e-01	-2.528363e-01
PC 15	1.772885e-01	3.811786e-02	3.151917e-02	2.738722e-01
PC 16	2.176140e-01	2.107096e-03	1.635412e-01	-2.830197e-01
PC 17	1.867955e-03	5.880994e-02	-2.748373e-02	4.723993e-01
PC 18	-9.501023e-02	6.184775e-02	3.372572e-02	-2.402896e-02
PC 19	7.747367e-03	3.953505e-03	4.452410e-02	-8.745139e-02
PC 20	6.531588e-02	1.667663e-02	1.036916e-01	8.682118e-03
PC 21	-2.139132e-02	-1.898969e-03	1.762196e-02	1.692053e-02
PC 22	1.760487e-01	8.483728e-02	4.296354e-02	-2.882099e-01
PC 23	5.317121e-02	2.279628e-02	-9.051787e-03	1.604834e-01
PC 24	2.367886e-01	2.457308e-01	1.473534e-02	-5.362255e-02
PC 25	5.315893e-01	6.511722e-01	5.034098e-03	1.272488e-01
PC 26	-2.056371e-02	-3.487180e-02	1.013266e-02	3.966864e-03
PC 27	-1.039231e-03	1.443068e-03	-2.176432e-03	4.715023e-04
PC 28	7.424929e-17	1.221065e-16	-1.023838e-16	1.253994e-17

[29 rows x 29 columns]

```
[ ]: def getMaxComponent(df,var):
    maxPC = []
    name    = var
    row     = df[name]
    mean    = df[name].mean()

    for i in range(0,len(PCNames)):
        weight = row[i]
        if weight >= mean*8:
            PC_num = i
            maxPC.append(PC_num)
    return(maxPC)
```

```
[ ]: dfSystem.columns
```

```
[ ]: Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
           'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
           'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
           'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
           'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
           'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
           'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
           'TBF_Equipment'],
          dtype='object')
```

```
[ ]: maxPCvars = 
    ↪ ["TBF_Equipment", "ActualDuration", "TotalCost", "IsAffectingProduction"]
```

```
[ ]: commonPCs = []
for i in range(0, len(maxPCvars)):
    name = maxPCvars[i]
    maxPCAit = getMaxComponent(pca_table, name)
    commonPCs.append(maxPCAit)

dictPCs = dict(zip(maxPCvars, commonPCs))

print(dictPCs)
```

```
{'TBF_Equipment': [6, 8, 9, 11, 13, 15, 17, 23, 25], 'ActualDuration': [4, 25],
'TotalCost': [0, 1, 27], 'IsAffectingProduction': [0, 1, 2, 4, 5, 8, 9, 10, 11,
12, 15, 16, 18, 20, 22, 25, 26, 27, 28]}
```

```
[ ]: import heapq

for i in range(0, len(dictPCs)):
    name = maxPCvars[i]
    print(name)
    PC_num = dictPCs[name]
    print(PC_num)
    for j in range(0, len(PC_num)):
        print(PC_num[j])
        PCval = pca_table.iloc[j]
        mean = PCval.mean()
        largest = heapq.nlargest(3, enumerate(PCval), key=lambda x: x[1])
        lowest = heapq.nsmallest(3, enumerate(PCval), key=lambda x: x[1])
        print("---Highest---")
        for k in range(0, len(largest)):
            indexTabla = largest[k][0]
            print(pca_table.columns[indexTabla])
        print("---Lowest---")
        for k in range(0, len(lowest)):
            indexTabla = lowest[k][0]
            print(pca_table.columns[indexTabla])
```

```
TBF_Equipment
[6, 8, 9, 11, 13, 15, 17, 23, 25]
6
---Highest---
TotalCost
MaterialCost
Trade
---Lowest---
EquipmentClass
```


Priority
 ReportMonth
 8
 ---Highest---
 MaterialCost
 TotalCost
 GrossProductionLoss
 ---Lowest---
 Trade
 TradeGroup
 SupervisorRole
 9
 ---Highest---
 IsAffectingProduction
 ScheduleCompliant
 EquipmentCriticality
 ---Lowest---
 Priority
 ActualDuration
 MaterialCost
 11
 ---Highest---
 Model
 EquipmentCode
 Assigned
 ---Lowest---
 EquipmentClass
 TBF_Equipment
 Trade
 13
 ---Highest---
 ActualDuration
 IsAffectingProduction
 AffectedProduction
 ---Lowest---
 ScheduleCompliant
 Priority
 ReportMonth
 15
 ---Highest---
 FieldProductionTeam
 Assigned
 EquipmentCriticality
 ---Lowest---
 ScheduleCompliant
 Safety
 Model
 17

```

---Highest---
ReportMonth
Assigned
TBF_Equipment
---Lowest---
StatusCode
Manufacturer
EquipmentCriticality
23
---Highest---
Duration
AffectedProduction
Safety
---Lowest---
EquipmentCriticality
Cause
EquipmentCode
25
---Highest---
Cause
Duration
AffectedProduction
---Lowest---
Safety
StatusCode
EquipmentCode
ActualDuration
[4, 25]
4
---Highest---
TotalCost
MaterialCost
Trade
---Lowest---
EquipmentClass
Priority
ReportMonth
25
---Highest---
MaterialCost
TotalCost
GrossProductionLoss
---Lowest---
Trade
TradeGroup
SupervisorRole
TotalCost
[0, 1, 27]

```

```

0
---Highest---
TotalCost
MaterialCost
Trade
---Lowest---
EquipmentClass
Priority
ReportMonth
1
---Highest---
MaterialCost
TotalCost
GrossProductionLoss
---Lowest---
Trade
TradeGroup
SupervisorRole
27
---Highest---
IsAffectingProduction
ScheduleCompliant
EquipmentCriticality
---Lowest---
Priority
ActualDuration
MaterialCost
IsAffectingProduction
[0, 1, 2, 4, 5, 8, 9, 10, 11, 12, 15, 16, 18, 20, 22, 25, 26, 27, 28]
0
---Highest---
TotalCost
MaterialCost
Trade
---Lowest---
EquipmentClass
Priority
ReportMonth
1
---Highest---
MaterialCost
TotalCost
GrossProductionLoss
---Lowest---
Trade
TradeGroup
SupervisorRole
2

```

```

---Highest---
IsAffectingProduction
ScheduleCompliant
EquipmentCriticality
---Lowest---
Priority
ActualDuration
MaterialCost
4
---Highest---
Model
EquipmentCode
Assigned
---Lowest---
EquipmentClass
TBF_Equipment
Trade
5
---Highest---
ActualDuration
IsAffectingProduction
AffectedProduction
---Lowest---
ScheduleCompliant
Priority
ReportMonth
8
---Highest---
FieldProductionTeam
Assigned
EquipmentCriticality
---Lowest---
ScheduleCompliant
Safety
Model
9
---Highest---
ReportMonth
Assigned
TBF_Equipment
---Lowest---
StatusCode
Manufacturer
EquipmentCriticality
10
---Highest---
Duration
AffectedProduction

```

Safety
 ---Lowest---
 EquipmentCriticality
 Cause
 EquipmentCode
 11
 ---Highest---
 Cause
 Duration
 AffectedProduction
 ---Lowest---
 Safety
 StatusCode
 EquipmentCode
 12
 ---Highest---
 SupervisorRole
 Cause
 Assigned
 ---Lowest---
 ReportWeekDay
 EquipmentCriticality
 TBF
 15
 ---Highest---
 Reopened
 StatusCode
 Cause
 ---Lowest---
 Manufacturer
 Model
 Safety
 16
 ---Highest---
 Duration
 TBF_Equipment
 TBF
 ---Lowest---
 Reopened
 Manufacturer
 AffectedProduction
 18
 ---Highest---
 ReportWeekDay
 AffectedProduction
 ReportMonth
 ---Lowest---
 TBF

Reopened
 Manufacturer
 20
 ---Highest---
 ReportWeekDay
 StatusCode
 TBF
 ---Lowest---
 Reopened
 EquipmentCriticality
 IsAffectingProduction
 22
 ---Highest---
 StatusCode
 ReportMonth
 Priority
 ---Lowest---
 Safety
 ReportWeekDay
 Reopened
 25
 ---Highest---
 Duration
 StatusCode
 Reopened
 ---Lowest---
 Cause
 FieldProductionTeam
 FailureReason
 26
 ---Highest---
 Cause
 StatusCode
 Duration
 ---Lowest---
 TBF_Equipment
 EquipmentCode
 Assigned
 27
 ---Highest---
 TBF_Equipment
 FieldProductionTeam
 Safety
 ---Lowest---
 Assigned
 IsAffectingProduction
 TradeGroup
 28

```

---Highest---
EquipmentCode
Priority
TradeGroup
---Lowest---
Model
SupervisorRole
StatusCode

```

1.2 Regression

```

[ ]: ## We know are going to create a model to predict TBF ##
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report

```

```

[ ]: column_names = dfSystem_scaled.columns
print(column_names)

```

```

Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
      'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
      'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
      'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
      'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
      'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
      'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
      'TBF_Equipment'],
      dtype='object')

```

```

[ ]: ## Target ##
y = dfSystem_scaled['TotalCost']
y.head()

```

```

[ ]: 0    0.010981
     1    0.004522
     2    0.002952
     3    0.003120
     4    0.005600
     Name: TotalCost, dtype: float64

```

```

[ ]: ## Rest of Variables ##
x = dfSystem_scaled[column_names.drop(['TotalCost'])]
x.head()

```

```

[ ]:   FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
0             0.766667         0.161445             0.0         0.142857
1             0.866667         0.184073             0.0         0.000000
2             0.866667         0.996954             0.0         0.285714

```

3	0.866667	0.996954	0.0	0.285714
4	0.866667	0.996954	0.0	0.285714

	EquipmentCriticality	StatusCode	Priority	Cause	FailureReason \
0	0.4	0.0	0.4	0.765957	0.742718
1	0.2	0.0	0.8	0.531915	0.033981
2	0.4	0.0	0.6	0.404255	0.742718
3	0.4	0.0	0.6	0.276596	0.786408
4	0.4	0.0	0.6	0.170213	0.786408

	Duration ...	Manufacturer	Model	Safety	Reopened	ReportMonth \
0	0.0 ...	0.714286	0.996466	0.0	0.0	1.0
1	0.0 ...	0.714286	0.996466	0.0	0.0	1.0
2	0.0 ...	0.714286	0.996466	0.0	0.0	1.0
3	0.0 ...	0.714286	0.996466	0.0	0.0	1.0
4	0.0 ...	0.714286	0.996466	0.0	0.0	1.0

	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
0	0.500000	0.002308		0.0 0.0	0.003223
1	0.166667	0.005385		0.0 0.0	0.098881
2	0.833333	0.017692		0.0 0.0	0.000000
3	1.000000	0.017692		0.0 0.0	0.000000
4	0.333333	0.025385		0.0 0.0	0.000000

[5 rows x 28 columns]

```
[ ]: #Separate train and test data
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
↪20,random_state=0)
```

```
[ ]: print("Size of the full data set: ",x.shape)
print("Size of the training data set: ",x_train.shape)
print("Size of the test data set: ",x_test.shape)
```

```
Size of the full data set: (19305, 28)
Size of the training data set: (15444, 28)
Size of the test data set: (3861, 28)
```

```
[ ]: EquipmentCode = dfSystem_scaled["EquipmentCode"]
EquipmentCode.head()
```

```
[ ]: 0    0.161445
1    0.184073
2    0.996954
3    0.996954
4    0.996954
Name: EquipmentCode, dtype: float64
```



```
[ ]: TotalCost_real = dfSystem_scaled["TotalCost"]
TotalCost_real.head()
```

```
[ ]: 0    0.010981
     1    0.004522
     2    0.002952
     3    0.003120
     4    0.005600
     Name: TotalCost, dtype: float64
```

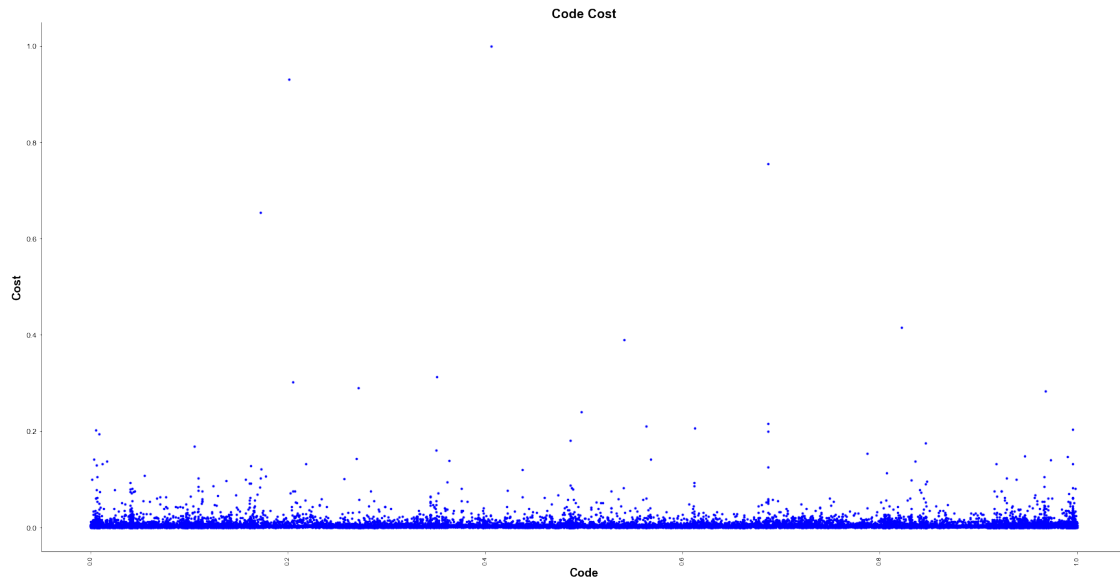
```
[ ]: df_code_cost = pd.DataFrame()

df_code_cost["Code"] = EquipmentCode
df_code_cost["Cost"] = TotalCost_real

df_code_cost.head()
```

```
[ ]:      Code      Cost
0  0.161445  0.010981
1  0.184073  0.004522
2  0.996954  0.002952
3  0.996954  0.003120
4  0.996954  0.005600
```

```
[ ]: ## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c =_
↪"blue" , s = 30)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```



```
[ ]: # importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
# fitting the training data
LR.fit(x_train,y_train)
```

```
[ ]: LinearRegression()
```

```
[ ]: y_prediction = LR.predict(x_test)
y_prediction
```

```
[ ]: array([0.00547437, 0.00825126, 0.00131596, ..., 0.02891223, 0.00279568,
0.00318972])
```

```
[ ]: # importing r2_score module
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

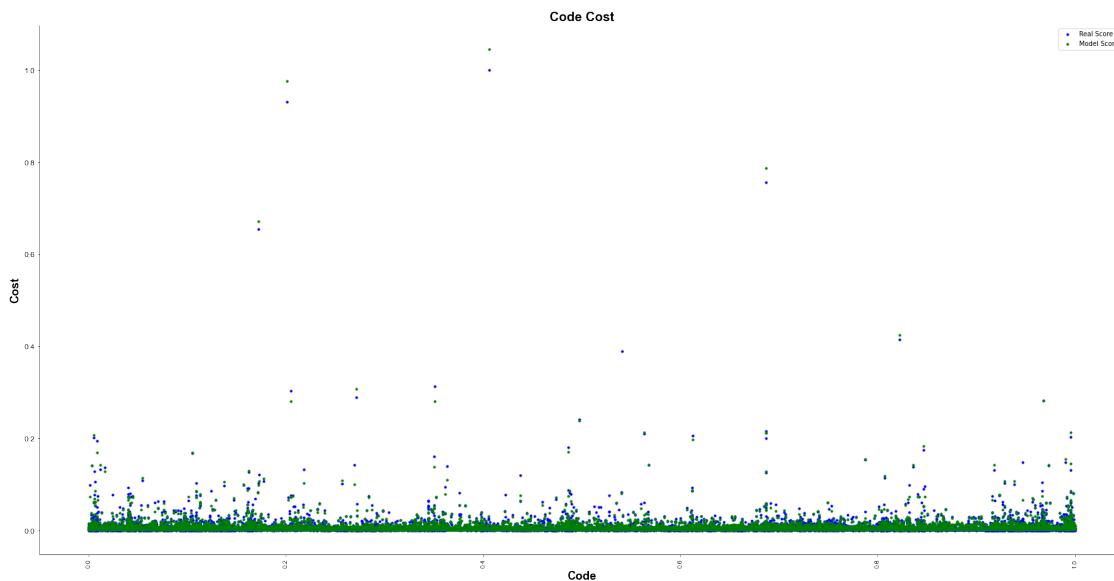
score = r2_score(y_test,y_prediction)
print("R2 score is ",score)
print("Mean squared error is ",mean_squared_error(y_test,y_prediction))
print("Mean absolute error is ",mean_absolute_error(y_test,y_prediction))
print("Root mean squared error is ",np.
↪sqrt(mean_squared_error(y_test,y_prediction)))
```

R2 score is 0.9432519187361306

Mean squared error is 1.9222211992265905e-05
Mean absolute error is 0.0018775864045447598
Root mean squared error is 0.004384314312668049

```
[ ]: y_prediction = LR.predict(x)

## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c = "blue", s = 40)
plt.scatter(x = df_code_cost["Code"],y = y_prediction, marker=".", c = "green", s = 40)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.legend(['Real Score', 'Model Score'])
plt.show()
```



2 Location

```
[ ]: ## Location Dataframe ##
dfLocation = df.loc[df["EquipmentType"] == 'Location']
dfLocation.head()
```

```
[ ]:      WorkOrder FieldProductionTeam EquipmentCode EquipmentType \
9      15513009      MCELROY      ED1777001-L      Location
118    15517654      MCELROY      DY7289001-L      Location
180    15468440      CONGER      UJ91CSF-L      Location
255    15449016      CARTHAGE      BP1146-L      Location
269    15435054      MCELROY      HM6555-L      Location

      EquipmentClass EquipmentCriticality StatusCode Priority \
9      Vessel - Separator      3 - Low      C      3
118      None      NA - Not Applicable      C      3
180      None      NA - Not Applicable      C      3
255      None      NA - Not Applicable      C      4
269      None      NA - Not Applicable      C      3

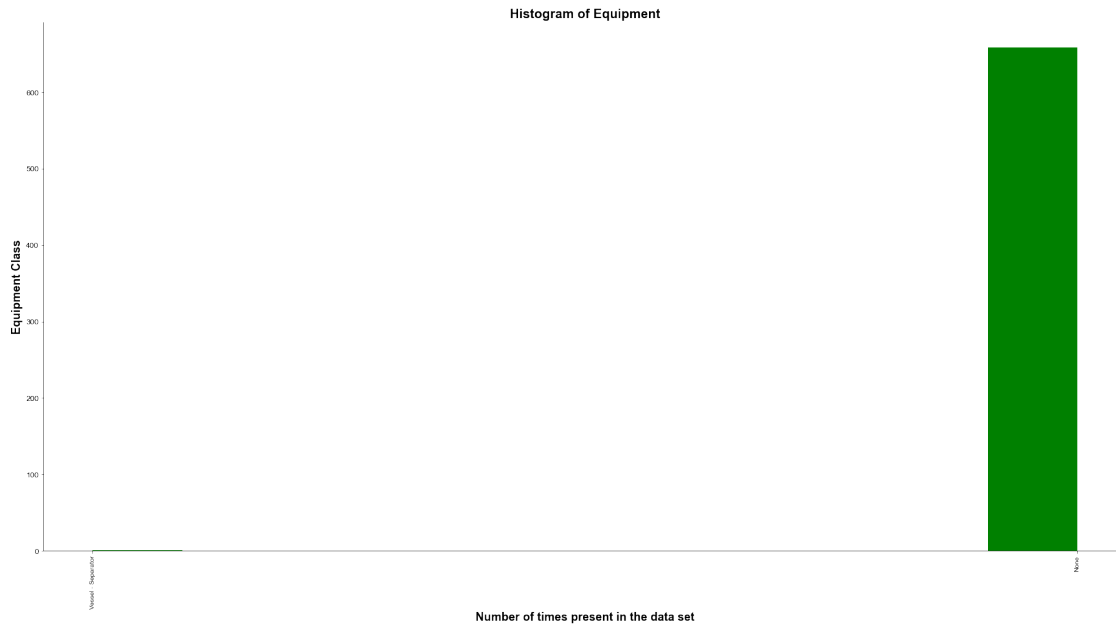
      Cause      FailureReason ... \
9      Broken      No code available, See comments ...
118      Open / Short / Ground      END DEVICE - SENSOR, THERMOWELL ...
180      No code available, See comments      No code available, See comments ...
255      No code needed, See comments      No code needed, See comments ...
269      No code needed, See comments      No code needed, See comments ...

      Manufacturer Model Safety Reopened ReportMonth ReportWeekDay \
9      None      None      N      N      12      4
118    None      None      N      N      1      7
180    None      None      N      N      12      2
255    None      None      N      N      11      3
269    None      None      N      N      11      6

      ActualDuration ScheduleCompliant TBF TBF_Equipment
9      1      Future Week      0      218.0
118    2      Future Week      0      NaN
180    29     Future Week      0      119.0
255    50     Future Week      0      NaN
269    61     Future Week      0      29.0
```

[5 rows x 30 columns]

```
[ ]: createGraphs(dfLocation)
```



```
[ ]: dfLocationCommonValues = getCommonValues(dfLocation,200)
print(dfLocationCommonValues)
```

```
['None']
```

```
[ ]: dfLocation = dfLocation.loc[dfLocation["EquipmentClass"] .
    ↳isin(dfLocationCommonValues)]
dfLocation.head()
```

```
[ ]:      WorkOrder FieldProductionTeam EquipmentCode EquipmentType EquipmentClass \
118    15517654          MCELROY    DY7289001-L      Location      None
180    15468440          CONGER     UJ91CSF-L      Location      None
255    15449016        CARTHAGE     BP1146-L      Location      None
269    15435054        MCELROY     HM6555-L      Location      None
367    15532276        MCELROY    BQ0614001-L      Location      None
```

```
      EquipmentCriticality StatusCode Priority \
118  NA - Not Applicable      C        3
180  NA - Not Applicable      C        3
255  NA - Not Applicable      C        4
269  NA - Not Applicable      C        3
367  NA - Not Applicable      C        3
```

```
      Cause FailureReason ... \
118      Open / Short / Ground  END DEVICE - SENSOR, THERMOWELL ...
180  No code available, See comments  No code available, See comments ...
255    No code needed, See comments    No code needed, See comments ...
```

```

269      No code needed, See comments      No code needed, See comments ...
367      No code needed, See comments      No code needed, See comments ...

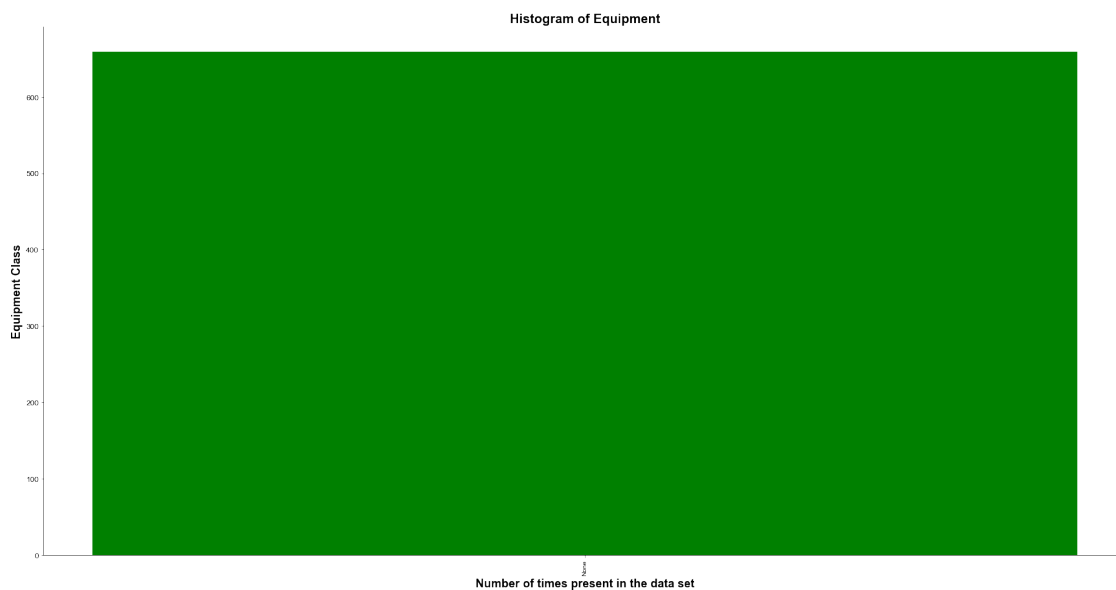
```

	Manufacturer	Model	Safety	Reopened	ReportMonth	ReportWeekDay	\
118	None	None	N	N	1	7	
180	None	None	N	N	12	2	
255	None	None	N	N	11	3	
269	None	None	N	N	11	6	
367	None	None	N	N	1	5	

	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
118	2	Future Week	0	NaN
180	29	Future Week	0	119.0
255	50	Future Week	0	NaN
269	61	Future Week	0	29.0
367	1	Future Week	0	NaN

[5 rows x 30 columns]

```
[ ]: createGraphs(dfLocation)
```



```

[ ]: ## Now that the data has been narrowed to the most common case we will encode
      ↪ it make some changes to it ##
dfLocation = EncodeData(dfLocation)
dfLocation.head()

```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
WorkOrder
15517654          11          121          0          0
15468440           6          432          0          0
15449016           4           59          0          0
15435054          11          265          0          0
15532276          11           77          0          0
```

```
      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason  \
WorkOrder
15517654           0           0           3    26           0
15468440           0           0           3    23           2
15449016           0           0           4    24           3
15435054           0           0           3    24           3
15532276           0           0           3    24           3
```

```
      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth  \
WorkOrder
15517654      1  ...           0     0     0           0           1
15468440      1  ...           0     0     0           0          12
15449016      1  ...           0     0     0           0          11
15435054      1  ...           0     0     0           0          11
15532276      1  ...           0     0     0           0           1
```

```
      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  \
WorkOrder
15517654           7           2           0  0
15468440           2          29           0  0
15449016           3          50           0  0
15435054           6          61           0  0
15532276           5           1           0  0
```

```
      TBF_Equipment
WorkOrder
15517654      NaN
15468440    119.0
15449016      NaN
15435054    29.0
15532276      NaN
```

[5 rows x 29 columns]

```
[ ]: ## We need to handle the datasets NA Values ##
dfLocation = dfLocation.fillna(dfLocation.mean())
dfLocation.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
WorkOrder
15517654      11      121      0      0
15468440      6      432      0      0
15449016      4      59      0      0
15435054      11      265      0      0
15532276      11      77      0      0
```

```
      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason  \
WorkOrder
15517654      0      0      3      26      0
15468440      0      0      3      23      2
15449016      0      0      4      24      3
15435054      0      0      3      24      3
15532276      0      0      3      24      3
```

```
      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth  \
WorkOrder
15517654      1  ...      0      0      0      0      1
15468440      1  ...      0      0      0      0      12
15449016      1  ...      0      0      0      0      11
15435054      1  ...      0      0      0      0      11
15532276      1  ...      0      0      0      0      1
```

```
      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  \
WorkOrder
15517654      7      2      0      0
15468440      2      29      0      0
15449016      3      50      0      0
15435054      6      61      0      0
15532276      5      1      0      0
```

```
      TBF_Equipment
WorkOrder
15517654      112.348718
15468440      119.000000
15449016      112.348718
15435054      29.000000
15532276      112.348718
```

[5 rows x 29 columns]

```
[ ]:  ## For further analysis we know need to standarize the dataset##
      ## Scales ##
      scaling_procedure_1 = MinMaxScaler(feature_range= (0,1))
```



```
[ ]: ## Scaled Data ##
columnSystem = dfLocation.columns
dfLocation_scaled = scaling_procedure_1.fit_transform(dfLocation)
dfLocation_scaled = pd.DataFrame(dfLocation_scaled, columns = columnSystem)
dfLocation_scaled.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
0          0.611111      0.261339          0.0          0.0
1          0.333333      0.933045          0.0          0.0
2          0.222222      0.127430          0.0          0.0
3          0.611111      0.572354          0.0          0.0
4          0.611111      0.166307          0.0          0.0

      EquipmentCriticality  StatusCode  Priority      Cause  FailureReason  \
0          0.0          0.0  0.666667  0.742857      0.000000
1          0.0          0.0  0.666667  0.657143      0.666667
2          0.0          0.0  1.000000  0.685714      1.000000
3          0.0          0.0  0.666667  0.685714      1.000000
4          0.0          0.0  0.666667  0.685714      1.000000

      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth  \
0          0.0  ...          0.0    0.0    0.0    0.0    0.000000
1          0.0  ...          0.0    0.0    0.0    0.0    1.000000
2          0.0  ...          0.0    0.0    0.0    0.0    0.909091
3          0.0  ...          0.0    0.0    0.0    0.0    0.909091
4          0.0  ...          0.0    0.0    0.0    0.0    0.000000

      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  TBF_Equipment
0          1.000000      0.006211          0.0  0.0    0.158238
1          0.166667      0.090062          0.0  0.0    0.167606
2          0.333333      0.155280          0.0  0.0    0.158238
3          0.833333      0.189441          0.0  0.0    0.040845
4          0.666667      0.003106          0.0  0.0    0.158238
```

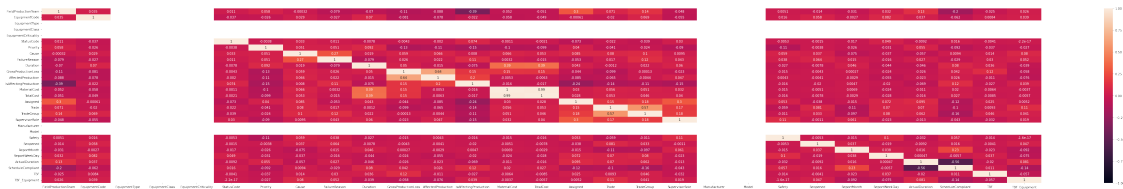
[5 rows x 29 columns]

```
[ ]: ##Here is the correlation plot of the system##

corr_df = dfLocation_scaled

corr = dfLocation_scaled.corr()
corr.style.background_gradient(cmap="coolwarm")

plt.figure(figsize=(70, 10))
heatmap = sns.heatmap(corr, vmin=-1, vmax=1, annot=True)
```



```
[ ]: ## PCA ##
#Only O3 Dataframe

Xdat = dfLocation_scaled

header_row= dfLocation_scaled.columns.values

y=dfLocation_scaled["TBF"]
```

```
[ ]: # create the PCA instance
pca = PCA()

# fit on data
pca = pca.fit(Xdat)

#Q eigenvectors
Q=pca.components_.T

#Explained deviations
s= np.sqrt(pca.explained_variance_)
```

```
[ ]: # transform data obtain sample scores
F = pca.transform(Xdat)
print("Sample scores")

## Sample Scores ##
sample_scores = pd.DataFrame(F)
print(sample_scores)
```

```
Sample scores
      0      1      2      3      4      5      6  \
0  -0.348656  0.861803  0.541134  0.011141  0.058621 -0.142754  0.554142
1  -0.514299  0.032709  0.403154  0.504064 -0.094284  0.565039 -0.546511
2  -0.595228 -0.088713  0.461464  0.073139  0.153430  0.895296  0.078454
3  -0.701071  0.068650  0.413186  0.544030 -0.216847  0.656864  0.254108
4    0.023437 -0.069105  1.273839 -0.018687  0.031892  0.308926  0.402669
..      ...      ...      ...      ...      ...      ...
654  0.210969  0.701423 -0.418999 -0.185785  0.130502  0.010487  0.313528
655 -0.564702 -0.144906 -0.138152 -0.259555 -0.025556  0.115344  0.525334
```

```

656 -0.553180 -0.152889 -0.150196 -0.307689 0.224370 0.099443 0.278315
657 -0.604635 -0.062690 0.125711 -0.464068 -0.399107 -0.086788 0.628350
658 0.571968 -0.501367 0.392586 -0.080358 -0.031239 -0.047796 0.330443

```

```

          7          8          9  ...          19          20          21  \
0  -0.138054 -0.008193 0.632368 ... -0.016922 0.003530 -0.008721
1  -0.373064 0.522876 0.300515 ... 0.005694 -0.017009 0.012988
2  -0.285791 0.070432 0.142696 ... 0.000938 -0.002930 0.008109
3  -0.242140 0.001566 -0.051445 ... -0.013206 -0.005488 0.001094
4   0.148073 -0.452209 -0.287078 ... -0.012046 0.006654 -0.017685
..
654 0.086220 -0.212105 -0.374855 ... -0.003462 -0.000082 -0.009028
655 0.363954 -0.150752 -0.250366 ... -0.006656 0.005511 -0.011052
656 0.480962 -0.022993 -0.360377 ... -0.013878 0.005640 -0.005487
657 0.194733 0.029195 -0.063292 ... -0.014846 -0.004253 -0.010365
658 -0.223978 -0.251087 -0.114427 ... 0.009234 0.420888 0.871870

```

```

          22          23          24          25          26  \
0  -0.002263 -0.001661 2.499793e-17 -5.934782e-17 3.702548e-17
1  -0.011429 0.000921 8.455344e-17 -4.952402e-17 4.165667e-17
2  -0.001796 0.000624 -2.154523e-17 -3.564787e-17 3.619770e-17
3  -0.004901 0.000182 1.802319e-17 -4.941218e-17 6.083798e-18
4  -0.011842 0.000774 7.868490e-18 -3.919516e-17 -1.456105e-17
..
654 0.002862 -0.000219 -2.821359e-17 1.682284e-17 1.759325e-17
655 -0.000895 0.001585 -1.772185e-17 -2.854621e-17 -7.637435e-18
656 0.001524 0.007281 -4.082744e-17 -1.347482e-17 3.174052e-17
657 -0.005426 -0.000062 1.169923e-16 -4.107262e-17 -1.023839e-16
658 0.007458 0.000017 -2.946214e-16 -3.905665e-17 2.380484e-16

```

```

          27          28
0   5.228026e-17 -3.470723e-17
1   2.818173e-17 -2.289229e-17
2   7.881672e-17 -1.180468e-17
3   1.000944e-16 -4.699377e-17
4   1.499235e-16 -5.196871e-17
..
654 -5.267604e-17 3.072855e-17
655 9.313330e-17 -3.180547e-17
656 8.552666e-17 -1.547490e-17
657 5.713849e-17 -2.102294e-17
658 -9.164133e-17 -6.021478e-18

```

[659 rows x 29 columns]

```

[ ]: # Obtain Factor Loadings
     L = np.zeros(shape=(Q.shape))

```

```

for col in range(0,len(s)-1):
    L[:,col] = Q[:,col] * s[col]
print("Factor Loadings")

columns_factorLoading = []

for i in range(0, 29):
    nombre = "F" + str(i)
    columns_factorLoading.append(nombre)

## Factor Loadings Dataframe ##
factor_loadings = pd.DataFrame(L, columns= columns_factorLoading)
print(factor_loadings)

```

Factor Loadings

	F0	F1	F2	F3	F4 \
0	-1.192397e-01	1.089286e-01	-3.790054e-02	1.389097e-02	-5.664050e-03
1	-4.870021e-03	5.237285e-02	8.881471e-03	1.690657e-01	-1.634028e-01
2	0.000000e+00	2.270964e-17	1.043625e-17	8.709398e-18	-2.351146e-17
3	2.661834e-17	-9.083857e-17	2.087251e-17	2.395084e-17	-2.778628e-17
4	-6.654586e-18	-2.270964e-17	-6.261752e-17	-4.354699e-17	-2.137406e-18
5	1.812781e-03	-1.862516e-03	1.586770e-03	-3.891950e-04	-1.541274e-04
6	-1.266442e-02	2.757553e-02	-6.132767e-03	-1.106369e-02	2.306439e-04
7	-9.046666e-03	-6.498845e-03	4.526550e-02	-1.035214e-01	-1.981604e-01
8	1.544263e-02	-2.973861e-02	4.987765e-02	-3.779352e-02	-8.962266e-02
9	-5.252130e-03	-6.116302e-03	-9.237816e-03	-2.775785e-04	-4.438142e-03
10	6.178809e-03	-3.512351e-03	2.658214e-03	-2.350830e-03	4.296474e-04
11	7.592145e-03	-7.697699e-03	6.456278e-03	-2.686481e-03	1.413906e-03
12	3.207540e-01	-1.355748e-01	1.854420e-01	5.728562e-02	9.660897e-04
13	-9.493105e-04	-2.024424e-03	-3.276683e-05	-2.761268e-03	-8.918968e-04
14	-1.076447e-03	-2.206059e-03	-1.277572e-04	-2.795352e-03	-6.196518e-04
15	-1.509653e-01	-2.435100e-02	1.411830e-02	1.446812e-02	-4.267988e-02
16	-6.943028e-02	-8.379762e-03	3.849674e-02	-1.733506e-02	-4.220447e-02
17	-5.749751e-02	-3.829603e-03	3.990257e-02	1.769899e-03	-4.633973e-02
18	-2.505203e-01	-3.065077e-01	8.544158e-02	1.376422e-02	1.851718e-02
19	0.000000e+00	1.290894e-30	-1.518675e-28	0.000000e+00	6.522845e-23
20	0.000000e+00	-0.000000e+00	0.000000e+00	0.000000e+00	-0.000000e+00
21	-7.570313e-03	-1.384622e-02	-1.168299e-03	4.980594e-04	-1.452853e-02
22	-1.621805e-04	-2.404644e-04	-7.517896e-04	3.828990e-04	-2.771332e-03
23	2.465701e-02	-1.138353e-01	-1.809917e-01	1.618662e-01	2.582602e-02
24	-1.206014e-02	1.481342e-02	-1.695135e-02	1.241493e-01	-8.058058e-02
25	-2.357213e-02	3.004893e-02	3.706086e-02	3.269293e-02	2.154914e-02
26	1.309327e-01	-1.537688e-01	-2.354118e-01	-1.035323e-01	-8.286070e-02
27	-2.130364e-04	1.815563e-03	-8.434603e-04	-9.648415e-04	-2.979027e-03
28	-3.388242e-03	4.504621e-03	1.874058e-02	-2.178528e-03	-4.070634e-03
	F5	F6	F7	F8	F9 ... \

0	3.520664e-02	3.875916e-02	1.134960e-01	9.758792e-03	-9.661672e-02	...
1	-8.766872e-02	-1.571022e-01	2.068354e-02	5.794219e-03	-1.281436e-02	...
2	-2.439000e-17	7.891286e-18	1.441745e-17	2.860834e-17	8.987151e-18	...
3	-4.065000e-18	1.578257e-17	2.883490e-17	4.168644e-17	7.489292e-18	...
4	0.000000e+00	-0.000000e+00	1.441745e-17	-5.312977e-18	-5.991434e-18	...
5	-2.479870e-04	1.766336e-03	-1.310603e-03	-1.619700e-03	-8.056036e-04	...
6	1.858235e-02	-3.663970e-03	-7.074948e-03	-1.342238e-02	1.035787e-03	...
7	1.197265e-01	7.057969e-04	-3.050198e-03	-4.511089e-02	9.267820e-02	...
8	1.075622e-01	3.792004e-03	-7.565737e-02	-4.357683e-03	-1.579420e-01	...
9	-4.935354e-03	-9.988457e-03	1.266034e-03	5.857298e-04	1.238988e-02	...
10	1.857820e-03	1.295357e-03	5.044566e-04	-2.316660e-03	1.826164e-03	...
11	2.611866e-03	-4.388444e-05	8.306936e-04	-1.658885e-03	1.290863e-03	...
12	9.821215e-03	3.972449e-02	8.554151e-02	1.392307e-02	-1.280738e-02	...
13	1.203365e-03	7.261468e-04	-4.124455e-04	1.813425e-03	3.514209e-03	...
14	2.629730e-04	2.658571e-04	-1.499431e-04	1.765123e-03	3.666391e-03	...
15	1.262702e-02	7.668473e-02	1.755032e-01	-3.172434e-02	1.389748e-02	...
16	-1.478440e-02	4.875121e-02	-1.602717e-02	1.815473e-01	2.983966e-02	...
17	-1.105851e-03	3.001159e-02	-3.843926e-03	1.163417e-01	-8.324824e-03	...
18	-3.654307e-02	-3.474265e-02	-2.420778e-02	-2.751087e-02	-1.275514e-02	...
19	-3.101349e-23	-0.000000e+00	0.000000e+00	-7.982237e-22	-3.656881e-22	...
20	4.845858e-25	-0.000000e+00	0.000000e+00	-0.000000e+00	0.000000e+00	...
21	-8.327726e-03	9.464100e-03	-2.729268e-03	-2.229029e-02	-7.117509e-03	...
22	1.312005e-03	-1.967293e-03	-2.024514e-03	3.308390e-03	-6.980248e-04	...
23	1.792732e-01	-1.593693e-03	-2.955037e-03	3.210503e-02	2.781716e-02	...
24	-7.752885e-02	2.056815e-01	-8.799666e-02	-5.230474e-02	3.253653e-03	...
25	3.201493e-02	-1.010280e-03	-5.258932e-03	2.056544e-03	6.370963e-03	...
26	-9.745752e-02	1.133168e-02	3.982651e-02	1.795691e-02	-3.194465e-02	...
27	-1.209389e-03	2.096408e-03	-1.422687e-03	5.135249e-04	-9.024720e-05	...
28	5.626340e-03	-6.296008e-03	1.560073e-03	8.420470e-03	1.112934e-03	...

	F19	F20	F21	F22	F23	\
0	2.493458e-04	-1.588463e-04	-5.051830e-04	-2.460551e-04	-1.861961e-06	
1	9.274834e-04	-2.128349e-04	3.222502e-04	-3.803892e-05	-4.868487e-06	
2	-8.977396e-18	9.355552e-19	7.982285e-19	5.137116e-18	-3.278921e-18	
3	1.484723e-17	1.323142e-17	1.170735e-17	-7.610542e-18	-4.482253e-18	
4	1.104910e-17	6.415236e-18	-0.000000e+00	6.088433e-18	-4.944910e-18	
5	2.820124e-04	1.660465e-02	3.455494e-02	4.133740e-04	5.370152e-06	
6	3.478960e-03	-3.412054e-05	-9.760366e-05	-8.675924e-05	-5.416201e-06	
7	-4.156181e-04	-1.882811e-04	-2.323256e-04	-5.902903e-05	4.926895e-06	
8	-4.383594e-04	-3.226451e-04	2.209602e-04	3.347823e-05	8.610454e-06	
9	-1.644896e-02	8.619182e-05	-1.637540e-04	-3.318841e-04	-4.344915e-06	
10	4.306040e-03	5.448815e-04	9.057986e-04	-2.254304e-02	1.510380e-05	
11	-5.620584e-03	-3.411348e-04	8.330429e-04	1.497818e-02	-6.577194e-06	
12	1.609095e-04	-2.592480e-05	-4.088178e-04	-1.150470e-04	-6.652750e-07	
13	3.235734e-02	9.376444e-05	-2.143327e-04	2.501543e-03	-3.491769e-03	
14	3.291248e-02	1.149019e-05	-2.505817e-04	2.672693e-03	3.428508e-03	
15	-2.368237e-04	4.606440e-04	4.415997e-04	2.258378e-04	2.236822e-06	
16	-3.817636e-04	-6.365340e-04	3.646877e-04	-3.730393e-04	1.015042e-06	

17	-1.866578e-04	3.691024e-04	2.351568e-04	2.477106e-04	2.794073e-06
18	3.431549e-04	-1.122423e-04	-3.465967e-04	-2.085837e-04	-4.515070e-06
19	5.524551e-18	6.415236e-18	-1.064305e-18	-8.561860e-19	-2.716517e-19
20	0.000000e+00	5.346030e-19	-1.064305e-18	0.000000e+00	9.779461e-18
21	4.616908e-04	9.009351e-05	2.165425e-04	-1.284014e-04	1.261745e-06
22	-3.083866e-04	3.473443e-02	-1.652088e-02	2.914719e-04	5.473397e-06
23	1.189597e-04	-1.404400e-04	2.349803e-04	9.978607e-06	6.146384e-06
24	-9.713122e-05	-4.177708e-05	-3.869239e-04	7.770429e-05	2.097432e-06
25	-6.915915e-05	5.062678e-05	-3.397591e-04	4.984555e-05	-2.491810e-06
26	2.744750e-04	-9.263476e-05	-7.504773e-05	4.832571e-05	-4.756586e-06
27	3.304412e-03	-5.826131e-05	-3.168544e-04	2.240226e-03	9.886478e-07
28	-1.377375e-04	-5.596113e-04	3.707762e-04	3.447163e-04	9.730957e-07

	F24	F25	F26	F27	F28
0	1.775001e-33	0.000000e+00	-0.000000e+00	0.000000e+00	0.0
1	2.229621e-33	1.599323e-33	-3.545409e-33	-3.083260e-33	0.0
2	3.067376e-17	1.186347e-18	2.062304e-17	-4.787914e-18	0.0
3	2.078252e-17	-5.464098e-18	-2.687823e-17	1.329628e-17	0.0
4	-4.724075e-18	-2.011715e-17	1.513865e-17	2.495282e-17	0.0
5	-1.633031e-32	-1.373777e-33	9.713033e-33	-6.097649e-33	0.0
6	4.141830e-33	2.642143e-33	-1.183746e-33	-3.951070e-33	0.0
7	1.944110e-33	-4.097564e-34	-5.441805e-33	1.151571e-33	0.0
8	2.863477e-33	-2.933254e-34	-1.746627e-33	-3.960881e-35	0.0
9	1.041016e-33	-5.042585e-34	3.741171e-33	3.856140e-33	0.0
10	-4.792193e-33	-4.258129e-33	1.019312e-32	-7.055095e-33	0.0
11	7.260893e-33	1.286927e-33	-9.054006e-33	-4.690712e-33	0.0
12	-9.363375e-34	-2.106983e-34	-9.351854e-34	5.281348e-34	0.0
13	-1.606951e-32	-2.568553e-32	-1.448024e-32	-4.811695e-32	0.0
14	3.266141e-32	1.800684e-32	9.756045e-33	4.211985e-32	0.0
15	-6.272640e-33	4.160648e-35	-3.458773e-33	-1.204566e-33	0.0
16	5.174223e-33	-2.115755e-33	-7.842221e-34	-2.952833e-33	0.0
17	-2.607800e-33	-2.385656e-33	2.948022e-33	6.178273e-33	0.0
18	4.159884e-33	1.071789e-34	3.370185e-34	3.403571e-33	0.0
19	-3.853171e-20	-7.463443e-19	-2.512905e-19	1.444743e-17	0.0
20	5.779041e-19	-3.096651e-17	-4.295856e-18	-1.908819e-17	0.0
21	-1.102057e-33	-1.649308e-34	5.107868e-33	-1.889225e-33	0.0
22	-8.420906e-34	-3.173225e-33	1.337335e-33	-2.657232e-33	0.0
23	-5.177692e-33	-7.259731e-34	2.246469e-33	4.094941e-34	0.0
24	1.808602e-33	-4.963142e-34	-6.103435e-34	1.143533e-33	0.0
25	9.302675e-34	4.219963e-33	-3.166917e-33	1.904115e-33	0.0
26	1.082303e-33	1.183565e-33	-2.928323e-33	-1.726383e-33	0.0
27	-1.192253e-32	4.815726e-33	1.091330e-32	-8.822297e-33	0.0
28	-2.204875e-33	1.967641e-33	-4.061021e-33	-2.014120e-33	0.0

[29 rows x 29 columns]

```
[ ]: #Obtain squared cosines
COS2=L**2
print("Square Cosines")

## COS2 Dataframe ##
COS2_DF = pd.DataFrame(COS2)
print(COS2_DF)

## Create a table ##
```

Square Cosines

	0	1	2	3	4	\
0	1.421811e-02	1.186543e-02	1.436451e-03	1.929591e-04	3.208147e-05	
1	2.371710e-05	2.742916e-03	7.888053e-05	2.858321e-02	2.670046e-02	
2	0.000000e+00	5.157279e-34	1.089154e-34	7.585362e-35	5.527889e-34	
3	7.085362e-34	8.251646e-33	4.356615e-34	5.736430e-34	7.720771e-34	
4	4.428351e-35	5.157279e-34	3.920954e-33	1.896340e-33	4.568504e-36	
5	3.286175e-06	3.468966e-06	2.517840e-06	1.514728e-07	2.375527e-08	
6	1.603876e-04	7.604097e-04	3.761083e-05	1.224051e-04	5.319663e-08	
7	8.184216e-05	4.223498e-05	2.048966e-03	1.071667e-02	3.926755e-02	
8	2.384749e-04	8.843851e-04	2.487780e-03	1.428351e-03	8.032221e-03	
9	2.758487e-05	3.740915e-05	8.533724e-05	7.704980e-08	1.969710e-05	
10	3.817768e-05	1.233661e-05	7.066101e-06	5.526402e-06	1.845969e-07	
11	5.764066e-05	5.925457e-05	4.168353e-05	7.217180e-06	1.999130e-06	
12	1.028831e-01	1.838053e-02	3.438875e-02	3.281642e-03	9.333293e-07	
13	9.011904e-07	4.098293e-06	1.073665e-09	7.624600e-06	7.954799e-07	
14	1.158738e-06	4.866695e-06	1.632189e-08	7.813990e-06	3.839684e-07	
15	2.279052e-02	5.929712e-04	1.993265e-04	2.093264e-04	1.821572e-03	
16	4.820564e-03	7.022041e-05	1.481999e-03	3.005045e-04	1.781217e-03	
17	3.305964e-03	1.466586e-05	1.592215e-03	3.132541e-06	2.147370e-03	
18	6.276042e-02	9.394700e-02	7.300264e-03	1.894537e-04	3.428860e-04	
19	0.000000e+00	1.666407e-60	2.306373e-56	0.000000e+00	4.254751e-45	
20	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
21	5.730963e-05	1.917179e-04	1.364922e-06	2.480631e-07	2.110781e-04	
22	2.630250e-08	5.782315e-08	5.651876e-07	1.466116e-07	7.680281e-06	
23	6.079682e-04	1.295848e-02	3.275798e-02	2.620067e-02	6.669834e-04	
24	1.454470e-04	2.194375e-04	2.873481e-04	1.541306e-02	6.493231e-03	
25	5.556451e-04	9.029379e-04	1.373507e-03	1.068827e-03	4.643654e-04	
26	1.714336e-02	2.364485e-02	5.541870e-02	1.071893e-02	6.865896e-03	
27	4.538451e-08	3.296271e-06	7.114253e-07	9.309191e-07	8.874604e-06	
28	1.148019e-05	2.029161e-05	3.512093e-04	4.745982e-06	1.657006e-05	
	5	6	7	8	9	...
0	1.239508e-03	1.502272e-03	1.288134e-02	9.523401e-05	9.334791e-03	...
1	7.685804e-03	2.468111e-02	4.278087e-04	3.357297e-05	1.642079e-04	...
2	5.948722e-34	6.227240e-35	2.078629e-34	8.184370e-34	8.076888e-35	...
3	1.652423e-35	2.490896e-34	8.314514e-34	1.737759e-33	5.608950e-35	...
4	0.000000e+00	0.000000e+00	2.078629e-34	2.822773e-35	3.589728e-35	...

5	6.149756e-08	3.119944e-06	1.717680e-06	2.623428e-06	6.489972e-07	...
6	3.453038e-04	1.342468e-05	5.005489e-05	1.801603e-04	1.072855e-06	...
7	1.433444e-02	4.981493e-07	9.303710e-06	2.034993e-03	8.589248e-03	...
8	1.156962e-02	1.437930e-05	5.724038e-03	1.898940e-05	2.494567e-02	...
9	2.435772e-05	9.976928e-05	1.602842e-06	3.430793e-07	1.535091e-04	...
10	3.451495e-06	1.677951e-06	2.544764e-07	5.366911e-06	3.334876e-06	...
11	6.821842e-06	1.925844e-09	6.900519e-07	2.751901e-06	1.666328e-06	...
12	9.645625e-05	1.578035e-03	7.317350e-03	1.938519e-04	1.640290e-04	...
13	1.448088e-06	5.272892e-07	1.701113e-07	3.288509e-06	1.234966e-05	...
14	6.915480e-08	7.067997e-08	2.248293e-08	3.115659e-06	1.344242e-05	...
15	1.594415e-04	5.880548e-03	3.080138e-02	1.006434e-03	1.931400e-04	...
16	2.185785e-04	2.376680e-03	2.568703e-04	3.295942e-02	8.904056e-04	...
17	1.222907e-06	9.006956e-04	1.477577e-05	1.353540e-02	6.930270e-05	...
18	1.335396e-03	1.207051e-03	5.860164e-04	7.568482e-04	1.626935e-04	...
19	9.618366e-46	0.000000e+00	0.000000e+00	6.371611e-43	1.337278e-43	...
20	2.348234e-49	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	...
21	6.935102e-05	8.956919e-05	7.448906e-06	4.968571e-04	5.065893e-05	...
22	1.721356e-06	3.870242e-06	4.098659e-06	1.094544e-05	4.872386e-07	...
23	3.213887e-02	2.539857e-06	8.732244e-06	1.030733e-03	7.737946e-04	...
24	6.010723e-03	4.230490e-02	7.743413e-03	2.735786e-03	1.058626e-05	...
25	1.024956e-03	1.020666e-06	2.765637e-05	4.229375e-06	4.058916e-05	...
26	9.497968e-03	1.284070e-04	1.586151e-03	3.224507e-04	1.020461e-03	...
27	1.462622e-06	4.394928e-06	2.024039e-06	2.637079e-07	8.144557e-09	...
28	3.165570e-05	3.963972e-05	2.433827e-06	7.090432e-05	1.238622e-06	...

	19	20	21	22	23	\
0	6.217332e-08	2.523216e-08	2.552098e-07	6.054309e-08	3.466898e-12	
1	8.602254e-07	4.529868e-08	1.038452e-07	1.446959e-09	2.370217e-11	
2	8.059363e-35	8.752636e-37	6.371688e-37	2.638996e-35	1.075132e-35	
3	2.204403e-34	1.750706e-34	1.370621e-34	5.792035e-35	2.009059e-35	
4	1.220827e-34	4.115525e-35	0.000000e+00	3.706902e-35	2.445213e-35	
5	7.953098e-08	2.757144e-04	1.194044e-03	1.708781e-07	2.883853e-11	
6	1.210316e-05	1.164212e-09	9.526475e-09	7.527165e-09	2.933523e-11	
7	1.727384e-07	3.544979e-08	5.397517e-08	3.484426e-09	2.427429e-11	
8	1.921590e-07	1.040999e-07	4.882342e-08	1.120792e-09	7.413991e-11	
9	2.705684e-04	7.429030e-09	2.681536e-08	1.101470e-07	1.887828e-11	
10	1.854198e-05	2.968958e-07	8.204711e-07	5.081889e-04	2.281249e-10	
11	3.159097e-05	1.163730e-07	6.939604e-07	2.243460e-04	4.325948e-11	
12	2.589186e-08	6.720955e-10	1.671320e-07	1.323581e-08	4.425908e-13	
13	1.046997e-03	8.791771e-09	4.593852e-08	6.257718e-06	1.219245e-05	
14	1.083231e-03	1.320245e-10	6.279117e-08	7.143288e-06	1.175466e-05	
15	5.608547e-08	2.121929e-07	1.950103e-07	5.100273e-08	5.003372e-12	
16	1.457435e-07	4.051756e-07	1.329971e-07	1.391583e-07	1.030310e-12	
17	3.484114e-08	1.362366e-07	5.529872e-08	6.136054e-08	7.806844e-12	
18	1.177553e-07	1.259834e-08	1.201293e-07	4.350716e-08	2.038586e-11	
19	3.052067e-35	4.115525e-35	1.132744e-36	7.330544e-37	7.379464e-38	
20	0.000000e+00	2.858004e-37	1.132744e-36	0.000000e+00	9.563786e-35	
21	2.131584e-07	8.116841e-09	4.689064e-08	1.648691e-08	1.592000e-12	

22	9.510227e-08	1.206481e-03	2.729395e-04	8.495584e-08	2.995808e-11
23	1.415141e-08	1.972340e-08	5.521575e-08	9.957260e-11	3.777804e-11
24	9.434473e-09	1.745325e-09	1.497101e-07	6.037956e-09	4.399219e-12
25	4.782989e-09	2.563071e-09	1.154363e-07	2.484579e-09	6.209119e-12
26	7.533652e-08	8.581199e-09	5.632161e-09	2.335374e-09	2.262511e-11
27	1.091914e-05	3.394380e-09	1.003967e-07	5.018614e-06	9.774245e-13
28	1.897161e-08	3.131648e-07	1.374750e-07	1.188293e-07	9.469153e-13

	24	25	26	27	28
0	3.150628e-66	0.000000e+00	0.000000e+00	0.000000e+00	0.0
1	4.971210e-66	2.557834e-66	1.256992e-65	9.506495e-66	0.0
2	9.408797e-34	1.407419e-36	4.253099e-34	2.292412e-35	0.0
3	4.319132e-34	2.985637e-35	7.224390e-34	1.767910e-34	0.0
4	2.231688e-35	4.046997e-34	2.291789e-34	6.226431e-34	0.0
5	2.666790e-64	1.887262e-66	9.434302e-65	3.718132e-65	0.0
6	1.715476e-65	6.980922e-66	1.401254e-66	1.561095e-65	0.0
7	3.779563e-66	1.679003e-67	2.961324e-65	1.326117e-66	0.0
8	8.199498e-66	8.603981e-68	3.050707e-66	1.568858e-69	0.0
9	1.083715e-66	2.542767e-67	1.399636e-65	1.486981e-65	0.0
10	2.296511e-65	1.813167e-65	1.038998e-64	4.977436e-65	0.0
11	5.272056e-65	1.656180e-66	8.197502e-65	2.200278e-65	0.0
12	8.767279e-67	4.439379e-68	8.745718e-67	2.789264e-67	0.0
13	2.582292e-64	6.597467e-64	2.096774e-64	2.315241e-63	0.0
14	1.066768e-63	3.242464e-64	9.518041e-65	1.774082e-63	0.0
15	3.934601e-65	1.731099e-69	1.196311e-65	1.450978e-66	0.0
16	2.677258e-65	4.476418e-66	6.150043e-67	8.719222e-66	0.0
17	6.800622e-66	5.691354e-66	8.690836e-66	3.817105e-65	0.0
18	1.730463e-65	1.148732e-68	1.135815e-67	1.158430e-65	0.0
19	1.484692e-39	5.570299e-37	6.314691e-38	2.087282e-34	0.0
20	3.339731e-37	9.589248e-34	1.845438e-35	3.643589e-34	0.0
21	1.214531e-66	2.720216e-68	2.609032e-65	3.569170e-66	0.0
22	7.091166e-67	1.006936e-65	1.788466e-66	7.060881e-66	0.0
23	2.680850e-65	5.270369e-67	5.046624e-66	1.676854e-67	0.0
24	3.271040e-66	2.463278e-67	3.725192e-67	1.307668e-66	0.0
25	8.653977e-67	1.780809e-65	1.002937e-65	3.625654e-66	0.0
26	1.171379e-66	1.400825e-66	8.575075e-66	2.980398e-66	0.0
27	1.421467e-64	2.319122e-65	1.191002e-64	7.783292e-65	0.0
28	4.861474e-66	3.871610e-66	1.649189e-65	4.056681e-66	0.0

[29 rows x 29 columns]

```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
plt.grid()
```

```

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
↪05),L[row,1]+random.uniform(-0.05,0.05)),**fontL2)

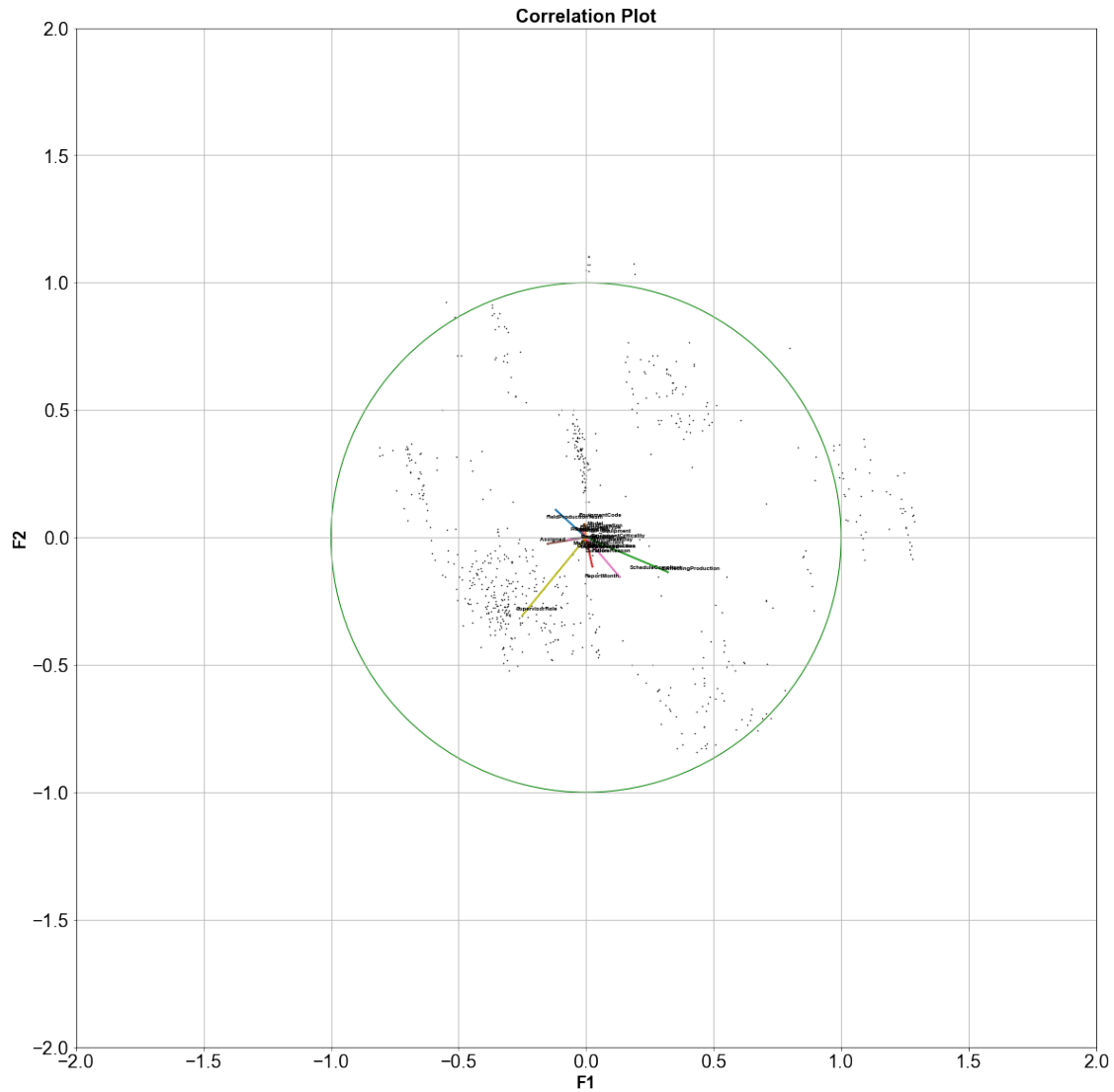
plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-2,2)
plt.ylim(-2,2)

plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)
plt.show()

```



```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
plt.grid()

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)
```

```

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
→05),L[row,1]+random.uniform(-0.05,0.05)),**fontL)

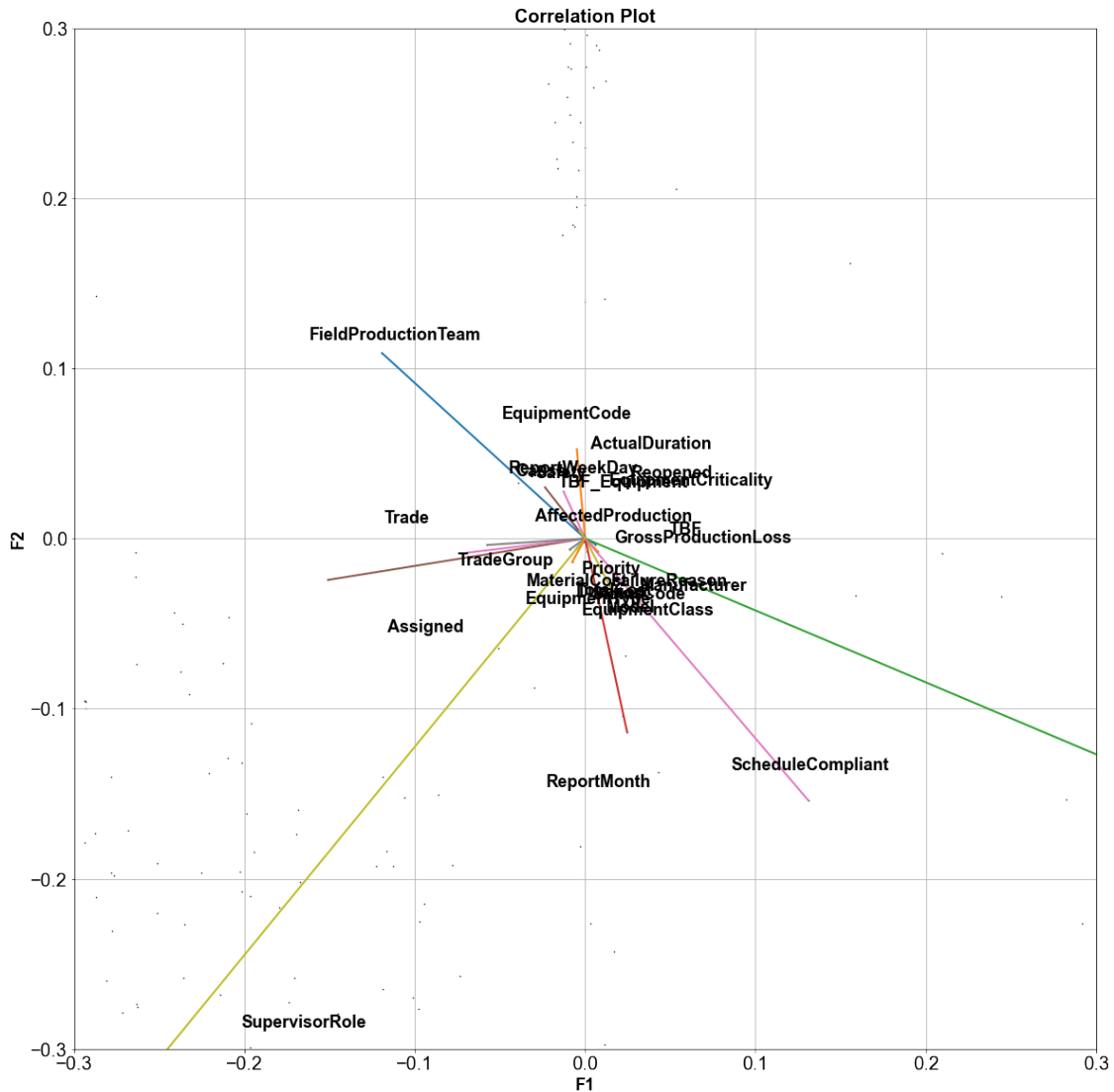
plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-0.3,0.3)
plt.ylim(-0.3,0.3)

plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)

```



```
[ ]: pca_pipe = make_pipeline(StandardScaler(), PCA())
pca_pipe.fit(dfLocation)
dfLocationColumns = dfLocation.columns

modelo_pca = pca_pipe.named_steps['pca']
```

```
[ ]: PCNames = []
for i in range(0, len(dfLocationColumns)):
    nombre = "PC " + str(i)
    PCNames.append(nombre)
```

```
[ ]: # Se convierte el array a dataframe para añadir nombres a los ejes.
```

```
pca_table = pd.DataFrame(
    data = modelo_pca.components_,
    columns = dfSystem.columns,
    index = PCNames
)
```

```
print(pca_table)
```

	FieldProductionTeam	EquipmentCode	EquipmentType	EquipmentClass	\
PC 0	-3.279884e-01	-9.104279e-02	-5.551115e-17	-2.775558e-17	
PC 1	1.105740e-01	-3.177288e-03	5.551115e-17	1.387779e-17	
PC 2	-8.904411e-02	-6.082843e-02	1.110223e-16	-5.551115e-17	
PC 3	7.769053e-02	8.726323e-03	1.387779e-17	-8.326673e-17	
PC 4	2.748459e-01	-1.116910e-01	0.000000e+00	1.110223e-16	
PC 5	1.796227e-01	-1.851196e-01	-2.775558e-17	5.551115e-17	
PC 6	1.034196e-02	-2.564336e-02	-7.806256e-17	-8.847090e-17	
PC 7	1.295520e-01	4.879020e-01	1.665335e-16	-1.110223e-16	
PC 8	-1.259673e-01	3.041206e-01	5.551115e-17	-5.204170e-17	
PC 9	-1.839943e-01	-3.311677e-01	5.551115e-17	-2.983724e-16	
PC 10	3.633918e-01	4.061731e-03	2.220446e-16	2.775558e-17	
PC 11	1.628386e-01	-4.019669e-01	-1.387779e-16	-1.804112e-16	
PC 12	-1.883036e-01	3.046928e-01	2.775558e-17	-0.000000e+00	
PC 13	-2.613815e-01	-2.167576e-01	-1.110223e-16	-2.775558e-17	
PC 14	-9.641401e-02	-2.773410e-01	-6.938894e-17	2.775558e-17	
PC 15	-1.195167e-01	5.224305e-02	6.938894e-18	0.000000e+00	
PC 16	-1.000164e-01	-1.629304e-01	1.509209e-16	-0.000000e+00	
PC 17	-2.721007e-01	-4.058067e-02	-1.786765e-16	-0.000000e+00	
PC 18	1.635084e-01	-2.646961e-01	-1.595946e-16	-3.330669e-16	
PC 19	4.230366e-01	-7.317017e-02	-2.220446e-16	1.804112e-16	
PC 20	2.615992e-01	1.196624e-01	1.110223e-16	-8.326673e-17	
PC 21	1.163502e-01	-2.219526e-02	5.551115e-17	-2.220446e-16	
PC 22	-1.848557e-01	-2.778393e-02	-1.110223e-16	-5.551115e-17	
PC 23	-2.605907e-03	-7.478360e-03	-4.770490e-16	-3.087808e-16	
PC 24	0.000000e+00	7.992721e-17	2.020249e-01	3.166067e-02	
PC 25	9.427856e-18	-2.816065e-17	9.520470e-01	7.499250e-02	
PC 26	0.000000e+00	-4.342667e-17	-1.629604e-01	-2.903019e-01	
PC 27	-0.000000e+00	5.093577e-17	-3.599261e-02	7.167419e-01	
PC 28	0.000000e+00	-8.641415e-17	-1.579265e-01	6.287921e-01	

	EquipmentCriticality	StatusCode	Priority	Cause	\
PC 0	0.000000e+00	3.685532e-02	-1.436375e-01	3.379923e-03	
PC 1	5.551115e-17	-3.255973e-02	-3.504815e-02	1.054836e-01	
PC 2	0.000000e+00	1.548395e-02	-1.705327e-01	2.429254e-01	
PC 3	8.326673e-17	-2.618056e-03	1.539327e-01	-1.283787e-01	
PC 4	-2.775558e-17	-8.460281e-02	-1.078869e-01	-2.937491e-01	
PC 5	-5.551115e-17	-4.258975e-02	5.101869e-01	4.659542e-01	
PC 6	-1.665335e-16	1.512318e-01	-2.152986e-01	1.686316e-01	

PC 7	1.526557e-16	2.664430e-02	1.901980e-02	2.147928e-01
PC 8	-1.942890e-16	-3.071857e-01	2.028102e-02	-1.016102e-01
PC 9	-2.775558e-17	4.598563e-01	2.105439e-01	-1.051102e-01
PC 10	1.665335e-16	6.029675e-01	-2.916929e-02	5.195099e-02
PC 11	-2.220446e-16	-2.554701e-02	-3.201707e-01	-7.781111e-02
PC 12	-4.163336e-17	4.745978e-01	2.226426e-02	-3.068224e-02
PC 13	-6.938894e-17	-8.706957e-02	2.902745e-01	-1.850836e-01
PC 14	-7.285839e-17	-1.216728e-01	3.284818e-01	-4.183074e-02
PC 15	-5.551115e-17	-1.049418e-01	-1.364957e-01	1.149478e-01
PC 16	1.249001e-16	1.581253e-02	-4.332057e-02	4.680871e-01
PC 17	-0.000000e+00	-6.375910e-02	-1.834518e-01	4.483477e-01
PC 18	-1.110223e-16	-5.821017e-02	-4.326180e-01	7.725645e-03
PC 19	-5.551115e-17	-1.460124e-01	1.105820e-01	1.664098e-01
PC 20	1.526557e-16	-5.765961e-02	9.394987e-02	-4.306553e-02
PC 21	-1.387779e-16	-4.439656e-02	4.937740e-02	6.295899e-02
PC 22	-1.110223e-16	4.204543e-02	-3.363757e-02	-3.695907e-02
PC 23	3.469447e-16	1.069355e-03	-4.128568e-03	7.038507e-03
PC 24	-6.564411e-01	1.524811e-17	3.503118e-17	2.016214e-16
PC 25	2.966185e-01	-1.117720e-16	-3.980270e-18	9.562986e-18
PC 26	5.964480e-01	-4.931679e-17	-4.211696e-17	-5.832836e-18
PC 27	-6.567476e-02	1.066914e-16	-1.393455e-16	-2.881120e-17
PC 28	3.479068e-01	1.581130e-16	-4.949946e-17	-7.823591e-17

	FailureReason	Duration	...	Manufacturer	Model \
PC 0	5.132916e-03	1.790158e-01	...	0.000000e+00	0.000000e+00
PC 1	7.053799e-03	3.255860e-01	...	0.000000e+00	0.000000e+00
PC 2	2.733332e-01	-2.157313e-01	...	5.293956e-23	0.000000e+00
PC 3	-8.057880e-02	1.877379e-02	...	1.355253e-20	1.694066e-21
PC 4	-3.794833e-01	2.235349e-03	...	-2.710505e-20	-3.388132e-21
PC 5	3.372205e-01	1.189010e-01	...	3.469447e-18	1.734723e-18
PC 6	2.128134e-01	-1.113462e-01	...	-1.110223e-16	-2.220446e-16
PC 7	1.408424e-01	7.231637e-02	...	-1.387779e-17	2.255141e-17
PC 8	1.358963e-01	1.284785e-01	...	-1.110223e-16	-5.551115e-17
PC 9	2.447623e-01	-9.500316e-03	...	2.775558e-17	1.110223e-16
PC 10	-1.427230e-01	-8.488989e-02	...	-2.775558e-17	5.551115e-17
PC 11	1.256146e-01	-2.792974e-01	...	0.000000e+00	-9.020562e-17
PC 12	-1.757030e-01	3.204430e-01	...	-5.551115e-17	8.326673e-17
PC 13	-6.828878e-02	1.103051e-01	...	0.000000e+00	-2.775558e-17
PC 14	-1.540243e-01	1.142666e-01	...	-5.551115e-17	-1.665335e-16
PC 15	1.499760e-02	-2.300063e-01	...	-8.326673e-17	0.000000e+00
PC 16	-5.011680e-01	1.059327e-01	...	2.220446e-16	-5.551115e-17
PC 17	-2.326435e-01	-7.145029e-02	...	-8.326673e-17	2.775558e-17
PC 18	2.309089e-01	6.923145e-01	...	-2.775558e-17	1.387779e-17
PC 19	-2.038966e-01	2.266522e-02	...	2.602085e-18	-6.505213e-19
PC 20	1.161654e-01	7.907560e-03	...	-1.734723e-18	8.673617e-19
PC 21	-8.977225e-02	-5.221744e-02	...	1.734723e-18	-5.421011e-19
PC 22	2.655206e-03	-4.159055e-02	...	3.252607e-19	-6.776264e-20
PC 23	1.069332e-02	-2.148777e-03	...	-2.456396e-20	3.970467e-21

PC 24	1.442589e-17	-3.991227e-17	...	-1.133433e-01	7.172321e-01
PC 25	1.927203e-17	1.296526e-16	...	-4.732213e-06	7.456129e-07
PC 26	-4.494964e-17	-1.028328e-16	...	3.220700e-01	6.555070e-01
PC 27	5.096217e-17	5.604650e-17	...	6.927452e-01	2.786446e-02
PC 28	5.749803e-17	1.980065e-17	...	-6.352388e-01	2.347600e-01

	Safety	Reopened	ReportMonth	ReportWeekDay	\
PC 0	8.458975e-06	-1.729460e-02	1.040294e-01	-7.509123e-02	
PC 1	-5.682482e-03	1.247500e-02	-5.403388e-02	7.691214e-03	
PC 2	3.905478e-02	5.981974e-02	-1.605235e-01	-1.167288e-02	
PC 3	-2.251804e-01	-8.585660e-02	-1.658508e-01	-1.695608e-01	
PC 4	8.705433e-02	-1.955338e-01	1.555261e-01	1.080962e-01	
PC 5	-7.763114e-02	8.583808e-03	3.719998e-02	-2.955240e-01	
PC 6	5.783649e-01	-1.044835e-01	1.132859e-01	7.332965e-02	
PC 7	2.206427e-01	5.482533e-02	-3.400729e-02	5.120484e-01	
PC 8	-1.049578e-01	4.942411e-01	6.128639e-01	-7.119124e-02	
PC 9	-1.871741e-01	-1.796655e-01	3.143368e-01	3.088013e-01	
PC 10	-8.865331e-02	4.698766e-01	5.439262e-02	9.366654e-02	
PC 11	3.338053e-02	3.476164e-01	9.005041e-02	-1.654950e-01	
PC 12	-5.255200e-02	7.264957e-02	-1.176288e-01	-2.501738e-01	
PC 13	4.154643e-01	4.825037e-01	-4.201728e-01	7.378973e-02	
PC 14	-4.391048e-03	6.468935e-02	1.150020e-01	4.902467e-01	
PC 15	-4.982791e-01	2.383862e-01	-2.846522e-01	2.502567e-01	
PC 16	1.497922e-01	8.452148e-02	1.662573e-01	-1.516285e-01	
PC 17	-1.616378e-01	-7.252649e-02	2.431099e-02	1.932408e-01	
PC 18	-1.292166e-01	3.954135e-02	-1.326585e-01	1.388225e-01	
PC 19	-2.495061e-02	4.698865e-02	9.962849e-02	8.798864e-02	
PC 20	7.810770e-03	-4.240879e-02	-1.234147e-01	2.906061e-02	
PC 21	-7.578360e-02	3.037061e-02	-2.285071e-01	5.238528e-02	
PC 22	-2.356995e-02	1.388355e-02	6.546702e-04	1.702030e-02	
PC 23	8.727091e-04	1.056072e-03	1.031172e-02	3.028180e-03	
PC 24	1.584929e-16	-6.587215e-17	7.242953e-17	4.049096e-18	
PC 25	-5.238803e-17	8.418984e-17	-5.337851e-17	1.166265e-17	
PC 26	-3.439581e-19	9.979604e-17	1.252640e-17	3.004184e-18	
PC 27	-1.217862e-16	-9.965786e-18	-8.575103e-17	-7.591584e-17	
PC 28	-2.583793e-17	-1.378818e-17	-5.129646e-17	-5.468470e-17	

	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
PC 0	-2.480750e-01	3.195196e-01	2.384169e-02	-1.123042e-01
PC 1	8.863314e-02	-1.241205e-01	2.004611e-02	4.782659e-02
PC 2	1.260834e-01	-1.874203e-01	7.827619e-02	1.156268e-01
PC 3	5.066760e-01	-5.045897e-01	-5.610707e-02	1.239617e-01
PC 4	5.052713e-02	-3.756081e-03	9.182080e-02	-3.322063e-01
PC 5	-1.204609e-01	8.609611e-02	4.468630e-02	-1.151444e-01
PC 6	1.854941e-01	-1.219366e-01	-2.712899e-01	1.865363e-01
PC 7	3.658389e-02	-3.931296e-02	3.859022e-01	-2.053185e-01
PC 8	2.611201e-01	-2.898741e-02	-1.031462e-01	-1.963466e-02
PC 9	2.468152e-01	-8.703385e-02	1.072174e-01	-3.518514e-01

PC 10	-9.360799e-02	8.300941e-02	-3.651395e-01	6.650069e-02
PC 11	5.639006e-02	5.873331e-03	5.897133e-01	8.738808e-02
PC 12	-6.182622e-03	-6.054047e-02	4.329066e-01	1.964040e-01
PC 13	2.906197e-02	-1.318802e-01	-5.527828e-02	-3.099428e-01
PC 14	-5.113526e-02	6.285944e-02	9.909552e-02	6.412800e-01
PC 15	-9.750080e-02	-6.967416e-02	-7.657383e-02	-1.747423e-01
PC 16	2.408675e-01	2.463917e-02	3.324639e-02	-1.764220e-01
PC 17	8.614895e-02	-8.819341e-02	-1.430389e-02	-7.446172e-03
PC 18	5.344134e-02	-2.620057e-02	-1.079876e-01	-1.512095e-02
PC 19	-2.397884e-01	-2.781091e-01	1.295969e-01	-2.891532e-02
PC 20	2.290229e-01	2.830417e-01	9.963455e-02	-4.255684e-02
PC 21	5.215051e-01	5.956328e-01	-1.607765e-03	5.761641e-02
PC 22	3.020415e-02	4.832594e-02	8.798226e-02	7.589338e-02
PC 23	-1.657634e-03	-8.784987e-03	3.118668e-04	4.842437e-04
PC 24	8.867958e-18	9.410034e-18	-9.152804e-17	7.259716e-17
PC 25	1.910812e-17	-6.943436e-17	-1.220531e-17	1.370228e-16
PC 26	-8.491613e-17	-5.912065e-17	-1.808415e-18	8.152361e-17
PC 27	3.956364e-17	9.577648e-17	1.593810e-16	-1.356799e-17
PC 28	7.311588e-17	6.203100e-17	7.963310e-17	8.847175e-17

[29 rows x 29 columns]

```
[ ]: dfLocation.columns
```

```
[ ]: Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
          'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
          'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
          'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
          'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
          'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
          'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
          'TBF_Equipment'],
          dtype='object')
```

```
[ ]: maxPCvars = [
    ↪ ["TBF_Equipment", "ActualDuration", "TotalCost", "IsAffectingProduction"]
```

```
[ ]: commonPCs = []
for i in range(0, len(maxPCvars)):
    name = maxPCvars[i]
    maxPCAit = getMaxComponent(pca_table, name)
    commonPCs.append(maxPCAit)

dictPCs = dict(zip(maxPCvars, commonPCs))

print(dictPCs)
```

```
{'TBF_Equipment': [1, 2, 3, 6, 8, 10, 11, 12, 14, 17, 18, 19, 20, 21, 22, 23,
```

```
24, 25, 26, 27, 28], 'ActualDuration': [21], 'TotalCost': [1, 23],
'IsAffectingProduction': [0, 2, 3, 6, 9, 15, 16, 19, 20]}
```

```
[ ]: import heapq

for i in range(0,len(dictPCs)):
    name = maxPCvars[i]
    print(name)
    PC_num = dictPCs[name]
    print(PC_num)
    for j in range(0,len(PC_num)):
        print(PC_num[j])
        PCval = pca_table.iloc[j]
        mean = PCval.mean()
        largest = heapq.nlargest(3, enumerate(PCval), key=lambda x: x[1])
        lowest = heapq.nsmallest(3, enumerate(PCval), key=lambda x: x[1])
        print("---Highest---")
        for k in range(0,len(largest)):
            indexTabla = largest[k][0]
            print(pca_table.columns[indexTabla])
        print("---Lowest---")
        for k in range(0,len(lowest)):
            indexTabla = lowest[k][0]
            print(pca_table.columns[indexTabla])
```

```
TBF_Equipment
[1, 2, 3, 6, 8, 10, 11, 12, 14, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
1
---Highest---
ScheduleCompliant
GrossProductionLoss
IsAffectingProduction
---Lowest---
FieldProductionTeam
Assigned
ActualDuration
2
---Highest---
MaterialCost
TotalCost
Duration
---Lowest---
IsAffectingProduction
ScheduleCompliant
ReportMonth
3
---Highest---
AffectedProduction
```

GrossProductionLoss
 TradeGroup
 ---Lowest---
 Duration
 ScheduleCompliant
 Priority
 6
 ---Highest---
 ActualDuration
 Priority
 GrossProductionLoss
 ---Lowest---
 ScheduleCompliant
 SupervisorRole
 Trade
 8
 ---Highest---
 AffectedProduction
 GrossProductionLoss
 Assigned
 ---Lowest---
 FailureReason
 TBF_Equipment
 Cause
 10
 ---Highest---
 Priority
 Cause
 FailureReason
 ---Lowest---
 ReportWeekDay
 IsAffectingProduction
 Trade
 11
 ---Highest---
 Safety
 SupervisorRole
 FailureReason
 ---Lowest---
 Trade
 TradeGroup
 TBF
 12
 ---Highest---
 ReportWeekDay
 EquipmentCode
 TBF
 ---Lowest---

SupervisorRole
 TBF_Equipment
 Trade
 14
 ---Highest---
 ReportMonth
 Reopened
 EquipmentCode
 ---Lowest---
 StatusCode
 FieldProductionTeam
 Safety
 17
 ---Highest---
 StatusCode
 ReportMonth
 ReportWeekDay
 ---Lowest---
 TBF_Equipment
 EquipmentCode
 Safety
 18
 ---Highest---
 StatusCode
 Reopened
 FieldProductionTeam
 ---Lowest---
 TBF
 SupervisorRole
 FailureReason
 19
 ---Highest---
 TBF
 Reopened
 FieldProductionTeam
 ---Lowest---
 EquipmentCode
 Priority
 Duration
 20
 ---Highest---
 StatusCode
 TBF
 SupervisorRole
 ---Lowest---
 ReportWeekDay
 FieldProductionTeam
 FailureReason

21
 ---Highest---
 Reopened
 Safety
 Priority
 ---Lowest---
 ReportMonth
 TBF_Equipment
 FieldProductionTeam
 22
 ---Highest---
 TBF_Equipment
 ReportWeekDay
 Priority
 ---Lowest---
 EquipmentCode
 FailureReason
 StatusCode
 23
 ---Highest---
 Assigned
 ReportWeekDay
 Reopened
 ---Lowest---
 Safety
 ReportMonth
 Duration
 24
 ---Highest---
 Cause
 IsAffectingProduction
 Assigned
 ---Lowest---
 FailureReason
 SupervisorRole
 TBF_Equipment
 25
 ---Highest---
 Cause
 ReportWeekDay
 SupervisorRole
 ---Lowest---
 IsAffectingProduction
 Assigned
 FieldProductionTeam
 26
 ---Highest---
 Duration

FailureReason
 FieldProductionTeam
 ---Lowest---
 Priority
 EquipmentCode
 TotalCost
 27
 ---Highest---
 FieldProductionTeam
 IsAffectingProduction
 SupervisorRole
 ---Lowest---
 Assigned
 ScheduleCompliant
 ActualDuration
 28
 ---Highest---
 Trade
 ScheduleCompliant
 AffectedProduction
 ---Lowest---
 TradeGroup
 GrossProductionLoss
 ReportMonth
 ActualDuration
 [21]
 21
 ---Highest---
 ScheduleCompliant
 GrossProductionLoss
 IsAffectingProduction
 ---Lowest---
 FieldProductionTeam
 Assigned
 ActualDuration
 TotalCost
 [1, 23]
 1
 ---Highest---
 ScheduleCompliant
 GrossProductionLoss
 IsAffectingProduction
 ---Lowest---
 FieldProductionTeam
 Assigned
 ActualDuration
 23
 ---Highest---

MaterialCost
 TotalCost
 Duration
 ---Lowest---
 IsAffectingProduction
 ScheduleCompliant
 ReportMonth
 IsAffectingProduction
 [0, 2, 3, 6, 9, 15, 16, 19, 20]
 0
 ---Highest---
 ScheduleCompliant
 GrossProductionLoss
 IsAffectingProduction
 ---Lowest---
 FieldProductionTeam
 Assigned
 ActualDuration
 2
 ---Highest---
 MaterialCost
 TotalCost
 Duration
 ---Lowest---
 IsAffectingProduction
 ScheduleCompliant
 ReportMonth
 3
 ---Highest---
 AffectedProduction
 GrossProductionLoss
 TradeGroup
 ---Lowest---
 Duration
 ScheduleCompliant
 Priority
 6
 ---Highest---
 ActualDuration
 Priority
 GrossProductionLoss
 ---Lowest---
 ScheduleCompliant
 SupervisorRole
 Trade
 9
 ---Highest---
 AffectedProduction

GrossProductionLoss
 Assigned
 ---Lowest---
 FailureReason
 TBF_Equipment
 Cause
 15
 ---Highest---
 Priority
 Cause
 FailureReason
 ---Lowest---
 ReportWeekDay
 IsAffectingProduction
 Trade
 16
 ---Highest---
 Safety
 SupervisorRole
 FailureReason
 ---Lowest---
 Trade
 TradeGroup
 TBF
 19
 ---Highest---
 ReportWeekDay
 EquipmentCode
 TBF
 ---Lowest---
 SupervisorRole
 TBF_Equipment
 Trade
 20
 ---Highest---
 ReportMonth
 Reopened
 EquipmentCode
 ---Lowest---
 StatusCode
 FieldProductionTeam
 Safety

2.1 Regresion

```
[ ]: column_names = dfLocation_scaled.columns
      print(column_names)
```

```
Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
      'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
      'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
      'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
      'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
      'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
      'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
      'TBF_Equipment'],
      dtype='object')
```

```
[ ]: ## Target ##
      y = dfLocation_scaled['TotalCost']
      y.head()
```

```
[ ]: 0    0.000000
      1    0.008593
      2    0.000570
      3    0.000000
      4    0.004219
      Name: TotalCost, dtype: float64
```

```
[ ]: ## Rest of Variables ##
      x = dfLocation_scaled[column_names.drop(['TotalCost'])]
      x.head()
```

```
[ ]: FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
0          0.611111      0.261339          0.0          0.0
1          0.333333      0.933045          0.0          0.0
2          0.222222      0.127430          0.0          0.0
3          0.611111      0.572354          0.0          0.0
4          0.611111      0.166307          0.0          0.0

      EquipmentCriticality  StatusCode  Priority    Cause  FailureReason  \
0          0.0          0.0  0.666667  0.742857      0.000000
1          0.0          0.0  0.666667  0.657143      0.666667
2          0.0          0.0  1.000000  0.685714      1.000000
3          0.0          0.0  0.666667  0.685714      1.000000
4          0.0          0.0  0.666667  0.685714      1.000000

      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth  \
0          0.0  ...          0.0    0.0    0.0    0.0    0.000000
1          0.0  ...          0.0    0.0    0.0    0.0    1.000000
2          0.0  ...          0.0    0.0    0.0    0.0    0.909091
```

3	0.0	...	0.0	0.0	0.0	0.0	0.909091
4	0.0	...	0.0	0.0	0.0	0.0	0.000000

	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
0	1.000000	0.006211	0.0	0.0	0.158238
1	0.166667	0.090062	0.0	0.0	0.167606
2	0.333333	0.155280	0.0	0.0	0.158238
3	0.833333	0.189441	0.0	0.0	0.040845
4	0.666667	0.003106	0.0	0.0	0.158238

[5 rows x 28 columns]

```
[ ]: #Separate train and test data
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
↳20,random_state=0)
```

```
[ ]: print("Size of the full data set: ",x.shape)
print("Size of the training data set: ",x_train.shape)
print("Size of the test data set: ",x_test.shape)
```

```
Size of the full data set: (659, 28)
Size of the training data set: (527, 28)
Size of the test data set: (132, 28)
```

```
[ ]: EquipmentCode = dfLocation_scaled["EquipmentCode"]
EquipmentCode.head()
```

```
[ ]: 0    0.261339
1    0.933045
2    0.127430
3    0.572354
4    0.166307
Name: EquipmentCode, dtype: float64
```

```
[ ]: TotalCost_real = dfLocation_scaled["TotalCost"]
TotalCost_real.head()
```

```
[ ]: 0    0.000000
1    0.008593
2    0.000570
3    0.000000
4    0.004219
Name: TotalCost, dtype: float64
```

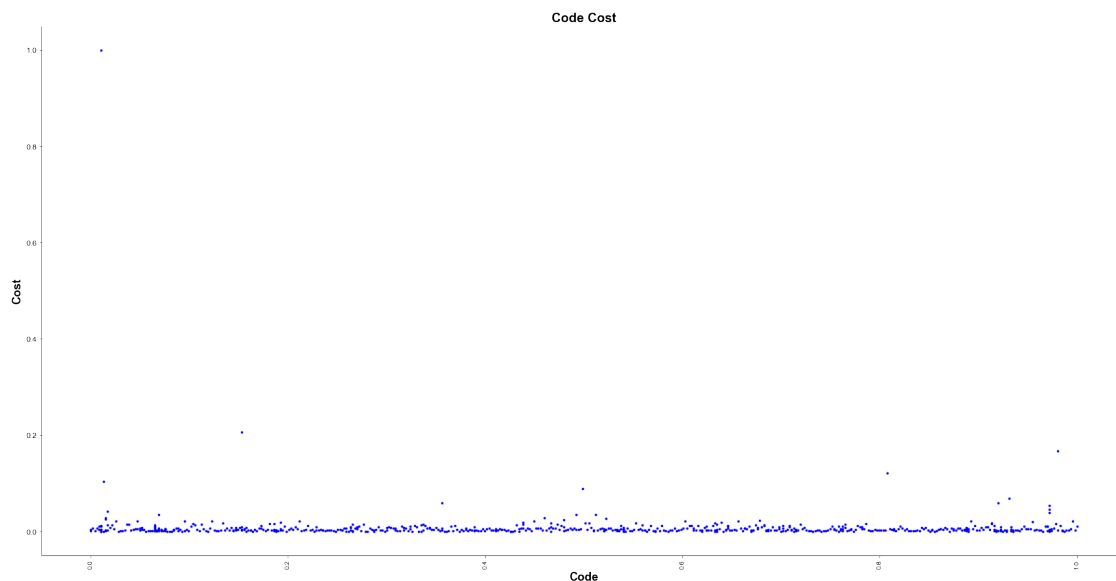
```
[ ]: df_code_cost = pd.DataFrame()

df_code_cost["Code"] = EquipmentCode
df_code_cost["Cost"] = TotalCost_real
```

```
df_code_cost.head()
```

```
[ ]:      Code      Cost
0  0.261339  0.000000
1  0.933045  0.008593
2  0.127430  0.000570
3  0.572354  0.000000
4  0.166307  0.004219
```

```
[ ]: ## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c =_
↪"blue" , s = 30)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```



```
[ ]: # importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
# fitting the training data
LR.fit(x_train,y_train)
```

```
[ ]: LinearRegression()
```

```
[ ]: y_prediction = LR.predict(x_test)
      y_prediction
```

```
[ ]: array([ 7.35838976e-03,  3.18180456e-03, -5.74579708e-05,  4.07591968e-03,
            6.48858869e-03,  1.62152433e-03,  1.64411073e-02,  4.16127997e-03,
            1.87127871e-03,  3.26409661e-03,  4.07152963e-03,  1.08184138e-02,
            3.80533708e-03,  7.52713281e-03,  3.43576594e-03,  3.93809519e-03,
            5.98780567e-03,  6.30786047e-03,  1.37090318e-03,  1.66491872e-02,
            4.03010085e-03,  8.52262587e-03,  1.35914890e-02,  4.37761333e-03,
            1.18447007e-02,  3.65751051e-03,  2.75579063e-03,  4.63884775e-03,
            9.79844605e-04,  3.39984601e-03,  7.34024010e-03,  3.77333998e-02,
            4.33414953e-03,  2.10224360e-03,  1.95703834e-02,  4.60269030e-03,
            4.13955670e-03,  5.80644873e-03,  3.43304315e-03,  6.93124394e-04,
            1.04796807e-03,  1.47732195e-03,  3.76024780e-03,  4.38069212e-03,
            5.00636699e-03,  5.87804867e-03,  1.79582539e-03,  2.60807074e-03,
            3.94627062e-03,  2.81751692e-03,  7.56847036e-03,  6.35454774e-03,
            5.55334734e-03,  5.93168263e-03,  1.88512164e-03,  2.82569562e-03,
            1.53267494e-03,  1.24567897e-03,  6.34738056e-03,  4.30527254e-03,
            8.66018126e-03,  4.04113258e-03,  4.97104967e-04,  2.66166654e-03,
            1.65006508e-03,  3.33269159e-03,  7.73363959e-03,  1.70601950e-03,
            4.25634648e-03,  8.51964796e-04,  2.58011333e-03,  1.50875763e-03,
            5.63270340e-03,  5.55407635e-03,  1.34847929e-02,  1.55209825e-02,
            3.96546614e-03,  2.82394950e-03,  2.12168499e-03,  2.21652838e-03,
            1.20985002e-02,  9.87038843e-03,  2.26719955e-03,  3.82798914e-03,
            9.82876044e-03,  3.58927062e-03,  6.96359450e-02,  2.64609117e-03,
            -3.44365222e-04,  4.06872713e-03,  2.22123127e-03,  2.13126443e-03,
            3.11693129e-03,  4.98802113e-03,  2.61918800e-03,  4.31897226e-03,
            6.79987745e-03,  8.60201766e-03,  6.52750638e-03,  5.96704598e-03,
            7.08183154e-03,  6.06652947e-03,  1.10409171e-02,  4.74535880e-03,
            5.36766668e-03,  7.31794320e-03,  6.31058785e-04,  8.00216317e-03,
            3.47966052e-03,  1.53269523e-03,  1.19769584e-02,  1.31403956e-02,
            1.17899771e-02,  4.86876053e-03,  1.52129496e-03,  5.84434477e-03,
            3.45095550e-03,  3.41690035e-03,  4.78615587e-03,  5.96800298e-03,
            1.33732464e-02,  3.97001554e-03,  1.97958999e-03,  5.14168483e-03,
            1.93007897e-02,  2.63078819e-03,  6.44324610e-03,  1.69480929e-02,
            4.83485466e-03,  8.53394878e-03,  5.22684893e-03,  5.95569309e-03])
```

```
[ ]: # importing r2_score module
      from sklearn.metrics import r2_score
      from sklearn.metrics import mean_squared_error
      from sklearn.metrics import mean_absolute_error

      score = r2_score(y_test,y_prediction)
      print("R2 score is ",score)
```

```

print("Mean squared error is ",mean_squared_error(y_test,y_prediction))
print("Mean absolute error is ",mean_absolute_error(y_test,y_prediction))
print("Root mean squared error is ",np.
↳sqrt(mean_squared_error(y_test,y_prediction)))

```

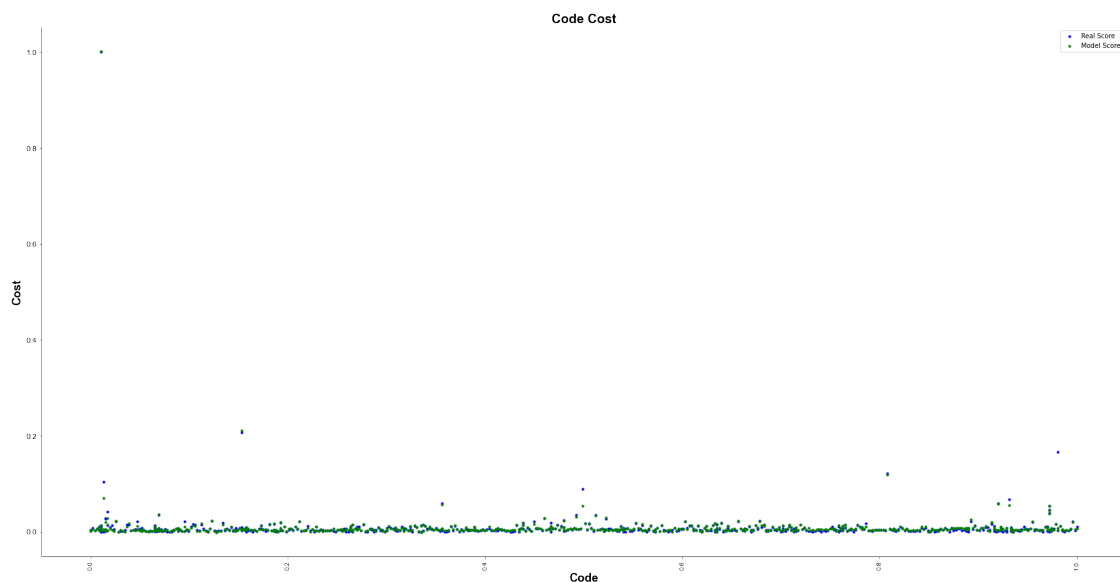
R2 score is 0.8438391640030621
Mean squared error is 1.6285399022356986e-05
Mean absolute error is 0.0022704180781530268
Root mean squared error is 0.004035517193911703

```

[ ]: y_prediction = LR.predict(x)

## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c =_
↳"blue" , s = 40)
plt.scatter(x = df_code_cost["Code"],y = y_prediction, marker=".", c = "green"_
↳, s = 40)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.legend(['Real Score','Model Score'])
plt.show()

```



3 Asset

```
[ ]: ## Asset Dataframe ##
dfAsset = df.loc[df["EquipmentType"] == 'Asset']
dfAsset.head()
```

```
[ ]:      WorkOrder FieldProductionTeam EquipmentCode EquipmentType \
2      15476748          VAN      22B0359734      Asset
6      15474454          VAN      22B0359765      Asset
10     15474690      CAYUGA      70A0052951      Asset
16     15506226      BUCKEYE      22B0360884      Asset
17     15444847          VAN      22B0359543      Asset

      EquipmentClass EquipmentCriticality StatusCode Priority \
2      Pumping Unit          3 - Low      C      3
6      Pumping Unit          3 - Low      C      3
10     Vessel - Separator      3 - Low      C      3
16     Pumping Unit          3 - Low      C      3
17     Pumping Unit          3 - Low      C      3

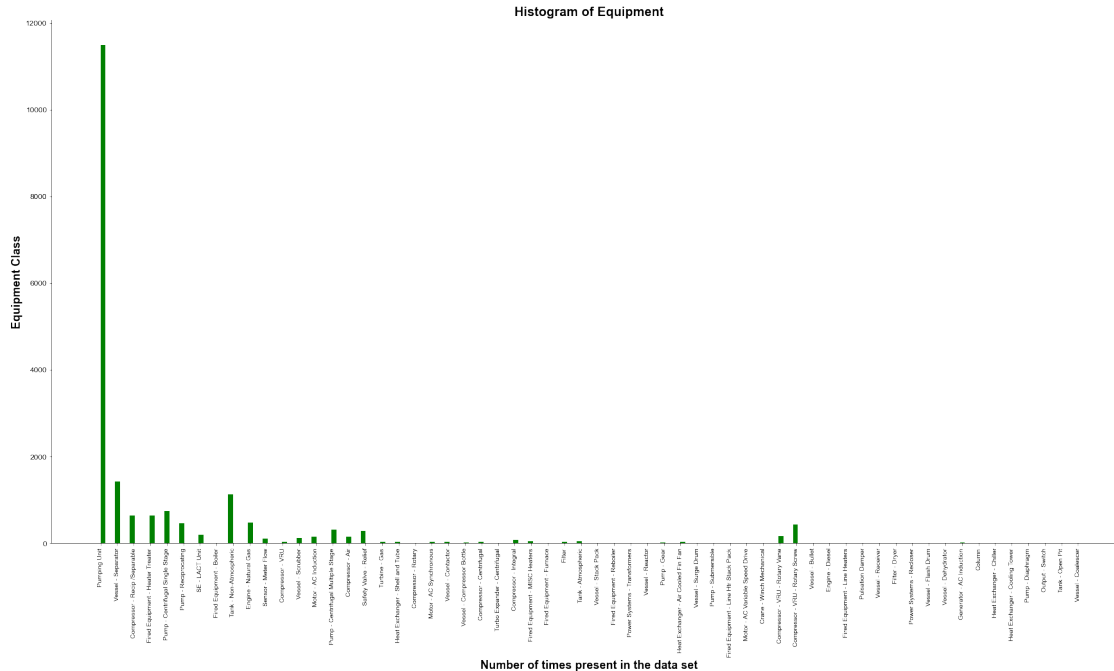
      Cause      FailureReason ...      Manufacturer \
2      Loose      Component - Belt ...      LUFKIN
6      Missing      POC - Cell, Load ...      LUFKIN
10     Flow Cut / Erosion / Wear      Pipe - Piping ...      MIDESSAEQUIPMENT
16      Loose      Component - Belt ...      AMERICAN
17     Open / Short / Ground      Assembly - Panel ...      LUFKIN

      Model Safety Reopened ReportMonth ReportWeekDay ActualDuration \
2      M320D-213-120      N      N      12      6      19
6      M228D-256-100      N      N      12      5      20
10      NONE      N      N      12      5      20
16      NONE      N      N      12      2      4
17     C-114D-119-86      N      N      11      1      47

      ScheduleCompliant TBF TBF_Equipment
2      Future Week      0      1.0
6      Future Week      0      1.0
10     Future Week      0      NaN
16     Future Week      0     139.0
17     Future Week      0     13.0
```

[5 rows x 30 columns]

```
[ ]: createGraphs(dfAsset)
```



```
[ ]: dfAssetCommonValues = getCommonValues(dfAsset,500)
print(dfAssetCommonValues)
```

```
['Pumping Unit', 'Vessel - Separator', 'Compressor - Recip /Separable', 'Fired
Equipment - Heater Treater', 'Pump - Centrifugal Single Stage', 'Tank - Non-
Atmospheric']
```

```
[ ]: dfAsset = dfAsset.loc[dfAsset["EquipmentClass"].isin(dfAssetCommonValues)]
dfAsset.head()
```

```
[ ]:      WorkOrder FieldProductionTeam EquipmentCode EquipmentType \
2      15476748          VAN      22B0359734      Asset
6      15474454          VAN      22B0359765      Asset
10     15474690      CAYUGA      70A0052951      Asset
16     15506226      BUCKEYE      22B0360884      Asset
17     15444847          VAN      22B0359543      Asset
```

```
      EquipmentClass EquipmentCriticality StatusCode Priority \
2      Pumping Unit      3 - Low      C      3
6      Pumping Unit      3 - Low      C      3
10     Vessel - Separator      3 - Low      C      3
16     Pumping Unit      3 - Low      C      3
17     Pumping Unit      3 - Low      C      3
```

```
      Cause      FailureReason ...      Manufacturer \
2      Loose      Component - Belt ...      LUFKIN
```

```

6          Missing POC - Cell, Load ...          LUFKIN
10 Flow Cut / Erosion / Wear      Pipe - Piping ... MIDESSAEQUIPMENT
16          Loose Component - Belt ...          AMERICAN
17      Open / Short / Ground Assembly - Panel ...          LUFKIN

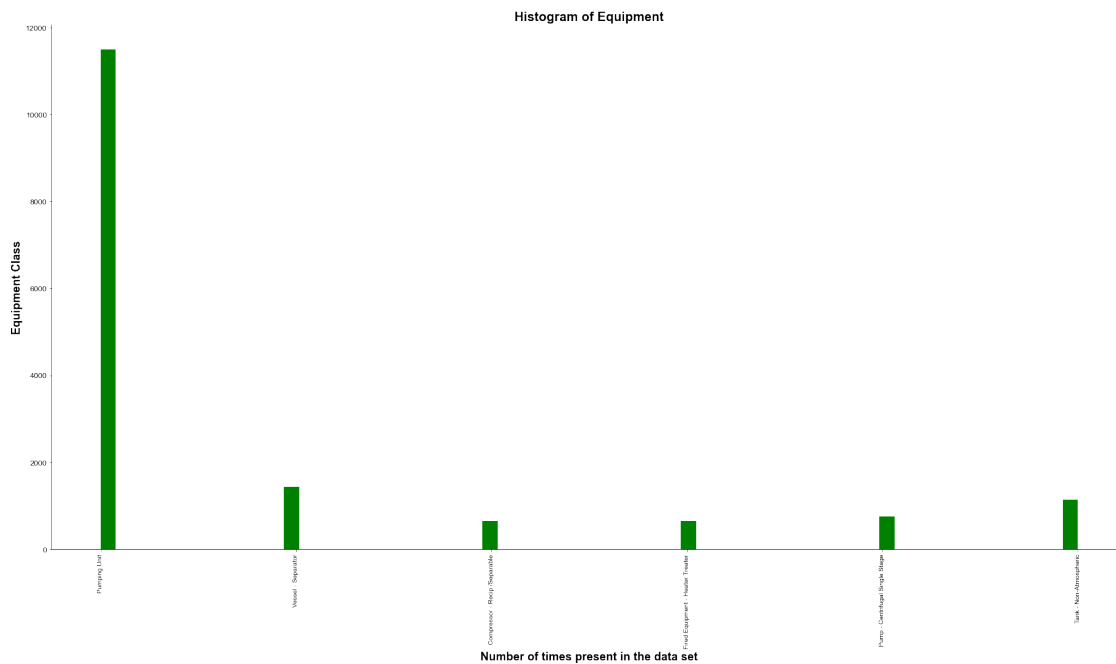
```

	Model	Safety	Reopened	ReportMonth	ReportWeekDay	ActualDuration \
2	M320D-213-120	N	N	12	6	19
6	M228D-256-100	N	N	12	5	20
10	NONE	N	N	12	5	20
16	NONE	N	N	12	2	4
17	C-114D-119-86	N	N	11	1	47

	ScheduleCompliant	TBF	TBF_Equipment
2	Future Week	0	1.0
6	Future Week	0	1.0
10	Future Week	0	NaN
16	Future Week	0	139.0
17	Future Week	0	13.0

[5 rows x 30 columns]

```
[ ]: createGraphs(dfAsset)
```



```
[ ]: ## Now that the data has been narrowed to the most common case we will encode
      ↳ it make some changes to it ##
```



```
dfAsset= EncodeData(dfAsset)
dfAsset.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
WorkOrder
15476748                27            1958                0                3
15474454                27            1975                0                3
15474690                 7            3914                0                5
15506226                 2            2317                0                3
15444847                27            1889                0                3
```

```
      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason  \
WorkOrder
15476748                 1           0         3    23           26
15474454                 1           0         3    25          169
15474690                 1           0         3    12          177
15506226                 1           0         3    23           26
15444847                 1           0         3    30           6
```

```
      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth  \
WorkOrder      ...
15476748        1  ...           99    945        0          0           12
15474454        1  ...           99    939        0          0           12
15474690        1  ...          110    989        0          0           12
15506226        1  ...            3    989        0          0           12
15444847        1  ...           99    544        0          0           11
```

```
      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  \
WorkOrder
15476748           6              19                   0    0
15474454           5              20                   0    0
15474690           5              20                   0    0
15506226           2               4                   0    0
15444847           1              47                   0    0
```

```
      TBF_Equipment
WorkOrder
15476748        1.0
15474454        1.0
15474690        NaN
15506226       139.0
15444847       13.0
```

[5 rows x 29 columns]

```
[ ]: ## We need to handle the datasets NA Values ##
dfAsset = dfAsset.fillna(dfAsset.mean())
```

```
dfAsset.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass \
WorkOrder
15476748                27            1958                0                3
15474454                27            1975                0                3
15474690                 7            3914                0                5
15506226                 2            2317                0                3
15444847                27            1889                0                3
```

```
      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason \
WorkOrder
15476748                 1           0         3    23           26
15474454                 1           0         3    25          169
15474690                 1           0         3    12          177
15506226                 1           0         3    23           26
15444847                 1           0         3    30           6
```

```
      Duration  ...  Manufacturer  Model  Safety  Reopened  ReportMonth \
WorkOrder      ...
15476748      1  ...           99    945        0          0           12
15474454      1  ...           99    939        0          0           12
15474690      1  ...          110    989        0          0           12
15506226      1  ...            3    989        0          0           12
15444847      1  ...           99    544        0          0           11
```

```
      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF \
WorkOrder
15476748           6              19                   0    0
15474454           5              20                   0    0
15474690           5              20                   0    0
15506226           2               4                   0    0
15444847           1              47                   0    0
```

```
      TBF_Equipment
WorkOrder
15476748      1.000000
15474454      1.000000
15474690     175.952588
15506226     139.000000
15444847     13.000000
```

```
[5 rows x 29 columns]
```

```
[ ]: ## For further analysis we know need to standarize the dataset##
## Scales ##
scaling_procedure_1 = MinMaxScaler(feature_range= (0,1))
```

```
[ ]: ## Scaled Data ##
columnSystem = dfAsset.columns
dfAsset_scaled = scaling_procedure_1.fit_transform(dfAsset)
dfAsset_scaled = pd.DataFrame(dfAsset_scaled, columns = columnSystem)
dfAsset_scaled.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
0                0.870968      0.316572            0.0            0.6
1                0.870968      0.319321            0.0            0.6
2                0.225806      0.632821            0.0            1.0
3                0.064516      0.374616            0.0            0.6
4                0.870968      0.305416            0.0            0.6

      EquipmentCriticality  StatusCode  Priority    Cause  FailureReason  \
0                0.2            0.0        0.6  0.500000      0.117117
1                0.2            0.0        0.6  0.543478      0.761261
2                0.2            0.0        0.6  0.260870      0.797297
3                0.2            0.0        0.6  0.500000      0.117117
4                0.2            0.0        0.6  0.652174      0.027027

      Duration  ...  Manufacturer      Model  Safety  Reopened  ReportMonth  \
0          0.0  ...      0.487685  0.834069      0.0        0.0      1.000000
1          0.0  ...      0.487685  0.828773      0.0        0.0      1.000000
2          0.0  ...      0.541872  0.872904      0.0        0.0      1.000000
3          0.0  ...      0.014778  0.872904      0.0        0.0      1.000000
4          0.0  ...      0.487685  0.480141      0.0        0.0      0.909091

      ReportWeekDay  ActualDuration  ScheduleCompliant  TBF  TBF_Equipment
0          0.833333          0.016323                0.0  0.0      0.000798
1          0.666667          0.017182                0.0  0.0      0.000798
2          0.666667          0.017182                0.0  0.0      0.140425
3          0.166667          0.003436                0.0  0.0      0.110934
4          0.000000          0.040378                0.0  0.0      0.010375
```

[5 rows x 29 columns]

```
[ ]: ##Here is the correlation plot of the system##

corr_df = dfAsset_scaled

corr = dfAsset_scaled.corr()
corr.style.background_gradient(cmap="coolwarm")

plt.figure(figsize=(70, 10))
heatmap = sns.heatmap(corr, vmin=-1, vmax=1, annot=True)
```



```
[ ]: ## PCA ##

Xdat = dfAsset_scaled

header_row= dfAsset_scaled.columns.values

y=dfAsset_scaled["TBF"]
```

```
[ ]: # create the PCA instance
pca = PCA()

# fit on data
pca = pca.fit(Xdat)

#Q eigenvectors
Q=pca.components_.T

#Explained deviations
s= np.sqrt(pca.explained_variance_)
```

```
[ ]: # transform data obtain sample scores
F = pca.transform(Xdat)
print("Sample scores")

## Sample Scores ##
sample_scores = pd.DataFrame(F)
print(sample_scores)
```

```
16095  0.815490  0.125163 -0.367066 -0.251980 -0.360783 -0.066025 -0.442516
16096  0.323908 -0.694227 -0.110502 -0.142589  0.011533  0.354593 -0.160829
```

```

          7          8          9  ...         19         20         21  \
0      -0.263521 -0.262175  0.229257 ... -0.073454  0.004382 -0.009475
1      -0.213078 -0.187691 -0.053253 ...  0.050396  0.008005 -0.009910
2       0.195772 -0.040072 -0.080253 ... -0.058988 -0.002405 -0.010851
3       0.282227 -0.356384  0.338909 ...  0.065671 -0.002157 -0.009069
4      -0.337184  0.011038  0.454801 ...  0.053979  0.006125 -0.008858
...      ...      ...      ...      ...      ...      ...
16092  0.554541  0.095903 -0.177251 ...  0.054665 -0.012702 -0.002579
16093 -0.348881  0.346907  0.114438 ... -0.063602 -0.016356 -0.001001
16094  0.084357 -0.248436  0.351796 ... -0.050907 -0.018472 -0.000707
16095  0.090978  0.216175 -0.397376 ... -0.027653 -0.012536 -0.002230
16096 -0.449203  0.130596 -0.330312 ... -0.041910 -0.009992 -0.001370
```

```

          22          23          24          25          26          27  \
0      -0.005856 -0.007531 -0.002875 -0.010732 -0.000574 -0.000228
1      -0.001261 -0.005180  0.001325 -0.009021 -0.000494 -0.000168
2      -0.005776 -0.010619 -0.002663 -0.009169  0.000605 -0.000179
3       0.003521 -0.007656  0.007236 -0.022581 -0.001393  0.000106
4      -0.000519  0.002109  0.000983  0.014455 -0.000830 -0.000125
...      ...      ...      ...      ...      ...
16092  0.000265 -0.001002 -0.000082 -0.001513  0.000719 -0.000191
16093 -0.006832  0.000148  0.007571  0.000601 -0.001158 -0.000447
16094 -0.007387 -0.000056 -0.000759 -0.001573 -0.002384 -0.000176
16095 -0.007638  0.000878 -0.005345  0.000831 -0.000044 -0.000105
16096 -0.008349  0.001298  0.000573  0.000588  0.000871 -0.000128
```

```

          28
0      -7.050181e-16
1      -4.829847e-16
2      -5.218356e-16
3      -2.759456e-16
4      -1.834749e-16
...      ...
16092  1.466917e-16
16093  5.686165e-16
16094  6.029674e-16
16095  2.115796e-16
16096  2.865118e-16
```

```
[16097 rows x 29 columns]
```

```
[ ]: # Obtain Factor Loadings
L = np.zeros(shape=(Q.shape))
for col in range(0, len(s)-1):
```

```

    L[:,col] = Q[:,col] * s[col]
print("Factor Loadings")

columns_factorLoading = []

for i in range(0, 29):
    nombre = "F" + str(i)
    columns_factorLoading.append(nombre)

## Factor Loadings Dataframe ##
factor_loadings = pd.DataFrame(L, columns= columns_factorLoading)
print(factor_loadings)

```

Factor Loadings

	F0	F1	F2	F3	F4	\
0	-0.012859	-6.272346e-03	-4.913279e-03	4.551945e-03	5.981462e-02	
1	-0.144821	8.233193e-03	1.601010e-02	-3.709681e-02	-1.027963e-01	
2	0.000000	-3.802925e-17	1.403518e-17	1.284469e-16	3.053408e-17	
3	-0.012897	9.644268e-03	1.026988e-02	-2.466773e-02	-4.888540e-02	
4	0.038387	-1.821437e-01	-3.267866e-01	-7.879868e-02	-8.074340e-03	
5	0.003493	-8.275421e-03	-1.734008e-02	-1.588787e-02	3.795335e-03	
6	-0.009995	1.639404e-02	-9.296621e-03	3.410619e-03	1.239219e-02	
7	-0.016518	1.891252e-02	-2.634337e-02	-6.270125e-02	-9.363622e-02	
8	-0.143883	5.404880e-04	2.097100e-02	-5.598235e-02	-1.680342e-01	
9	-0.000781	4.720578e-04	1.085637e-03	-4.163218e-05	-2.728690e-04	
10	0.002294	-2.429040e-03	2.776667e-04	2.290097e-04	-1.297085e-04	
11	0.001962	-2.049832e-03	1.808298e-03	-4.742601e-04	-1.614305e-03	
12	0.419624	-2.136143e-01	1.292226e-01	-2.204171e-02	-8.400829e-02	
13	-0.000107	-8.674061e-06	4.168253e-05	6.109216e-05	3.461186e-05	
14	-0.000167	-5.329262e-05	5.897221e-05	6.130755e-05	3.255672e-05	
15	-0.067232	-1.121663e-01	-1.970564e-02	-2.357975e-02	1.763404e-02	
16	-0.077401	-1.140176e-01	5.017867e-02	2.257551e-02	6.330265e-02	
17	-0.044021	-7.970812e-02	3.739049e-02	1.484449e-02	4.486489e-02	
18	-0.262008	-3.055710e-01	1.106568e-01	2.388598e-02	1.613302e-02	
19	-0.060718	3.340075e-03	9.685894e-03	-3.927430e-02	-1.192521e-01	
20	-0.022809	2.207692e-02	5.303241e-02	-6.720080e-03	-6.675676e-02	
21	-0.007807	-4.704150e-03	-1.027844e-02	5.065947e-04	-7.046826e-03	
22	0.001022	-3.681237e-03	-8.997127e-04	-5.643664e-04	4.937221e-04	
23	0.006124	-2.310620e-02	-7.626640e-02	2.779352e-01	-1.061255e-01	
24	-0.014202	4.077701e-03	-9.761532e-05	6.644669e-03	3.198799e-02	
25	-0.000970	1.271690e-03	2.394931e-03	-1.163102e-04	2.229615e-04	
26	0.011497	-4.751697e-02	-1.116475e-01	5.059004e-02	-1.884368e-02	
27	-0.000355	1.523609e-03	1.486924e-03	1.979125e-04	-4.064370e-04	
28	-0.003875	1.488531e-02	1.997470e-02	1.936466e-03	1.877160e-03	
	F5	F6	F7	F8	F9	...
0	1.245807e-01	4.575393e-02	-1.514744e-01	-2.255141e-03	-1.035252e-01	...

1	-3.541107e-02	-3.347576e-03	-1.958129e-02	1.083944e-01	2.283960e-02	...
2	-4.942823e-17	-1.634022e-17	2.263965e-17	3.770517e-17	-4.059244e-17	...
3	-6.632864e-03	4.036577e-03	-1.418802e-02	2.174663e-02	-4.782028e-02	...
4	-5.550394e-02	-1.016451e-02	-1.809389e-03	-1.959760e-02	-1.076737e-02	...
5	2.638768e-03	8.779190e-05	-3.456199e-03	2.200893e-03	-3.995914e-03	...
6	3.346178e-03	1.677044e-03	2.411233e-03	-8.091783e-03	-2.637873e-03	...
7	1.455639e-01	5.296259e-02	3.342017e-02	-1.102873e-01	1.174778e-01	...
8	2.084241e-02	2.203010e-02	9.243621e-02	-1.465512e-02	-1.542804e-01	...
9	-5.371258e-04	4.391909e-04	-3.500488e-05	2.560978e-04	-3.737587e-04	...
10	-1.246281e-03	-2.038859e-04	-5.425090e-04	4.923984e-04	-1.515463e-04	...
11	1.708262e-04	2.879359e-04	-3.577344e-04	1.456273e-03	4.069276e-04	...
12	3.656385e-03	1.731621e-02	-3.155786e-04	1.994261e-02	-1.407639e-02	...
13	-1.023548e-04	-1.389198e-04	-2.478534e-04	-1.205479e-06	1.313505e-04	...
14	-1.448646e-04	-1.577380e-04	-2.647627e-04	3.042907e-05	1.413767e-04	...
15	1.396583e-01	4.450058e-02	-6.335477e-02	5.673804e-02	-5.064633e-03	...
16	-6.522328e-02	-1.601746e-02	-1.059098e-02	-2.849513e-02	-4.073993e-02	...
17	-4.982078e-02	-1.400991e-02	-5.990402e-04	-1.418280e-02	-1.383938e-02	...
18	1.188886e-02	-7.850423e-03	2.624930e-02	-2.315516e-02	4.705389e-02	...
19	-7.031523e-02	-4.184360e-03	-1.419451e-01	7.438175e-02	7.429008e-02	...
20	-8.795636e-02	-8.884698e-04	-1.224398e-01	-1.739133e-01	-1.957059e-02	...
21	-8.162892e-03	-2.020175e-03	1.172785e-03	1.318408e-03	6.988873e-04	...
22	-2.785504e-03	-6.624553e-04	8.093631e-04	3.508048e-04	1.043203e-03	...
23	2.223986e-02	1.280956e-02	-2.638568e-03	1.747464e-03	4.572706e-03	...
24	-8.281465e-02	2.795379e-01	1.905535e-02	9.990272e-03	8.436238e-03	...
25	-3.377641e-04	8.150061e-05	6.570265e-04	-2.939510e-04	5.293810e-04	...
26	4.283118e-03	2.266084e-03	-1.956543e-02	-5.416909e-03	-1.963287e-02	...
27	3.724490e-04	3.356349e-04	1.100225e-04	2.741400e-04	9.912566e-05	...
28	5.487307e-03	1.155761e-03	-1.709879e-03	4.880571e-03	-5.460017e-03	...

	F19	F20	F21	F22	F23	\
0	-1.333799e-03	8.301217e-04	2.725634e-07	-1.920691e-05	3.207365e-05	
1	-1.034557e-02	-4.079147e-04	3.041042e-05	-1.906479e-04	-4.872831e-05	
2	-7.367062e-17	1.664155e-17	-3.296153e-18	6.181022e-17	-9.669670e-17	
3	9.529488e-03	7.438913e-05	-6.466375e-06	1.560096e-04	-7.385570e-05	
4	5.117918e-04	-4.975606e-04	1.775916e-04	3.073371e-05	5.345998e-05	
5	-3.219084e-03	-4.133459e-04	9.138251e-05	-3.394165e-05	1.915980e-06	
6	-1.017143e-01	-2.878569e-04	5.436416e-05	7.722300e-04	1.777702e-06	
7	5.418815e-04	2.455128e-04	-1.305104e-05	2.827520e-05	2.261018e-05	
8	-5.638881e-04	3.487189e-04	-5.526997e-05	-9.306975e-06	-1.455241e-05	
9	-4.538387e-06	-4.571534e-05	-1.489268e-04	-2.219663e-04	2.026000e-02	
10	8.157679e-04	4.584633e-04	-3.999922e-05	4.722479e-03	2.481206e-05	
11	3.014319e-03	3.645420e-04	-7.335738e-05	2.328315e-02	1.921325e-04	
12	-6.699188e-03	-3.116977e-04	2.903348e-05	-1.984323e-04	9.970677e-06	
13	1.633487e-04	4.759175e-05	5.834967e-06	6.478386e-05	1.258502e-04	
14	1.724568e-04	9.252543e-05	9.375242e-06	8.132165e-05	1.482896e-04	
15	1.457165e-03	-2.858596e-04	1.880084e-05	-4.240786e-05	-1.723324e-05	
16	-5.327537e-03	-2.749753e-04	3.651894e-05	4.406702e-05	-9.632959e-05	
17	8.854232e-03	-5.951800e-04	1.327542e-04	-1.897520e-05	7.198504e-05	

```

18 -1.326378e-03 -5.292985e-05 9.239479e-06 -2.086661e-05 1.135323e-05
19 8.693743e-05 1.697594e-04 -6.345011e-05 -2.351667e-05 1.009563e-05
20 -6.460802e-04 -1.946476e-04 2.868206e-05 3.540931e-06 -3.084417e-05
21 -8.991096e-03 -2.341384e-04 6.579563e-05 3.937579e-05 2.380817e-05
22 -4.689685e-04 7.265518e-02 -2.827985e-04 -1.458258e-04 1.250790e-05
23 4.869234e-04 7.622252e-05 -1.557187e-05 4.339502e-06 -9.340133e-06
24 7.055316e-04 -4.482600e-05 -3.312011e-05 -1.034088e-05 -3.916456e-05
25 -2.792471e-04 -2.773469e-05 -2.256692e-04 -2.623006e-05 4.261300e-03
26 -8.486790e-04 3.821566e-05 8.786368e-05 -1.021322e-05 1.193634e-04
27 1.736406e-04 4.900303e-04 4.246052e-02 4.221454e-05 9.375808e-05
28 5.423777e-03 -3.292021e-04 -2.420805e-04 4.334677e-05 5.360305e-05

```

	F24	F25	F26	F27	F28
0	9.777034e-06	8.821842e-06	-2.679453e-06	6.021727e-09	0.0
1	2.435927e-05	1.595009e-05	-1.290215e-05	-1.022087e-07	0.0
2	-1.617797e-17	-3.113856e-17	1.741761e-17	1.737081e-18	0.0
3	-5.587535e-06	1.488885e-05	2.871989e-05	9.240389e-08	0.0
4	-4.703472e-05	-3.476081e-06	2.684226e-06	1.500966e-09	0.0
5	1.180935e-05	-1.131426e-05	3.423389e-06	-2.217763e-08	0.0
6	3.008048e-05	-3.690415e-05	2.798014e-05	3.626635e-08	0.0
7	5.084473e-05	-3.992854e-06	-9.275150e-07	1.353297e-08	0.0
8	2.227034e-05	-6.713593e-07	7.502083e-06	-3.950668e-09	0.0
9	1.872205e-06	-2.989539e-03	-1.254238e-04	-3.992584e-07	0.0
10	1.680668e-02	-2.198523e-06	-6.194659e-04	-1.005360e-06	0.0
11	-3.414848e-03	-1.005908e-05	7.094088e-05	-7.975612e-08	0.0
12	-6.821306e-05	1.872657e-05	6.153316e-06	5.470968e-09	0.0
13	6.142906e-04	-7.342717e-05	8.281754e-03	-4.151524e-04	0.0
14	6.630408e-04	-7.433538e-05	8.391209e-03	4.096571e-04	0.0
15	3.627013e-06	1.629197e-05	-3.444109e-06	-2.774634e-09	0.0
16	-3.089001e-05	-1.694269e-05	-1.332112e-05	-8.566885e-08	0.0
17	-1.031311e-04	3.032968e-05	-2.257698e-06	-6.484472e-08	0.0
18	8.366823e-06	-1.312512e-05	5.071209e-06	1.638399e-09	0.0
19	-3.660755e-05	5.783959e-06	-9.122649e-06	-3.585721e-08	0.0
20	-6.639171e-06	-1.083251e-05	-8.384935e-06	1.214743e-08	0.0
21	3.826406e-06	-1.394025e-05	3.850632e-06	1.120605e-08	0.0
22	-9.166774e-05	3.306470e-06	-1.259704e-05	-2.444376e-07	0.0
23	-1.893795e-05	-5.119084e-05	-9.386091e-07	4.298090e-09	0.0
24	-1.140191e-06	-4.301992e-06	7.219364e-06	9.828337e-09	0.0
25	7.305967e-06	1.420686e-02	6.035203e-05	-8.572051e-08	0.0
26	5.896318e-06	3.405529e-04	-7.839581e-08	-1.333799e-08	0.0
27	9.713358e-06	6.450530e-05	-3.701531e-06	-3.763415e-08	0.0
28	1.779652e-05	2.930232e-05	-3.781726e-06	4.108435e-09	0.0

[29 rows x 29 columns]

```

[ ]: #Obtain squared cosines
COS2=L**2

```



```

print("Square Cosines")

## COS2 Dataframe ##
COS2_DF = pd.DataFrame(COS2)
print(COS2_DF)

## Create a table ##

```

Square Cosines

	0	1	2	3	4	\
0	1.653566e-04	3.934233e-05	2.414031e-05	2.072020e-05	3.577789e-03	
1	2.097301e-02	6.778547e-05	2.563232e-04	1.376173e-03	1.056709e-02	
2	0.000000e+00	1.446224e-33	1.969864e-34	1.649860e-32	9.323298e-34	
3	1.663407e-04	9.301190e-05	1.054704e-04	6.084971e-04	2.389782e-03	
4	1.473593e-03	3.317633e-02	1.067895e-01	6.209232e-03	6.519497e-05	
5	1.220337e-05	6.848259e-05	3.006783e-04	2.524244e-04	1.440456e-05	
6	9.989406e-05	2.687645e-04	8.642717e-05	1.163232e-05	1.535663e-04	
7	2.728344e-04	3.576833e-04	6.939732e-04	3.931447e-03	8.767742e-03	
8	2.070221e-02	2.921273e-07	4.397829e-04	3.134024e-03	2.823548e-02	
9	6.098830e-07	2.228385e-07	1.178608e-06	1.733238e-09	7.445752e-08	
10	5.263647e-06	5.900237e-06	7.709878e-08	5.244544e-08	1.682429e-08	
11	3.850484e-06	4.201813e-06	3.269941e-06	2.249226e-07	2.605982e-06	
12	1.760840e-01	4.563108e-02	1.669849e-02	4.858370e-04	7.057392e-03	
13	1.135483e-08	7.523934e-11	1.737434e-09	3.732252e-09	1.197981e-09	
14	2.795813e-08	2.840103e-09	3.477721e-09	3.758615e-09	1.059940e-09	
15	4.520102e-03	1.258128e-02	3.883121e-04	5.560045e-04	3.109594e-04	
16	5.990917e-03	1.300000e-02	2.517899e-03	5.096537e-04	4.007226e-03	
17	1.937860e-03	6.353384e-03	1.398049e-03	2.203588e-04	2.012858e-03	
18	6.864830e-02	9.337362e-02	1.224492e-02	5.705398e-04	2.602744e-04	
19	3.686701e-03	1.115610e-05	9.381654e-05	1.542471e-03	1.422105e-02	
20	5.202473e-04	4.873904e-04	2.812436e-03	4.515947e-05	4.456465e-03	
21	6.094989e-05	2.212903e-05	1.056464e-04	2.566382e-07	4.965776e-05	
22	1.044879e-06	1.355150e-05	8.094830e-07	3.185094e-07	2.437615e-07	
23	3.750679e-05	5.338965e-04	5.816564e-03	7.724797e-02	1.126261e-02	
24	2.017006e-04	1.662764e-05	9.528750e-09	4.415162e-05	1.023231e-03	
25	9.415874e-07	1.617195e-06	5.735696e-06	1.352807e-08	4.971183e-08	
26	1.321812e-04	2.257862e-03	1.246516e-02	2.559353e-03	3.550844e-04	
27	1.261492e-07	2.321385e-06	2.210944e-06	3.916938e-08	1.651910e-07	
28	1.501564e-05	2.215725e-04	3.989887e-04	3.749901e-06	3.523732e-06	

	5	6	7	8	9	...	\
0	1.552035e-02	2.093422e-03	2.294451e-02	5.085661e-06	1.071747e-02	...	
1	1.253944e-03	1.120626e-05	3.834271e-04	1.174935e-02	5.216475e-04	...	
2	2.443150e-33	2.670027e-34	5.125536e-34	1.421680e-33	1.647746e-33	...	
3	4.399488e-05	1.629395e-05	2.012998e-04	4.729159e-04	2.286779e-03	...	
4	3.080688e-03	1.033172e-04	3.273889e-06	3.840659e-04	1.159362e-04	...	
5	6.963094e-06	7.707418e-09	1.194531e-05	4.843932e-06	1.596733e-05	...	
6	1.119691e-05	2.812477e-06	5.814046e-06	6.547695e-05	6.958376e-06	...	

7	2.118886e-02	2.805036e-03	1.116908e-03	1.216328e-02	1.380103e-02	...
8	4.344063e-04	4.853254e-04	8.544452e-03	2.147726e-04	2.380244e-02	...
9	2.885041e-07	1.928887e-07	1.225342e-09	6.558611e-08	1.396955e-07	...
10	1.553215e-06	4.156946e-08	2.943160e-07	2.424562e-07	2.296629e-08	...
11	2.918158e-08	8.290708e-08	1.279739e-07	2.120731e-06	1.655901e-07	...
12	1.336915e-05	2.998512e-04	9.958988e-08	3.977077e-04	1.981446e-04	...
13	1.047650e-08	1.929870e-08	6.143130e-08	1.453181e-12	1.725295e-08	...
14	2.098574e-08	2.488127e-08	7.009927e-08	9.259284e-10	1.998737e-08	...
15	1.950443e-02	1.980302e-03	4.013827e-03	3.219206e-03	2.565051e-05	...
16	4.254076e-03	2.565590e-04	1.121688e-04	8.119726e-04	1.659742e-03	...
17	2.482111e-03	1.962775e-04	3.588491e-07	2.011517e-04	1.915284e-04	...
18	1.413451e-04	6.162914e-05	6.890259e-04	5.361615e-04	2.214068e-03	...
19	4.944232e-03	1.750887e-05	2.014842e-02	5.532645e-03	5.519015e-03	...
20	7.736321e-03	7.893786e-07	1.499149e-02	3.024582e-02	3.830078e-04	...
21	6.663280e-05	4.081109e-06	1.375424e-06	1.738199e-06	4.884434e-07	...
22	7.759033e-06	4.388471e-07	6.550686e-07	1.230640e-07	1.088272e-06	...
23	4.946113e-04	1.640849e-04	6.962040e-06	3.053630e-06	2.090964e-05	...
24	6.858266e-03	7.814143e-02	3.631062e-04	9.980553e-05	7.117011e-05	...
25	1.140846e-07	6.642350e-09	4.316839e-07	8.640717e-08	2.802443e-07	...
26	1.834510e-05	5.135138e-06	3.828060e-04	2.934290e-05	3.854495e-04	...
27	1.387183e-07	1.126508e-07	1.210495e-08	7.515276e-08	9.825897e-09	...
28	3.011053e-05	1.335785e-06	2.923687e-06	2.381997e-05	2.981178e-05	...

	19	20	21	22	23	\
0	1.779021e-06	6.891021e-07	7.429080e-14	3.689055e-10	1.028719e-09	
1	1.070309e-04	1.663944e-07	9.247934e-10	3.634661e-08	2.374448e-09	
2	5.427360e-33	2.769412e-34	1.086463e-35	3.820503e-33	9.350252e-33	
3	9.081115e-05	5.533742e-09	4.181400e-11	2.433898e-08	5.454664e-09	
4	2.619309e-07	2.475665e-07	3.153877e-08	9.445610e-10	2.857969e-09	
5	1.036250e-05	1.708549e-07	8.350763e-09	1.152036e-09	3.670980e-12	
6	1.034580e-02	8.286160e-08	2.955462e-09	5.963392e-07	3.160223e-12	
7	2.936355e-07	6.027652e-08	1.703297e-10	7.994870e-10	5.112204e-10	
8	3.179698e-07	1.216049e-07	3.054769e-09	8.661979e-11	2.117727e-10	
9	2.059696e-11	2.089892e-09	2.217920e-08	4.926904e-08	4.104677e-04	
10	6.654772e-07	2.101886e-07	1.599938e-09	2.230181e-05	6.156385e-10	
11	9.086117e-06	1.328908e-07	5.381305e-09	5.421052e-04	3.691488e-08	
12	4.487912e-05	9.715546e-08	8.429430e-10	3.937538e-08	9.941440e-11	
13	2.668280e-08	2.264974e-09	3.404684e-11	4.196949e-09	1.583828e-08	
14	2.974135e-08	8.560956e-09	8.789516e-11	6.613211e-09	2.198981e-08	
15	2.123329e-06	8.171571e-08	3.534715e-10	1.798427e-09	2.969846e-10	
16	2.838265e-05	7.561141e-08	1.333633e-09	1.941902e-09	9.279390e-09	
17	7.839742e-05	3.542392e-07	1.762367e-08	3.600582e-10	5.181846e-09	
18	1.759278e-06	2.801569e-09	8.536797e-11	4.354153e-10	1.288959e-10	
19	7.558116e-09	2.881825e-08	4.025917e-09	5.530339e-10	1.019217e-10	
20	4.174196e-07	3.788767e-08	8.226604e-10	1.253819e-11	9.513630e-10	
21	8.083980e-05	5.482078e-08	4.329065e-09	1.550453e-09	5.668290e-10	
22	2.199315e-07	5.278776e-03	7.997497e-08	2.126516e-08	1.564475e-10	
23	2.370944e-07	5.809872e-09	2.424832e-10	1.883128e-11	8.723809e-11	

24	4.977748e-07	2.009370e-09	1.096942e-09	1.069338e-10	1.533863e-09
25	7.797892e-08	7.692128e-10	5.092658e-08	6.880159e-10	1.815868e-05
26	7.202561e-07	1.460437e-09	7.720027e-09	1.043098e-10	1.424762e-08
27	3.015106e-08	2.401297e-07	1.802896e-03	1.782068e-09	8.790578e-09
28	2.941736e-05	1.083740e-07	5.860296e-08	1.878943e-09	2.873287e-09

	24	25	26	27	28
0	9.559039e-11	7.782490e-11	7.179469e-12	3.626119e-17	0.0
1	5.933740e-10	2.544053e-10	1.664656e-10	1.044662e-14	0.0
2	2.617267e-34	9.696099e-34	3.033731e-34	3.017450e-36	0.0
3	3.122055e-11	2.216779e-10	8.248321e-10	8.538479e-15	0.0
4	2.212265e-09	1.208314e-11	7.205069e-12	2.252899e-18	0.0
5	1.394607e-10	1.280125e-10	1.171959e-11	4.918472e-16	0.0
6	9.048352e-10	1.361916e-09	7.828882e-10	1.315248e-15	0.0
7	2.585187e-09	1.594288e-11	8.602841e-13	1.831413e-16	0.0
8	4.959682e-10	4.507233e-13	5.628125e-11	1.560778e-17	0.0
9	3.505153e-12	8.937343e-06	1.573112e-08	1.594073e-13	0.0
10	2.824646e-04	4.833505e-12	3.837380e-07	1.010749e-12	0.0
11	1.166119e-05	1.011852e-10	5.032608e-09	6.361039e-15	0.0
12	4.653022e-09	3.506843e-10	3.786330e-11	2.993149e-17	0.0
13	3.773529e-07	5.391550e-09	6.858746e-05	1.723515e-07	0.0
14	4.396232e-07	5.525749e-09	7.041239e-05	1.678190e-07	0.0
15	1.315522e-11	2.654283e-10	1.186189e-11	7.698595e-18	0.0
16	9.541928e-10	2.870547e-10	1.774522e-10	7.339152e-15	0.0
17	1.063602e-08	9.198897e-10	5.097201e-12	4.204837e-15	0.0
18	7.000373e-11	1.722687e-10	2.571716e-11	2.684352e-18	0.0
19	1.340113e-09	3.345418e-11	8.322273e-11	1.285739e-15	0.0
20	4.407859e-11	1.173433e-10	7.030713e-11	1.475600e-16	0.0
21	1.464138e-11	1.943305e-10	1.482737e-11	1.255755e-16	0.0
22	8.402974e-09	1.093274e-11	1.586854e-10	5.974975e-14	0.0
23	3.586459e-10	2.620502e-09	8.809871e-13	1.847358e-17	0.0
24	1.300035e-12	1.850713e-11	5.211922e-11	9.659621e-17	0.0
25	5.337715e-11	2.018348e-04	3.642368e-09	7.348006e-15	0.0
26	3.476657e-11	1.159763e-07	6.145903e-15	1.779019e-16	0.0
27	9.434932e-11	4.160934e-09	1.370133e-11	1.416329e-15	0.0
28	3.167163e-10	8.586257e-10	1.430145e-11	1.687923e-17	0.0

[29 rows x 29 columns]

```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
plt.grid()

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
```

```

ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
↪05),L[row,1]+random.uniform(-0.05,0.05)),**fontL2)

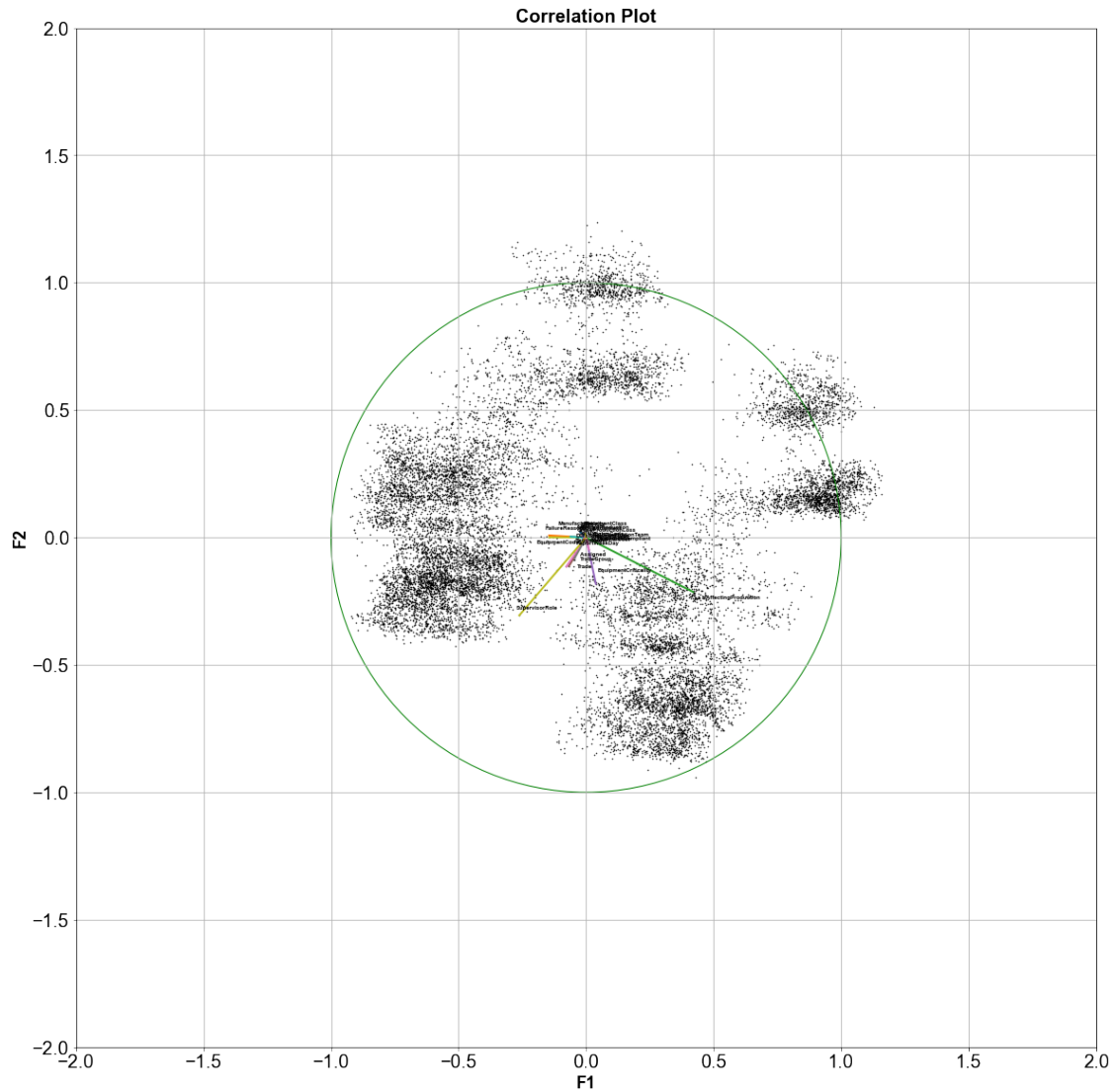
plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-2,2)
plt.ylim(-2,2)

plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)
plt.show()

```



```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
plt.grid()

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)
```

```

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
→05),L[row,1]+random.uniform(-0.05,0.05)),**fontL)

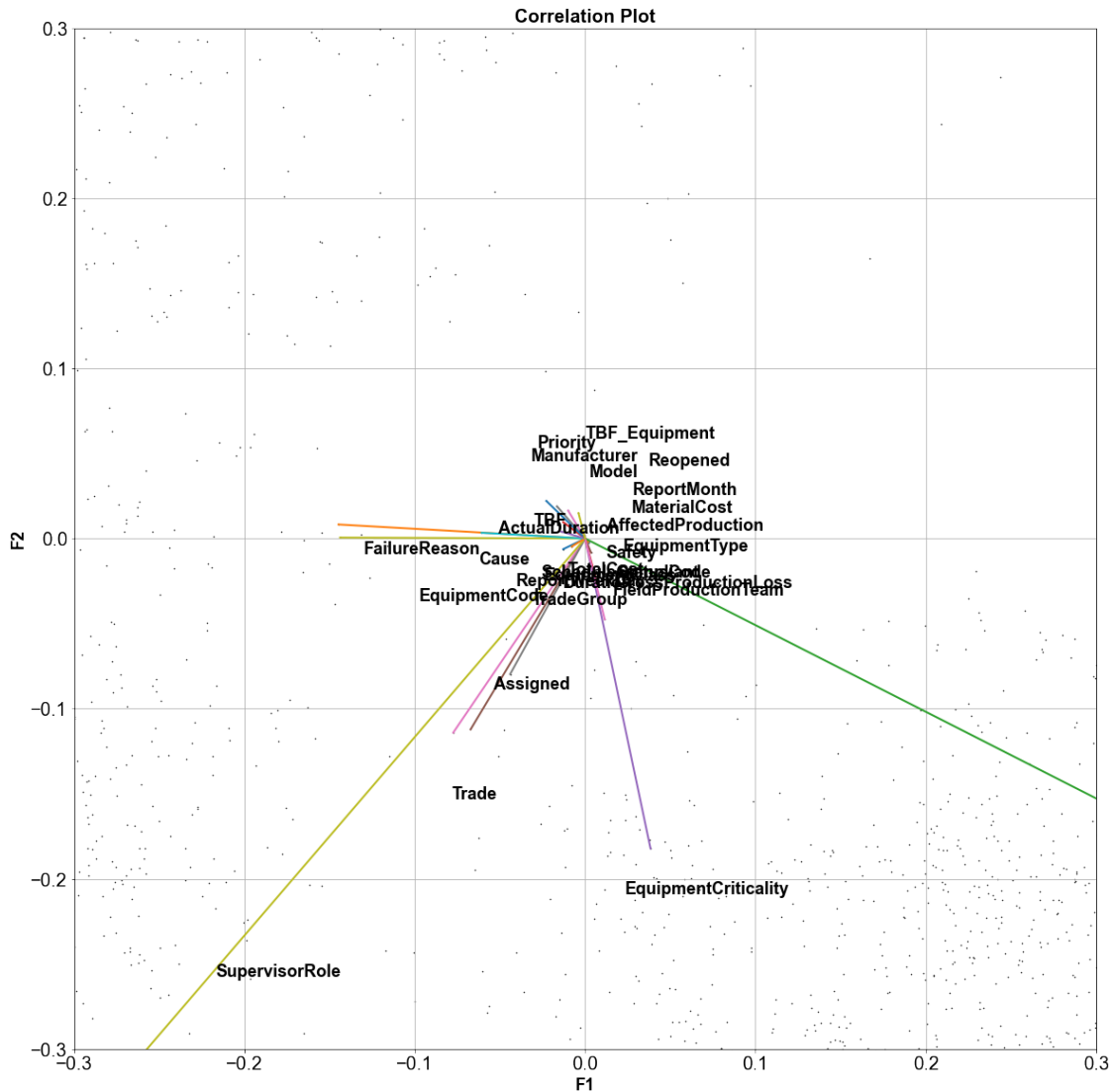
plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-0.3,0.3)
plt.ylim(-0.3,0.3)

plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)

```



```
[ ]: pca_pipe = make_pipeline(StandardScaler(), PCA())
pca_pipe.fit(dfAsset)
```

```
modelo_pca = pca_pipe.named_steps['pca']
```

```
[ ]: PCNames = []
for i in range(0, len(columnSystem)):
    nombre = "PC " + str(i)
    PCNames.append(nombre)
```

```
[ ]: # Se convierte el array a dataframe para añadir nombres a los ejes.
pca_table = pd.DataFrame(
```

```

data      = modelo_pca.components_,
columns   = dfAsset.columns,
index     = PCNames
)

print(pca_table)

```

	FieldProductionTeam	EquipmentCode	EquipmentType	EquipmentClass	\
PC 0	1.236528e-02	3.623062e-01	0.000000e+00	5.705974e-02	
PC 1	8.537809e-05	-1.540964e-01	3.144186e-17	-1.551039e-01	
PC 2	-2.865793e-02	2.004513e-01	-1.665335e-16	1.731984e-01	
PC 3	-8.723815e-02	-2.103153e-01	0.000000e+00	-1.021819e-01	
PC 4	-1.871246e-01	2.565924e-01	1.665335e-16	7.597408e-02	
PC 5	4.279456e-01	-3.894851e-02	-0.000000e+00	-8.142368e-02	
PC 6	-3.521052e-01	2.652774e-02	-1.387779e-17	-2.419492e-01	
PC 7	4.482886e-02	-9.995711e-03	-5.551115e-17	3.673004e-01	
PC 8	3.630391e-01	-5.101927e-02	1.387779e-16	2.535322e-01	
PC 9	-2.012211e-01	1.031889e-01	-1.387779e-17	2.277704e-01	
PC 10	-8.487362e-02	-3.267060e-02	-3.053113e-16	-3.668943e-01	
PC 11	-1.481033e-01	-4.640465e-03	-5.551115e-17	-4.159004e-02	
PC 12	2.148779e-01	-1.836097e-01	3.053113e-16	9.220969e-02	
PC 13	1.297532e-01	-3.934734e-02	-1.526557e-16	4.380093e-02	
PC 14	2.268682e-01	6.840667e-02	-5.551115e-17	-4.034535e-01	
PC 15	-2.979788e-02	-3.811372e-02	-3.469447e-17	-1.366432e-01	
PC 16	-2.402158e-01	-2.112946e-01	1.387779e-16	-9.207092e-03	
PC 17	-1.404062e-01	-2.475357e-02	-6.938894e-17	-3.492873e-01	
PC 18	2.449603e-01	-1.202167e-03	4.163336e-17	-6.056789e-02	
PC 19	2.899492e-01	1.527168e-01	8.326673e-17	-8.217309e-02	
PC 20	-3.580066e-02	-7.178898e-02	1.890849e-16	-1.525508e-01	
PC 21	1.409798e-01	-7.716048e-02	-3.747003e-16	-9.164298e-02	
PC 22	-2.158970e-01	3.300511e-01	-1.387779e-17	2.042573e-01	
PC 23	-2.620841e-02	6.883348e-02	-2.914335e-16	3.588807e-02	
PC 24	-8.719435e-03	-4.061841e-02	2.654127e-16	2.020125e-02	
PC 25	-1.587004e-01	-6.644461e-01	-1.665335e-16	2.706329e-01	
PC 26	-1.235827e-01	1.227973e-02	-1.110223e-16	-3.435015e-02	
PC 27	3.225652e-04	-5.648237e-03	-2.033963e-16	3.785687e-03	
PC 28	-1.466180e-17	3.807398e-18	1.000000e+00	-4.431948e-17	

	EquipmentCriticality	StatusCode	Priority	Cause	\
PC 0	-2.409633e-02	-2.604699e-02	-3.303243e-02	-1.860275e-02	
PC 1	1.282477e-01	2.759812e-02	-9.728611e-02	-7.521156e-02	
PC 2	-3.317347e-01	-7.702761e-02	9.733113e-02	1.154443e-01	
PC 3	-3.385961e-01	-1.144540e-01	-4.724997e-02	-2.628836e-01	
PC 4	3.259155e-03	2.167204e-02	-5.098706e-01	1.263089e-02	
PC 5	-1.704470e-02	2.360896e-01	-1.657818e-01	3.262119e-01	
PC 6	2.451923e-01	8.217520e-02	-4.300311e-03	1.337517e-01	
PC 7	2.512926e-01	5.561134e-01	1.985153e-01	-1.985241e-01	
PC 8	6.660990e-03	-2.560760e-01	6.539176e-02	-1.086807e-01	

PC 9	2.987408e-02	-4.458570e-02	8.610974e-02	-1.169456e-01
PC 10	-3.844665e-02	5.045301e-02	1.126537e-01	1.294075e-01
PC 11	-1.345987e-02	-1.000849e-01	1.839493e-01	7.543808e-02
PC 12	3.370437e-02	-2.738432e-01	9.965540e-02	3.555175e-01
PC 13	8.293121e-02	2.087385e-01	-8.865525e-02	-1.123465e-01
PC 14	-8.965127e-02	2.910259e-01	-1.176239e-01	-3.491524e-01
PC 15	-1.560262e-01	-2.289141e-02	-2.862238e-01	1.443104e-01
PC 16	7.752650e-03	4.496722e-01	2.984129e-02	3.351165e-01
PC 17	4.256292e-02	-1.188583e-01	3.407634e-01	1.045850e-01
PC 18	-1.928423e-01	1.529313e-01	2.749501e-01	-9.121553e-02
PC 19	-7.919604e-02	8.675446e-02	-2.544808e-01	3.751607e-01
PC 20	-6.283604e-02	-7.035877e-02	-3.007190e-01	-3.431687e-01
PC 21	6.160480e-01	-2.647506e-01	-1.422909e-01	-4.587570e-02
PC 22	9.672702e-03	9.540539e-03	-1.392052e-01	1.365475e-01
PC 23	-8.123858e-02	-1.306048e-02	-1.177834e-01	-6.423830e-02
PC 24	-3.736960e-01	2.754077e-02	4.781117e-02	2.277956e-02
PC 25	-1.047192e-01	8.453066e-03	-2.739671e-01	4.541295e-02
PC 26	-7.492377e-02	-3.860802e-03	2.116465e-02	2.201801e-02
PC 27	1.229885e-04	-6.774657e-04	8.142654e-04	7.542811e-04
PC 28	5.202736e-17	3.371640e-17	-5.476624e-17	-1.852273e-16

	FailureReason	Duration	...	Manufacturer	Model \
PC 0	2.936498e-01	5.609752e-02	...	1.985200e-01	9.240352e-02
PC 1	-1.882088e-01	-3.604430e-02	...	-8.121925e-02	-5.186149e-02
PC 2	1.941786e-01	9.913195e-02	...	1.639572e-01	1.518358e-01
PC 3	-2.206890e-01	1.164057e-01	...	-1.506404e-01	9.713207e-02
PC 4	9.794680e-02	1.477787e-02	...	3.151867e-01	1.640720e-01
PC 5	2.469910e-02	4.776409e-02	...	-1.469208e-01	-2.717999e-01
PC 6	5.184913e-02	3.260078e-01	...	-1.087019e-01	-1.454289e-01
PC 7	-5.781094e-02	2.077596e-01	...	1.864748e-01	1.830181e-01
PC 8	-6.215966e-02	4.828995e-01	...	4.610837e-02	9.763345e-02
PC 9	1.078989e-01	1.202527e-01	...	-2.010325e-01	-4.746774e-01
PC 10	-3.196568e-03	-5.963083e-02	...	1.510955e-01	2.272557e-01
PC 11	9.047885e-02	1.759972e-02	...	7.229416e-02	2.448261e-02
PC 12	-4.252081e-02	-1.647688e-01	...	6.473917e-02	4.075695e-01
PC 13	-8.403779e-02	1.611655e-01	...	8.805179e-02	1.108775e-01
PC 14	-2.112233e-01	-9.525811e-02	...	2.509566e-01	3.234359e-02
PC 15	2.281158e-02	6.010676e-01	...	-1.502072e-01	1.771267e-01
PC 16	2.500790e-01	-4.817016e-02	...	-1.900741e-01	2.643850e-01
PC 17	-1.449710e-01	3.209187e-01	...	2.940077e-01	-3.647360e-02
PC 18	1.865944e-01	4.192893e-02	...	-4.258493e-01	1.622210e-01
PC 19	-5.517470e-02	5.343205e-02	...	4.734372e-02	-2.533261e-01
PC 20	3.203991e-01	7.387160e-03	...	-2.702652e-01	1.997266e-01
PC 21	1.218605e-01	-6.812534e-02	...	-8.635721e-02	1.337278e-01
PC 22	-6.417166e-01	-6.042832e-02	...	-3.441226e-01	2.469236e-01
PC 23	2.097326e-01	-9.435482e-02	...	-1.147079e-02	1.457612e-02
PC 24	-3.443884e-02	-1.312407e-01	...	7.168132e-02	-8.844545e-02
PC 25	-4.477019e-02	1.134866e-02	...	2.404241e-01	-1.193897e-01

PC 26	-9.679596e-03	-2.656652e-02	...	2.575227e-02	-1.292026e-02
PC 27	-2.404557e-04	-1.684567e-03	...	-1.936871e-03	6.616846e-04
PC 28	7.200057e-17	-1.483362e-16	...	-3.262213e-18	-8.703950e-17

	Safety	Reopened	ReportMonth	ReportWeekDay	\
PC 0	5.891487e-02	2.050933e-02	-3.139300e-02	3.162645e-02	
PC 1	-1.441237e-04	4.093203e-02	4.812163e-02	-3.115183e-02	
PC 2	-3.800224e-02	-6.348747e-02	-9.875151e-02	5.697225e-03	
PC 3	-7.884854e-02	2.979007e-02	-1.841743e-01	2.043145e-02	
PC 4	1.326574e-01	6.475289e-02	-7.284631e-04	-5.510040e-02	
PC 5	-1.480800e-01	-6.562460e-02	-2.471367e-01	-8.584152e-03	
PC 6	1.883094e-01	7.778304e-02	1.487848e-01	6.463591e-02	
PC 7	-2.015067e-01	2.243274e-01	-3.308566e-01	7.260373e-02	
PC 8	-4.499318e-02	-1.568858e-01	4.884313e-01	2.821854e-01	
PC 9	2.103491e-02	5.738203e-01	9.441739e-02	1.831830e-05	
PC 10	-4.596717e-01	1.787723e-01	4.002123e-02	5.156218e-01	
PC 11	-6.250584e-01	1.797856e-02	2.427492e-01	-6.318882e-01	
PC 12	2.752869e-01	5.532693e-01	1.058670e-02	-1.470930e-01	
PC 13	1.367062e-01	-2.127284e-01	5.057630e-02	-3.986164e-01	
PC 14	1.005656e-01	3.068251e-01	3.307282e-01	-1.164105e-01	
PC 15	-1.029992e-01	2.176169e-01	-1.601560e-01	-1.041346e-01	
PC 16	1.394456e-01	-1.423154e-01	3.204639e-01	4.345729e-02	
PC 17	2.647752e-01	-1.281729e-01	-2.987373e-01	-1.184842e-01	
PC 18	1.588332e-01	6.568054e-02	-3.650636e-02	-1.334220e-01	
PC 19	-1.003021e-01	7.445208e-02	5.719881e-02	-1.135006e-02	
PC 20	-4.817487e-02	8.452851e-03	-9.659562e-02	1.467251e-02	
PC 21	-1.240854e-01	-1.258030e-02	-2.381375e-01	-2.981745e-02	
PC 22	-7.709898e-02	-4.178049e-02	1.749935e-03	-1.054444e-02	
PC 23	-5.520093e-02	-1.120265e-03	-3.491670e-02	-1.080657e-02	
PC 24	2.808503e-02	1.460402e-02	-2.241306e-01	9.744467e-03	
PC 25	-2.124372e-02	1.857138e-02	7.206583e-03	-8.853675e-04	
PC 26	2.882523e-02	-1.183748e-03	-2.329027e-02	-5.565138e-03	
PC 27	3.749568e-04	-3.656013e-03	2.773420e-04	5.962316e-04	
PC 28	-1.165169e-16	-7.777460e-17	-1.433701e-16	-1.465136e-16	

	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
PC 0	5.195509e-02	-4.433124e-02	-5.908347e-03	-3.176959e-03
PC 1	-9.495403e-02	1.237338e-01	-2.186154e-02	-7.521107e-02
PC 2	2.223379e-01	-3.020427e-01	5.834303e-02	1.678707e-01
PC 3	3.390302e-01	-4.473614e-01	2.278634e-02	1.220764e-01
PC 4	-2.480830e-02	-1.230843e-02	4.709177e-03	-9.469517e-02
PC 5	2.018313e-01	-2.454304e-01	5.303035e-03	3.826914e-02
PC 6	4.617011e-01	-1.216189e-01	-1.574360e-01	-4.917892e-01
PC 7	3.409604e-02	1.891204e-03	-1.948505e-01	-6.424588e-02
PC 8	1.759549e-01	1.501256e-01	-4.919323e-02	4.692561e-02
PC 9	1.966123e-02	1.110119e-02	3.562527e-01	2.920345e-01
PC 10	-4.930215e-02	3.590230e-02	4.322242e-01	-1.479197e-01
PC 11	7.447382e-02	1.048244e-02	-1.431872e-01	-5.279969e-02

PC 12	8.927999e-02	-9.401214e-02	6.480959e-03	-1.272802e-01
PC 13	9.717104e-02	5.960107e-03	7.479368e-01	-1.639345e-01
PC 14	1.538583e-01	-1.051525e-02	-1.665480e-01	2.477536e-01
PC 15	-2.694917e-01	2.451129e-01	-6.776935e-02	1.148236e-01
PC 16	1.023731e-01	3.945948e-02	4.036140e-03	4.638206e-01
PC 17	-8.974895e-02	6.974595e-02	1.354389e-02	2.688765e-01
PC 18	-1.044269e-01	9.364918e-02	1.679119e-03	-2.603121e-01
PC 19	-4.386396e-02	1.694588e-02	-2.081915e-02	-2.938658e-03
PC 20	-7.024036e-02	-1.755805e-02	-2.591117e-02	-3.132939e-02
PC 21	3.275021e-01	1.289438e-01	5.045373e-02	3.106660e-01
PC 22	7.584363e-02	6.710232e-02	1.610568e-02	6.568015e-02
PC 23	2.398373e-01	2.175953e-01	7.431313e-03	-8.458707e-03
PC 24	4.550521e-01	6.589247e-01	4.290790e-03	-5.250422e-02
PC 25	-3.590899e-02	-6.446943e-03	-1.128066e-02	-7.191629e-02
PC 26	-5.244866e-04	6.758727e-02	-2.996164e-03	5.895067e-03
PC 27	-2.823894e-04	-6.828917e-04	-3.287981e-04	1.025441e-04
PC 28	5.065499e-17	3.348183e-17	2.082889e-16	2.131773e-17

[29 rows x 29 columns]

```
[ ]: dfAsset.columns
```

```
[ ]: Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
          'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
          'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
          'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
          'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
          'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
          'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
          'TBF_Equipment'],
          dtype='object')
```

```
[ ]: commonPCs = []
for i in range(0, len(maxPCvars)):
    name = maxPCvars[i]
    maxPCAit = getMaxComponent(pca_table, name)
    commonPCs.append(maxPCAit)

dictPCs = dict(zip(maxPCvars, commonPCs))

print(dictPCs)
```

```
{'TBF_Equipment': [2, 9, 14, 16, 17, 21], 'ActualDuration': [], 'TotalCost': [1,
27], 'IsAffectingProduction': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 19, 20, 21, 22, 23, 24, 26, 27, 28]}
```

```
[ ]: import heapq
```

```

for i in range(0,len(dictPCs)):
    name = maxPCvars[i]
    print(name)
    PC_num = dictPCs[name]
    print(PC_num)
    for j in range(0,len(PC_num)):
        print(PC_num[j])
        PCval = pca_table.iloc[j]
        mean = PCval.mean()
        largest = heapq.nlargest(3, enumerate(PCval), key=lambda x: x[1])
        lowest = heapq.nsmallest(3, enumerate(PCval), key=lambda x: x[1])
        print("---Highest---")
        for k in range(0,len(largest)):
            indexTabla = largest[k][0]
            print(pca_table.columns[indexTabla])
        print("---Lowest---")
        for k in range(0,len(lowest)):
            indexTabla = lowest[k][0]
            print(pca_table.columns[indexTabla])

```

```

TBF_Equipment
[2, 9, 14, 16, 17, 21]
2
---Highest---
SupervisorRole
Trade
TradeGroup
---Lowest---
IsAffectingProduction
ScheduleCompliant
Priority
9
---Highest---
TotalCost
MaterialCost
IsAffectingProduction
---Lowest---
FailureReason
EquipmentClass
EquipmentCode
14
---Highest---
MaterialCost
TotalCost
ActualDuration
---Lowest---
EquipmentCriticality
ScheduleCompliant

```

TradeGroup
 16
 ---Highest---
 ActualDuration
 TradeGroup
 Trade
 ---Lowest---
 ScheduleCompliant
 EquipmentCriticality
 Cause
 17
 ---Highest---
 AffectedProduction
 GrossProductionLoss
 Manufacturer
 ---Lowest---
 Priority
 FieldProductionTeam
 Trade
 21
 ---Highest---
 Assigned
 FieldProductionTeam
 Cause
 ---Lowest---
 Model
 ReportMonth
 ScheduleCompliant
 ActualDuration
 []
 TotalCost
 [1, 27]
 1
 ---Highest---
 SupervisorRole
 Trade
 TradeGroup
 ---Lowest---
 IsAffectingProduction
 ScheduleCompliant
 Priority
 27
 ---Highest---
 TotalCost
 MaterialCost
 IsAffectingProduction
 ---Lowest---
 FailureReason

EquipmentClass
 EquipmentCode
 IsAffectingProduction
 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22,
 23, 24, 26, 27, 28]
 0
 ---Highest---
 SupervisorRole
 Trade
 TradeGroup
 ---Lowest---
 IsAffectingProduction
 ScheduleCompliant
 Priority
 1
 ---Highest---
 TotalCost
 MaterialCost
 IsAffectingProduction
 ---Lowest---
 FailureReason
 EquipmentClass
 EquipmentCode
 2
 ---Highest---
 MaterialCost
 TotalCost
 ActualDuration
 ---Lowest---
 EquipmentCriticality
 ScheduleCompliant
 TradeGroup
 3
 ---Highest---
 ActualDuration
 TradeGroup
 Trade
 ---Lowest---
 ScheduleCompliant
 EquipmentCriticality
 Cause
 4
 ---Highest---
 AffectedProduction
 GrossProductionLoss
 Manufacturer
 ---Lowest---
 Priority

FieldProductionTeam
 Trade
 5
 ---Highest---
 Assigned
 FieldProductionTeam
 Cause
 ---Lowest---
 Model
 ReportMonth
 ScheduleCompliant
 6
 ---Highest---
 ActualDuration
 Duration
 EquipmentCriticality
 ---Lowest---
 TBF_Equipment
 FieldProductionTeam
 EquipmentClass
 7
 ---Highest---
 StatusCode
 EquipmentClass
 EquipmentCriticality
 ---Lowest---
 ReportMonth
 Safety
 Cause
 8
 ---Highest---
 ReportMonth
 Duration
 FieldProductionTeam
 ---Lowest---
 StatusCode
 Reopened
 Cause
 9
 ---Highest---
 Reopened
 TBF
 TBF_Equipment
 ---Lowest---
 Model
 FieldProductionTeam
 Manufacturer
 10

```

---Highest---
ReportWeekDay
TBF
Model
---Lowest---
Safety
EquipmentClass
TBF_Equipment
11
---Highest---
ReportMonth
Priority
GrossProductionLoss
---Lowest---
ReportWeekDay
Safety
FieldProductionTeam
12
---Highest---
Reopened
Model
Cause
---Lowest---
StatusCode
EquipmentCode
Duration
13
---Highest---
TBF
StatusCode
Duration
---Lowest---
ReportWeekDay
Reopened
TBF_Equipment
14
---Highest---
ReportMonth
Reopened
StatusCode
---Lowest---
EquipmentClass
Cause
IsAffectingProduction
15
---Highest---
Duration
ScheduleCompliant

```


Reopened
 ---Lowest---
 GrossProductionLoss
 Priority
 ActualDuration
 16
 ---Highest---
 TBF_Equipment
 StatusCode
 Cause
 ---Lowest---
 FieldProductionTeam
 EquipmentCode
 Manufacturer
 17
 ---Highest---
 GrossProductionLoss
 Priority
 Duration
 ---Lowest---
 EquipmentClass
 ReportMonth
 FailureReason
 19
 ---Highest---
 AffectedProduction
 Priority
 FieldProductionTeam
 ---Lowest---
 Manufacturer
 IsAffectingProduction
 TBF_Equipment
 20
 ---Highest---
 GrossProductionLoss
 Cause
 FieldProductionTeam
 ---Lowest---
 Assigned
 AffectedProduction
 Priority
 21
 ---Highest---
 GrossProductionLoss
 FailureReason
 Assigned
 ---Lowest---
 AffectedProduction

Cause
 Priority
 22
 ---Highest---
 EquipmentCriticality
 ActualDuration
 TBF_Equipment
 ---Lowest---
 Assigned
 StatusCode
 ReportMonth
 23
 ---Highest---
 EquipmentCode
 Model
 EquipmentClass
 ---Lowest---
 FailureReason
 Manufacturer
 IsAffectingProduction
 24
 ---Highest---
 Assigned
 TradeGroup
 ActualDuration
 ---Lowest---
 SupervisorRole
 Priority
 Duration
 26
 ---Highest---
 ScheduleCompliant
 ActualDuration
 SupervisorRole
 ---Lowest---
 EquipmentCriticality
 ReportMonth
 Duration
 27
 ---Highest---
 EquipmentClass
 Manufacturer
 AffectedProduction
 ---Lowest---
 EquipmentCode
 IsAffectingProduction
 Priority
 28

```

---Highest---
Trade
Assigned
ScheduleCompliant
---Lowest---
TradeGroup
SupervisorRole
FieldProductionTeam

```

3.1 Regresion

```
[ ]: column_names = dfAsset_scaled.columns
      print(column_names)
```

```

Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
      'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
      'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
      'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
      'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
      'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
      'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
      'TBF_Equipment'],
      dtype='object')

```

```
[ ]: ## Target ##
      y = dfAsset_scaled['TotalCost']
      y.head()
```

```
[ ]: 0    0.000438
      1    0.000504
      2    0.000602
      3    0.000657
      4    0.000657
      Name: TotalCost, dtype: float64
```

```
[ ]: ## Rest of Variables ##
      x = dfAsset_scaled[column_names.drop(['TotalCost'])]
      x.head()
```

```
[ ]:  FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
0                0.870968         0.316572           0.0             0.6
1                0.870968         0.319321           0.0             0.6
2                0.225806         0.632821           0.0             1.0
3                0.064516         0.374616           0.0             0.6
4                0.870968         0.305416           0.0             0.6

      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason  \
0                0.2          0.0      0.6  0.500000      0.117117
```

1	0.2	0.0	0.6	0.543478	0.761261
2	0.2	0.0	0.6	0.260870	0.797297
3	0.2	0.0	0.6	0.500000	0.117117
4	0.2	0.0	0.6	0.652174	0.027027

	Duration	...	Manufacturer	Model	Safety	Reopened	ReportMonth	\
0	0.0	...	0.487685	0.834069	0.0	0.0	1.000000	
1	0.0	...	0.487685	0.828773	0.0	0.0	1.000000	
2	0.0	...	0.541872	0.872904	0.0	0.0	1.000000	
3	0.0	...	0.014778	0.872904	0.0	0.0	1.000000	
4	0.0	...	0.487685	0.480141	0.0	0.0	0.909091	

	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
0	0.833333	0.016323		0.0	0.000798
1	0.666667	0.017182		0.0	0.000798
2	0.666667	0.017182		0.0	0.140425
3	0.166667	0.003436		0.0	0.110934
4	0.000000	0.040378		0.0	0.010375

[5 rows x 28 columns]

```
[ ]: #Separate train and test data
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
↪20,random_state=0)
```

```
[ ]: print("Size of the full data set: ",x.shape)
print("Size of the training data set: ",x_train.shape)
print("Size of the test data set: ",x_test.shape)
```

```
Size of the full data set: (16097, 28)
Size of the training data set: (12877, 28)
Size of the test data set: (3220, 28)
```

```
[ ]: EquipmentCode = dfAsset_scaled["EquipmentCode"]
EquipmentCode.head()
```

```
[ ]: 0    0.316572
1    0.319321
2    0.632821
3    0.374616
4    0.305416
Name: EquipmentCode, dtype: float64
```

```
[ ]: TotalCost_real = dfAsset_scaled["TotalCost"]
TotalCost_real.head()
```

```
[ ]: 0    0.000438
1    0.000504
```

```

2    0.000602
3    0.000657
4    0.000657
Name: TotalCost, dtype: float64

```

```

[ ]: df_code_cost = pd.DataFrame()

df_code_cost["Code"] = EquipmentCode
df_code_cost["Cost"] = TotalCost_real

df_code_cost.head()

```

```

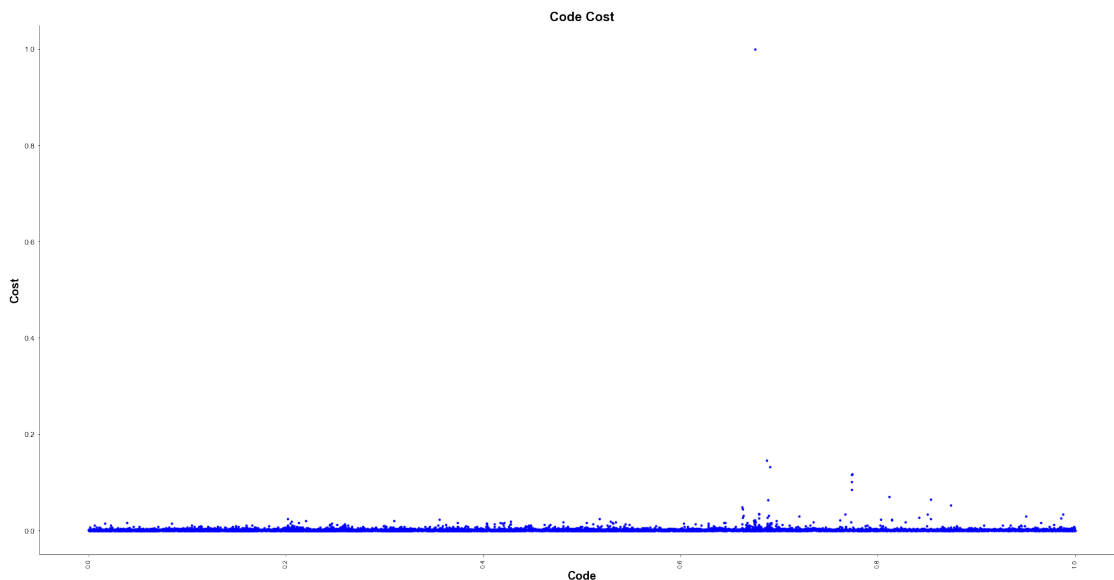
[ ]:      Code      Cost
0  0.316572  0.000438
1  0.319321  0.000504
2  0.632821  0.000602
3  0.374616  0.000657
4  0.305416  0.000657

```

```

[ ]: ## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c =_
↪"blue" , s = 30)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

```



```
[ ]: # importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
# fitting the training data
LR.fit(x_train,y_train)
```

```
[ ]: LinearRegression()
```

```
[ ]: y_prediction = LR.predict(x_test)
y_prediction
```

```
[ ]: array([0.00284065, 0.00035894, 0.00064312, ..., 0.00057127, 0.00065208,
0.00037969])
```

```
[ ]: # importing r2_score module
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

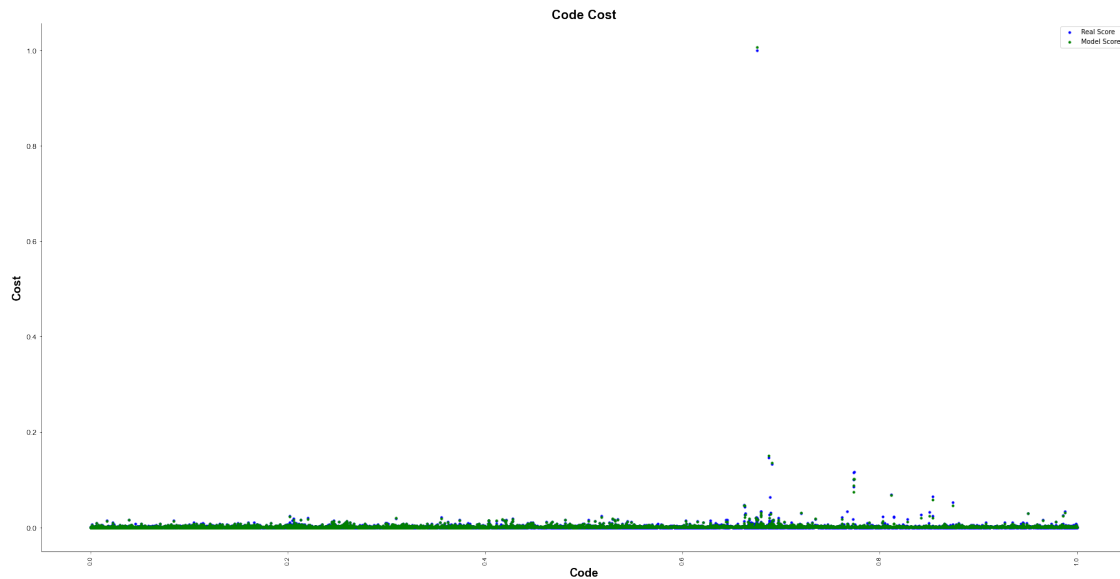
score = r2_score(y_test,y_prediction)
print("R2 score is ",score)
print("Mean squared error is ",mean_squared_error(y_test,y_prediction))
print("Mean absolute error is ",mean_absolute_error(y_test,y_prediction))
print("Root mean squared error is ",np.
    ↳sqrt(mean_squared_error(y_test,y_prediction)))
```

```
R2 score is 0.975512143276988
Mean squared error is 3.729304440891901e-07
Mean absolute error is 0.00032324128746031184
Root mean squared error is 0.0006106803125115383
```

```
[ ]: y_prediction = LR.predict(x)

## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c =_
    ↳"blue" , s = 40)
```

```
plt.scatter(x = df_code_cost["Code"],y = y_prediction, marker=".", c = "green",
            s = 40)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.legend(['Real Score','Model Score'])
plt.show()
```



4 Position

```
[ ]: ## Position Dataframe ##
dfPosition = df.loc[df["EquipmentType"] == 'Position']
dfPosition.head()
```

```
[ ]:   WorkOrder  FieldProductionTeam  EquipmentCode  EquipmentType  \
7      15465937             MCELROY   HF0549001WH      Position
11     15464576                VAN    BP8T300SP1      Position
13     15516278             RANGELY    C35CCK-1      Position
22     15514480             RANGELY  C35CS30V5-P5      Position
30     15496981             RANGELY      P-2010B      Position

      EquipmentClass  EquipmentCriticality  StatusCode  Priority  \
7      SE - Well Head                3 - Low          C         3
11      Pump - Gear                3 - Low          C         3
13      Compressor - Rotary          3 - Low          C         2
22  Pump - Centrifugal Single Stage          3 - Low          C         2
30  Pump - Centrifugal Single Stage          3 - Low          C         3
```

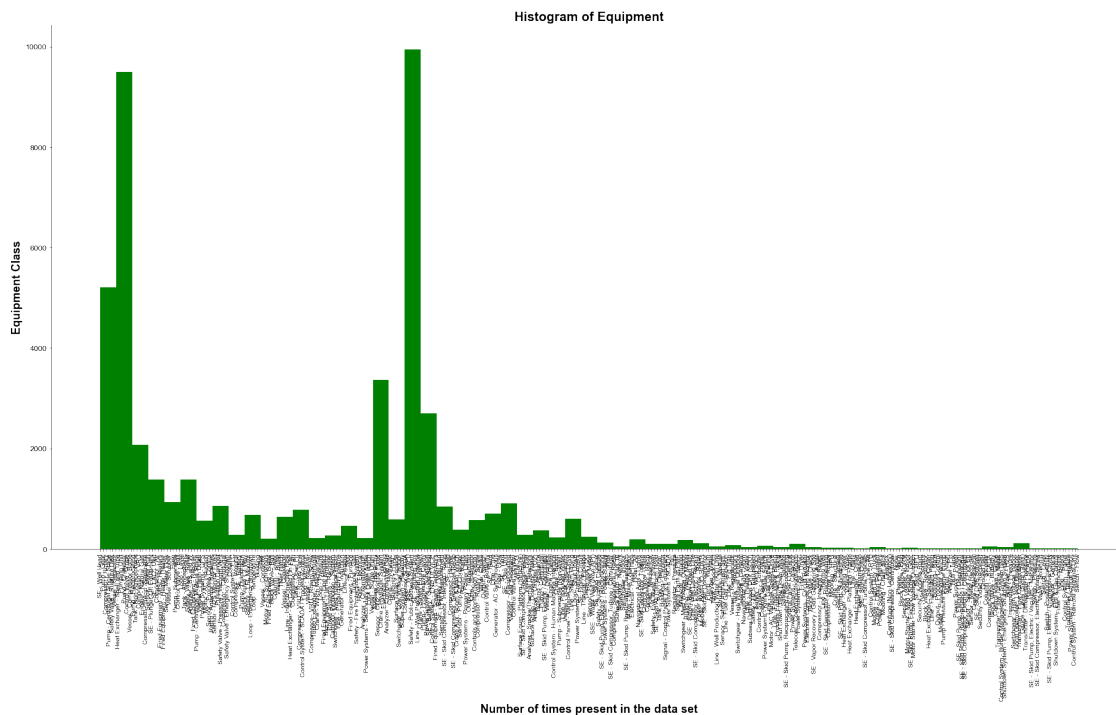
	Cause		FailureReason			
7	No code needed, See comments		No code needed, See comments		\	
11	Leaking		Pipe - Piping		\	
13	Communication Failure		Assembly - Control System		\	
22	Leaking		Component - Seal, Mechanical		\	
30	Leaking		Component - Seal, Mechanical		\	

	Manufacturer	Model	Safety	Reopened	ReportMonth	ReportWeekDay	\
7	NONEAVAILABLE	NONE	N	N	12	6	
11	None	None	N	N	12	4	
13	LEROI COMPRESSOR	A219-83-2	N	Y	1	5	
22	None	None	N	N	1	5	
30	GOULDS	None	N	N	12	3	

	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
7	27	Future Week	0	910.0
11	28	Future Week	0	274.0
13	-1	Future Week	0	76.0
22	1	Future Week	0	NaN
30	10	Future Week	0	1033.0

[5 rows x 30 columns]

```
[ ]: createGraphs(dfPosition)
```




```
[ ]: dfPositionCommonValues = getCommonValues(dfPosition,400)
print(dfPositionCommonValues)
```

```
['SE - Well Head', 'Pump - Centrifugal Single Stage', 'Vessel - Separator',
'Pumping Unit', 'Pump - Reciprocating', 'Compressor - Recip /Separable',
'Compressor - Air', 'Tank - Atmospheric', 'Safety Valve - Pressure / Vacuum
Vent', 'Control System - SCADA / Process Control', 'System - Hierarchy',
'Electrical - Lighting', 'SSE - Well (API)', 'SE - Stuffing Box', 'Line - Well
Production Flowline', 'SE - Battery / Gauge Setting', 'Fired Equipment - Steam
Generators']
```

```
[ ]: dfPosition = dfPosition.loc[dfPosition["EquipmentClass"].
↳isin(dfPositionCommonValues)]
dfPosition.head()
```

```
[ ]:      WorkOrder FieldProductionTeam EquipmentCode EquipmentType \
7      15465937          MCELROY    HF0549001WH      Position
22     15514480          RANGELY   C35CS30V5-P5      Position
30     15496981          RANGELY      P-2010B      Position
32     15494047           VAN      IS5327WH      Position
50     15516275          RANGELY   C35CS20P1      Position
```

```
      EquipmentClass EquipmentCriticality StatusCode Priority \
7      SE - Well Head          3 - Low          C          3
22  Pump - Centrifugal Single Stage          3 - Low          C          2
30  Pump - Centrifugal Single Stage          3 - Low          C          3
32      SE - Well Head          3 - Low          C          4
50  Pump - Centrifugal Single Stage          2 - Moderate          C          1
```

```
      Cause      FailureReason ... \
7  No code needed, See comments      No code needed, See comments ...
22      Leaking      Component - Seal, Mechanical ...
30      Leaking      Component - Seal, Mechanical ...
32      Leaking      Seal / Gasket - Gasket, Seals & O-Ring ...
50      Leaking      Component - Seal, Mechanical ...
```

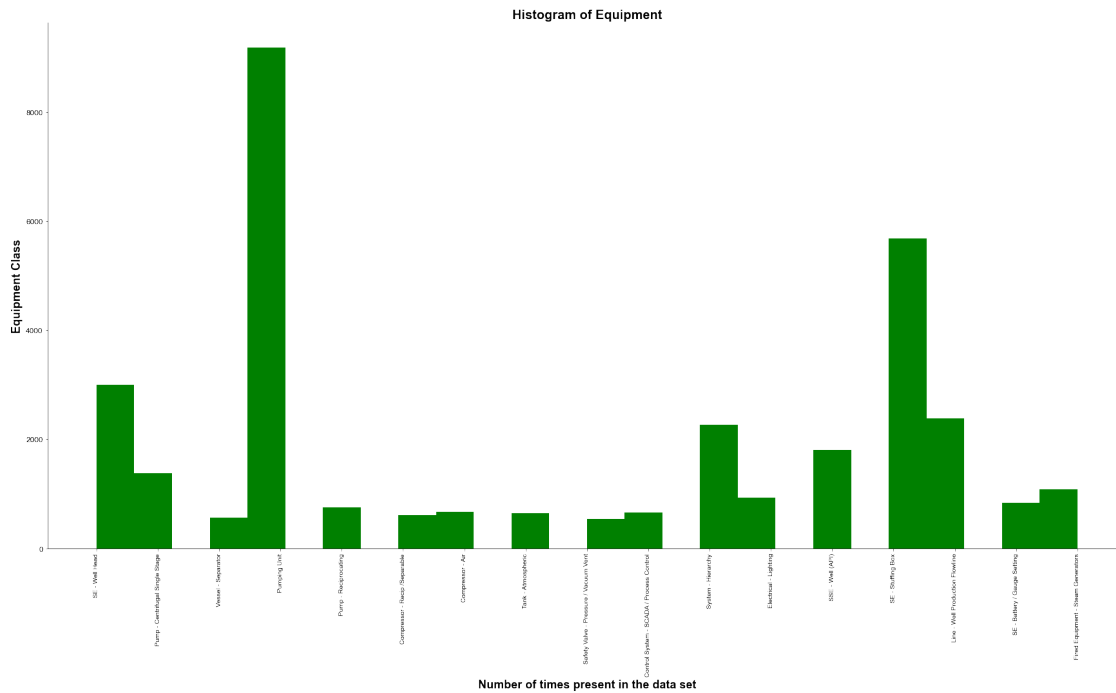
```
      Manufacturer Model Safety Reopened ReportMonth ReportWeekDay \
7  NONEAVAILABLE  NONE      N      N          12          6
22      None      None      N      N          1          5
30      GOULDS      None      N      N          12          3
32  NONEAVAILABLE  NONE      N      N          12          2
50  WORTHINGTON      None      N      N          1          6
```

```
      ActualDuration ScheduleCompliant TBF TBF_Equipment
7      27      Future Week      0      910.0
22      1      Future Week      0      NaN
30      10      Future Week      0      1033.0
32      11      Future Week      0      NaN
```

50 0 Future Week 0 26.0

[5 rows x 30 columns]

```
[ ]: createGraphs(dfPosition)
```



```
[ ]: ## Now that the data has been narrowed to the most common case we will encode
      ↳ it make some changes to it ##
dfPosition= EncodeData(dfPosition)
dfPosition.head()
```

```
[ ]:      FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass  \
WorkOrder
15465937                  30             9382              0              11
15514480                  38             6428              0              6
15496981                  38            15908              0              6
15494047                  45            11056              0              11
15516275                  38             6413              0              6
```

```
      EquipmentCriticality  StatusCode  Priority  Cause  FailureReason  \
WorkOrder
15465937                 2           0         3    28          217
15514480                 2           0         2    22           77
15496981                 2           0         3    22           77
15494047                 2           0         4    22          283
```

15516275	1	0	1	22	77
----------	---	---	---	----	----

	Duration	...	Manufacturer	Model	Safety	Reopened	ReportMonth	\
WorkOrder		...						
15465937	1	...	149	1308	0	0	12	
15514480	1	...	152	1316	0	0	1	
15496981	5	...	86	1316	0	0	12	
15494047	1	...	149	1308	0	0	12	
15516275	1	...	254	1316	0	0	1	

	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	\
WorkOrder					
15465937	6	27		0	0
15514480	5	1		0	0
15496981	3	10		0	0
15494047	2	11		0	0
15516275	6	0		0	0

	TBF_Equipment
WorkOrder	
15465937	910.0
15514480	NaN
15496981	1033.0
15494047	NaN
15516275	26.0

[5 rows x 29 columns]

```
[ ]: ## We need to handle the datasets NA Values ##
dfPosition = dfPosition.fillna(dfPosition.mean())
dfPosition.head()
```

	FieldProductionTeam	EquipmentCode	EquipmentType	EquipmentClass	\
WorkOrder					
15465937	30	9382	0	11	
15514480	38	6428	0	6	
15496981	38	15908	0	6	
15494047	45	11056	0	11	
15516275	38	6413	0	6	

	EquipmentCriticality	StatusCode	Priority	Cause	FailureReason	\
WorkOrder						
15465937	2	0	3	28	217	
15514480	2	0	2	22	77	
15496981	2	0	3	22	77	
15494047	2	0	4	22	283	
15516275	1	0	1	22	77	

	Duration	...	Manufacturer	Model	Safety	Reopened	ReportMonth	\
WorkOrder		...						
15465937	1	...	149	1308	0	0	12	
15514480	1	...	152	1316	0	0	1	
15496981	5	...	86	1316	0	0	12	
15494047	1	...	149	1308	0	0	12	
15516275	1	...	254	1316	0	0	1	

	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	\
WorkOrder					
15465937	6	27		0	0
15514480	5	1		0	0
15496981	3	10		0	0
15494047	2	11		0	0
15516275	6	0		0	0

	TBF_Equipment
WorkOrder	
15465937	910.000000
15514480	100.481815
15496981	1033.000000
15494047	100.481815
15516275	26.000000

[5 rows x 29 columns]

```
[ ]: ## For further analysis we know need to standarize the dataset##
## Scales ##
scaling_procedure_1 = MinMaxScaler(feature_range= (0,1))
```

```
[ ]: ## Scaled Data ##
columnSystem = dfPosition.columns
dfPosition_scaled = scaling_procedure_1.fit_transform(dfPosition)
dfPosition_scaled = pd.DataFrame(dfPosition_scaled, columns = columnSystem)
dfPosition_scaled.head()
```

	FieldProductionTeam	EquipmentCode	EquipmentType	EquipmentClass	\
0	0.60	0.459407	0.0	0.6875	
1	0.76	0.314759	0.0	0.3750	
2	0.76	0.778964	0.0	0.3750	
3	0.90	0.541377	0.0	0.6875	
4	0.76	0.314024	0.0	0.3750	

	EquipmentCriticality	StatusCode	Priority	Cause	FailureReason	\
0	0.333333	0.0	0.6	0.583333	0.623563	
1	0.333333	0.0	0.4	0.458333	0.221264	

2	0.333333	0.0	0.6	0.458333	0.221264
3	0.333333	0.0	0.8	0.458333	0.813218
4	0.166667	0.0	0.2	0.458333	0.221264

	Duration	...	Manufacturer	Model	Safety	Reopened	ReportMonth	\
0	0.000000	...	0.582031	0.876676	0.0	0.0	1.0	
1	0.000000	...	0.593750	0.882038	0.0	0.0	0.0	
2	0.003633	...	0.335938	0.882038	0.0	0.0	1.0	
3	0.000000	...	0.582031	0.876676	0.0	0.0	1.0	
4	0.000000	...	0.992188	0.882038	0.0	0.0	0.0	

	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
0	0.833333	0.039578		0.0	0.787879
1	0.666667	0.005277		0.0	0.086997
2	0.333333	0.017150		0.0	0.894372
3	0.166667	0.018470		0.0	0.086997
4	0.833333	0.003958		0.0	0.022511

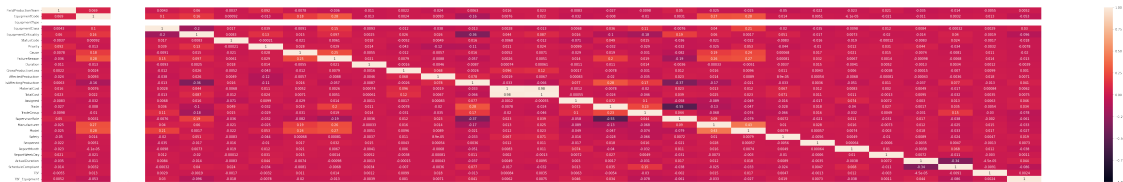
[5 rows x 29 columns]

```
[ ]: ##Here is the correlation plot of the system##
```

```
corr_df = dfPosition_scaled

corr = dfPosition_scaled.corr()
corr.style.background_gradient(cmap="coolwarm")

plt.figure(figsize=(70, 10))
heatmap = sns.heatmap(corr, vmin=-1, vmax=1, annot=True)
```



```
[ ]: ## PCA ##
#Only O3 Dataframe
```

```
Xdat = dfPosition_scaled

header_row= dfPosition_scaled.columns.values

y=dfPosition_scaled["TBF"]
```

```
[ ]: # create the PCA instance
```

```
pca = PCA()
```

```
# fit on data
```

```
pca = pca.fit(Xdat)
```

```
#Q eigenvectors
```

```
Q=pca.components_.T
```

```
#Explained deviations
```

```
s= np.sqrt(pca.explained_variance_)
```

```
[ ]: # transform data obtain sample scores
```

```
F = pca.transform(Xdat)
```

```
print("Sample scores")
```

```
## Sample Scores ##
```

```
sample_scores = pd.DataFrame(F)
```

```
print(sample_scores)
```

Sample scores

	0	1	2	3	4	5	6	\
0	0.377383	-0.469449	0.505591	0.468138	0.259979	0.781577	-0.023162	
1	0.542688	0.256959	-0.338437	0.232891	0.180123	-0.392463	-0.039968	
2	0.619713	0.092700	0.576509	-0.105719	-0.147326	-0.280003	0.331944	
3	0.690101	-0.294070	0.375884	-0.199484	-0.232225	0.532720	-0.225914	
4	0.549148	0.171528	-0.342735	0.181224	0.373109	-0.366207	-0.082318	
...	
32954	-0.312003	-0.028584	0.024953	-0.273731	0.457366	-0.124108	-0.397806	
32955	-0.655779	0.419331	0.042011	0.444182	-0.092063	0.493062	-0.058195	
32956	-0.644877	0.012484	-0.021703	0.503789	0.421256	0.429298	-0.355388	
32957	-0.313700	-0.022215	0.026008	-0.269598	0.456687	-0.122533	-0.408359	
32958	0.344957	-0.062453	-0.017791	0.394868	-0.417676	0.522973	0.368113	
	7	8	9	...	19	20	21	\
0	-0.264959	0.226853	0.122770	...	0.631069	-0.056468	-0.009740	
1	-0.224867	0.425840	0.000782	...	-0.047357	-0.011244	-0.006922	
2	-0.124614	0.224570	0.171521	...	0.756820	-0.042267	-0.005794	
3	-0.008980	-0.120836	0.117429	...	-0.138556	-0.017686	-0.022431	
4	-0.341155	0.490023	-0.116049	...	-0.117066	-0.012941	-0.008654	
...	
32954	-0.007435	-0.117970	0.042273	...	-0.001614	0.007915	-0.006155	
32955	0.177288	-0.141351	0.252147	...	0.001627	0.001012	-0.002083	
32956	0.249957	-0.052686	0.216005	...	0.003848	0.001977	-0.003009	
32957	-0.009303	-0.115030	0.040674	...	-0.001689	0.007943	-0.006139	
32958	-0.207426	0.055215	-0.000790	...	0.013733	-0.001472	-0.004952	
	22	23	24	25	26	27	\	

```

0    -0.014860  0.000235 -0.001779 -0.000995 -0.000452 -0.000389
1    -0.046190 -0.000801 -0.000168 -0.002236 -0.000915  0.000497
2    -0.040011  0.003615  0.003564 -0.001072 -0.000768  0.000779
3    -0.033413 -0.000225 -0.004597  0.000359 -0.000859 -0.001014
4    -0.046161 -0.000481  0.002857 -0.003589  0.008309  0.000251
...
32954  0.000564 -0.002651  0.000276 -0.000634  0.000007  0.000174
32955  0.000106 -0.002650  0.000665 -0.000363  0.000338  0.000228
32956  0.000165  0.009499  0.000505 -0.000289  0.000278  0.000232
32957  0.000555 -0.002661  0.000238 -0.001623  0.000082  0.000177
32958 -0.000650 -0.002480 -0.000199  0.001177  0.000256 -0.000243

```

```

28
0    2.289387e-16
1   -8.608825e-16
2   -3.890942e-16
3   -3.743169e-16
4   -6.669565e-16
...
32954 -1.721689e-16
32955 -1.454480e-16
32956 -1.810767e-16
32957 -1.745750e-16
32958 -3.517924e-16

```

[32959 rows x 29 columns]

```

[ ]: # Obtain Factor Loadings
L = np.zeros(shape=(Q.shape))
for col in range(0, len(s)-1):
    L[:, col] = Q[:, col] * s[col]
print("Factor Loadings")

columns_factorLoading = []

for i in range(0, 29):
    nombre = "F" + str(i)
    columns_factorLoading.append(nombre)

## Factor Loadings Dataframe ##
factor_loadings = pd.DataFrame(L, columns= columns_factorLoading)
print(factor_loadings)

```

Factor Loadings

```

          F0          F1          F2          F3          F4 \
0  2.619206e-03  4.178606e-03 -7.282249e-03 -1.087026e-02  8.801606e-03
1  6.478982e-02 -1.789463e-01 -2.474089e-02 -9.016877e-02  1.415127e-02

```

2	5.932523e-17	-4.367853e-17	-1.822443e-17	-1.697061e-17	6.080166e-18
3	1.956975e-03	-4.786716e-02	-7.292263e-03	-2.152058e-02	1.363107e-02
4	8.031859e-02	-1.397731e-02	-1.401154e-03	9.507220e-03	-8.996001e-03
5	-5.027853e-03	-1.932222e-03	-7.736050e-03	8.819545e-03	-2.041717e-03
6	9.070844e-03	-4.473791e-03	2.276884e-03	9.461392e-03	9.753631e-04
7	-1.659745e-03	-1.056977e-01	4.917540e-04	-4.368025e-02	2.119048e-02
8	1.293948e-02	-2.375816e-01	2.564928e-04	7.282096e-03	-6.124202e-03
9	7.205460e-05	-7.819513e-05	-2.573222e-05	2.769452e-04	-1.213889e-06
10	5.692219e-05	-1.326543e-05	2.589934e-05	-9.243447e-05	-9.058177e-06
11	-5.759328e-04	1.573468e-04	-1.428364e-04	-7.480729e-04	1.789176e-04
12	-4.753327e-01	2.260174e-02	-1.072862e-02	-1.203793e-01	2.759256e-02
13	2.945821e-04	-3.616851e-05	4.181273e-05	-3.158815e-05	3.180630e-05
14	6.123530e-04	-8.895799e-05	4.725686e-05	-4.388212e-05	8.154947e-06
15	-3.144541e-02	5.813971e-03	7.489267e-02	3.777256e-02	-2.539634e-02
16	-8.168867e-02	-5.609961e-02	8.300009e-03	8.337590e-02	-2.422243e-02
17	-1.725744e-02	3.466590e-03	-1.756090e-03	-7.157733e-03	2.003314e-03
18	1.931786e-01	1.477776e-01	-3.494913e-02	-1.971923e-01	5.798697e-02
19	4.975429e-02	-9.082852e-02	-1.265475e-02	-6.596580e-02	2.137903e-02
20	1.288605e-02	-1.369233e-01	-1.337750e-02	-7.675222e-02	2.571750e-02
21	2.023478e-03	-3.524243e-04	3.450230e-05	1.093528e-04	-6.713336e-04
22	-7.628551e-04	-2.953703e-03	-2.505978e-05	8.529326e-04	-3.802587e-04
23	2.356357e-02	-6.684543e-03	3.147051e-01	-5.670388e-02	-2.351353e-03
24	-8.397247e-03	-4.141703e-03	2.251094e-02	8.507717e-02	2.788994e-01
25	5.937897e-04	3.788494e-05	-1.018064e-04	2.554394e-04	1.172284e-04
26	-9.301541e-03	4.388233e-04	9.595274e-03	-3.138110e-03	-2.260331e-03
27	6.658301e-04	-1.042590e-03	6.032942e-04	-1.826280e-04	-1.478962e-04
28	-4.270117e-03	1.330078e-03	-1.277416e-03	4.337433e-03	-1.124241e-03

	F5	F6	F7	F8	F9	...	\
0	-5.843251e-03	2.506757e-02	1.318098e-02	-9.188044e-03	9.169765e-03	...	
1	-3.069085e-02	1.858856e-01	3.122623e-02	-4.830651e-02	2.349613e-02	...	
2	-3.410321e-17	1.369477e-17	-8.455858e-17	-2.873397e-17	1.112117e-16	...	
3	-1.122061e-02	-3.615212e-02	-1.317931e-01	-9.985970e-02	9.734349e-02	...	
4	-2.208940e-02	2.349706e-02	7.623103e-02	1.911253e-02	-5.668017e-02	...	
5	2.426413e-02	5.313723e-03	8.408471e-03	-2.658404e-02	7.484148e-03	...	
6	-1.870712e-04	-1.695770e-03	-3.017981e-03	3.977274e-03	9.619764e-04	...	
7	9.298736e-03	-3.820817e-02	5.034085e-02	1.491890e-01	1.297761e-01	...	
8	-3.550314e-02	-1.351251e-01	8.667242e-02	-8.286561e-02	-2.105749e-02	...	
9	1.304456e-05	-3.172747e-04	1.851659e-04	-1.252813e-04	-1.480663e-04	...	
10	-3.338444e-05	-2.276942e-05	6.541379e-05	-3.617984e-06	-2.315705e-04	...	
11	-1.285274e-04	1.087749e-04	4.104077e-04	9.511603e-05	-5.336604e-04	...	
12	-1.120646e-02	-1.291513e-03	2.754111e-02	-1.227052e-02	-1.558714e-02	...	
13	-2.757559e-05	-1.126518e-06	2.732598e-05	6.129771e-05	-6.773303e-05	...	
14	-6.882875e-05	8.174751e-05	1.169367e-04	1.824534e-04	-2.289856e-04	...	
15	-2.533262e-01	5.870991e-03	-3.203345e-02	3.574604e-02	-3.534162e-03	...	
16	4.688372e-03	9.990346e-03	1.184655e-02	-2.482445e-02	1.202889e-02	...	
17	-1.472122e-02	-4.819915e-03	-5.941565e-04	-1.909588e-02	1.486500e-02	...	
18	-5.657513e-02	-5.502096e-02	4.021897e-02	-3.668975e-02	9.681293e-03	...	

19	1.294151e-02	-2.079138e-02	-4.141166e-02	4.890547e-02	-8.059047e-02	...
20	1.333951e-02	-1.821614e-02	-1.227347e-01	6.259820e-02	-6.554778e-02	...
21	8.728249e-04	5.881196e-04	7.818498e-04	1.100603e-03	-1.587701e-03	...
22	2.078701e-03	-1.602080e-03	3.179655e-03	1.839315e-03	-2.232483e-03	...
23	5.298844e-02	4.561415e-03	5.264154e-03	-1.442913e-02	-3.718234e-04	...
24	-1.118925e-02	8.627183e-03	6.658504e-03	-4.723738e-03	-5.976611e-03	...
25	-4.178892e-05	-3.889383e-04	-6.189074e-04	1.572754e-04	-8.529726e-05	...
26	3.989833e-04	4.441416e-03	6.528680e-03	-4.768151e-03	-4.821239e-04	...
27	-1.641377e-04	-9.509175e-05	-3.779596e-04	4.072325e-04	-9.225083e-05	...
28	-2.111937e-03	-1.726295e-03	-4.046736e-03	-1.950361e-04	2.083568e-03	...

	F19	F20	F21	F22	F23	\
0	-2.872238e-04	9.995922e-04	6.000130e-05	7.048739e-05	1.299873e-05	
1	3.573153e-04	-1.017217e-04	-7.078149e-05	6.908721e-06	1.300320e-05	
2	3.033947e-17	-1.235581e-17	-5.002031e-18	6.871570e-17	1.310380e-17	
3	-8.226666e-04	2.554189e-06	-7.739896e-06	-3.931897e-06	1.412620e-05	
4	3.000683e-03	-9.718227e-04	2.438291e-04	-1.589518e-06	3.697444e-06	
5	3.312176e-04	1.270784e-04	-7.110516e-06	-3.551083e-07	-1.979688e-05	
6	-2.367155e-04	2.373376e-03	1.016754e-04	-1.054922e-04	-3.261050e-05	
7	2.191824e-04	1.575156e-05	-5.114032e-05	1.186513e-05	7.705093e-06	
8	-9.777711e-05	4.230651e-05	-8.339369e-05	8.882249e-06	-1.918268e-05	
9	-2.180145e-05	-2.671691e-05	2.902315e-05	-1.179498e-04	1.458206e-02	
10	-1.150780e-05	3.108585e-05	6.145779e-05	-5.820314e-06	-6.477999e-06	
11	4.881243e-05	-1.439887e-04	1.856047e-04	5.856177e-05	-3.389371e-05	
12	-1.347805e-04	4.746108e-05	4.989245e-05	5.444401e-06	6.207751e-06	
13	7.867117e-05	5.340418e-04	-5.869045e-06	-2.753366e-05	2.334560e-05	
14	1.045902e-04	5.650907e-04	8.292013e-06	-3.267766e-05	2.495904e-05	
15	-1.139343e-03	1.474313e-04	-1.238600e-04	-3.135093e-06	1.061057e-06	
16	-2.046825e-03	2.682539e-04	-1.303329e-04	-3.402634e-05	-7.221920e-06	
17	1.033653e-02	1.922050e-03	1.162648e-03	-5.831729e-05	-2.965993e-05	
18	-3.746332e-04	1.088299e-04	-6.109317e-05	-1.201435e-05	-6.805298e-07	
19	6.330967e-05	1.077318e-04	-5.438470e-05	-9.528984e-06	6.104850e-08	
20	7.244746e-04	-8.075659e-05	-7.715772e-05	-1.277533e-05	1.627343e-06	
21	2.729678e-03	5.547179e-02	-6.493310e-04	-5.957841e-06	9.407035e-06	
22	-9.089972e-04	2.705533e-04	4.392145e-05	7.547796e-06	-2.953101e-05	
23	5.795443e-04	-5.707405e-05	-7.608067e-05	-1.713895e-05	2.869308e-06	
24	7.160569e-05	5.678977e-05	2.580032e-05	-2.524542e-06	-3.543676e-06	
25	2.081857e-04	4.722108e-06	-5.301697e-05	1.613385e-02	1.064377e-04	
26	2.554145e-03	4.930776e-04	4.099721e-05	7.964247e-04	1.037446e-06	
27	-4.488070e-04	6.648386e-04	5.147126e-02	1.734258e-05	-7.187237e-06	
28	6.570799e-02	-2.549018e-03	1.796589e-04	-7.340835e-05	7.919447e-06	

	F24	F25	F26	F27	F28
0	-1.467895e-05	1.416231e-05	-1.417619e-06	-1.313863e-07	0.0
1	1.344596e-06	-5.292002e-06	1.288565e-06	-1.183070e-07	0.0
2	-1.342215e-17	-5.961478e-18	5.095097e-18	-1.037295e-17	0.0
3	-2.796608e-06	1.001596e-05	7.994276e-07	2.724446e-07	0.0
4	-3.844483e-05	-3.293605e-05	-3.276820e-06	-8.635041e-07	0.0

```

5  4.854748e-06 -3.631622e-06  9.640556e-07  9.631289e-08  0.0
6 -4.145220e-06  1.215635e-04  1.772423e-05 -4.753538e-07  0.0
7 -1.296942e-06  5.493775e-06  3.454629e-06  2.912995e-08  0.0
8  3.401206e-06 -1.063132e-06 -1.919714e-06  6.520396e-08  0.0
9 -2.543870e-05  2.551939e-05  3.534134e-06 -3.702565e-08  0.0
10 1.222307e-03  7.419158e-04  6.929648e-03 -1.838789e-05  0.0
11 4.171002e-04  1.012915e-02 -5.400416e-04 -1.970738e-06  0.0
12 4.184739e-06 -2.628462e-05 -3.511291e-07  1.805591e-07  0.0
13 7.909973e-03 -3.339753e-04 -5.911896e-04 -6.982252e-04  0.0
14 8.312342e-03 -2.991434e-04 -4.292903e-04  6.672173e-04  0.0
15 -1.635094e-06 -5.476494e-07 -6.274980e-07 -7.252238e-08  0.0
16 -1.169071e-05  7.865728e-06 -1.090782e-06  2.637628e-08  0.0
17 2.485576e-05  4.465845e-05  1.058082e-05  7.294712e-07  0.0
18 -9.726407e-06 -1.487652e-05 -8.660789e-07 -7.235635e-08  0.0
19 -4.959719e-06 -5.622730e-06 -1.882364e-06 -3.664319e-08  0.0
20 -9.434544e-06 -5.911233e-06 -1.412224e-06 -2.464731e-07  0.0
21 -1.624504e-04  2.441891e-05  3.770417e-06 -1.055668e-07  0.0
22 -1.927031e-05 -1.407034e-07 -1.108394e-06 -1.556810e-08  0.0
23 -3.967155e-06  7.675932e-07 -7.437185e-07 -2.487767e-08  0.0
24 -6.928842e-07 -2.010724e-06  3.946950e-07  4.697345e-08  0.0
25 2.782085e-05 -3.688502e-05  2.744123e-06  1.319227e-07  0.0
26 2.414818e-05  5.855657e-06  6.150027e-07  5.391813e-07  0.0
27 -5.761885e-06 -3.820437e-05 -6.545405e-06 -1.706299e-07  0.0
28 -1.997573e-05 -1.260418e-05  1.257436e-06 -3.177992e-07  0.0

```

[29 rows x 29 columns]

```

[ ]: #Obtain squared cosines
COS2=L**2
print("Square Cosines")

## COS2 Dataframe ##
COS2_DF = pd.DataFrame(COS2)
print(COS2_DF)

## Create a table ##

```

Square Cosines

	0	1	2	3	4 \
0	6.860238e-06	1.746075e-05	5.303115e-05	1.181626e-04	7.746827e-05
1	4.197721e-03	3.202178e-02	6.121118e-04	8.130406e-03	2.002583e-04
2	3.519483e-33	1.907814e-33	3.321298e-34	2.880017e-34	3.696841e-35
3	3.829752e-06	2.291265e-03	5.317710e-05	4.631354e-04	1.858062e-04
4	6.451077e-03	1.953653e-04	1.963232e-06	9.038723e-05	8.092804e-05
5	2.527931e-05	3.733481e-06	5.984647e-05	7.778438e-05	4.168610e-06
6	8.228021e-05	2.001480e-05	5.184199e-06	8.951793e-05	9.513331e-07
7	2.754753e-06	1.117200e-02	2.418220e-07	1.907964e-03	4.490363e-04
8	1.674302e-04	5.644500e-02	6.578856e-08	5.302892e-05	3.750585e-05

9	5.191865e-09	6.114478e-09	6.621472e-10	7.669862e-08	1.473527e-12
10	3.240136e-09	1.759718e-10	6.707757e-10	8.544132e-09	8.205057e-11
11	3.316986e-07	2.475802e-08	2.040225e-08	5.596130e-07	3.201151e-08
12	2.259412e-01	5.108385e-04	1.151033e-04	1.449118e-02	7.613493e-04
13	8.677861e-08	1.308161e-09	1.748304e-09	9.978113e-10	1.011641e-09
14	3.749762e-07	7.913524e-09	2.233211e-09	1.925640e-09	6.650316e-11
15	9.888137e-04	3.380226e-05	5.608912e-03	1.426766e-03	6.449741e-04
16	6.673038e-03	3.147166e-03	6.889014e-05	6.951541e-03	5.867260e-04
17	2.978192e-04	1.201725e-05	3.083851e-06	5.123315e-05	4.013267e-06
18	3.731797e-02	2.183822e-02	1.221442e-03	3.888479e-02	3.362489e-03
19	2.475489e-03	8.249819e-03	1.601426e-04	4.351486e-03	4.570629e-04
20	1.660504e-04	1.874798e-02	1.789574e-04	5.890904e-03	6.613896e-04
21	4.094464e-06	1.242029e-07	1.190409e-09	1.195803e-08	4.506888e-07
22	5.819479e-07	8.724362e-06	6.279924e-10	7.274941e-07	1.445967e-07
23	5.552417e-04	4.468311e-05	9.903930e-02	3.215329e-03	5.528860e-06
24	7.051376e-05	1.715370e-05	5.067423e-04	7.238125e-03	7.778485e-02
25	3.525862e-07	1.435269e-09	1.036455e-08	6.524930e-08	1.374250e-08
26	8.651866e-05	1.925659e-07	9.206928e-05	9.847735e-06	5.109098e-06
27	4.433297e-07	1.086993e-06	3.639639e-07	3.335299e-08	2.187327e-08
28	1.823390e-05	1.769109e-06	1.631791e-06	1.881332e-05	1.263918e-06

	5	6	7	8	9	...	\
0	3.414358e-05	6.283830e-04	1.737382e-04	8.442016e-05	8.408458e-05	...	
1	9.419283e-04	3.455346e-02	9.750777e-04	2.333519e-03	5.520679e-04	...	
2	1.163029e-33	1.875468e-34	7.150153e-33	8.256411e-34	1.236805e-32	...	
3	1.259022e-04	1.306976e-03	1.736943e-02	9.971960e-03	9.475754e-03	...	
4	4.879418e-04	5.521120e-04	5.811170e-03	3.652887e-04	3.212641e-03	...	
5	5.887479e-04	2.823565e-05	7.070238e-05	7.067112e-04	5.601247e-05	...	
6	3.499564e-08	2.875634e-06	9.108211e-06	1.581870e-05	9.253986e-07	...	
7	8.646649e-05	1.459864e-03	2.534201e-03	2.225737e-02	1.684185e-02	...	
8	1.260473e-03	1.825879e-02	7.512108e-03	6.866710e-03	4.434177e-04	...	
9	1.701607e-10	1.006632e-07	3.428642e-08	1.569541e-08	2.192362e-08	...	
10	1.114521e-09	5.184467e-10	4.278964e-09	1.308980e-11	5.362487e-08	...	
11	1.651930e-08	1.183198e-08	1.684345e-07	9.047059e-09	2.847934e-07	...	
12	1.255848e-04	1.668006e-06	7.585126e-04	1.505658e-04	2.429591e-04	...	
13	7.604129e-10	1.269043e-12	7.467094e-10	3.757409e-09	4.587763e-09	...	
14	4.737397e-09	6.682656e-09	1.367418e-08	3.328925e-08	5.243441e-08	...	
15	6.417418e-02	3.446853e-05	1.026142e-03	1.277779e-03	1.249030e-05	...	
16	2.198083e-05	9.980702e-05	1.403408e-04	6.162531e-04	1.446941e-04	...	
17	2.167144e-04	2.323158e-05	3.530220e-07	3.646526e-04	2.209683e-04	...	
18	3.200745e-03	3.027306e-03	1.617565e-03	1.346138e-03	9.372744e-05	...	
19	1.674827e-04	4.322816e-04	1.714925e-03	2.391745e-03	6.494823e-03	...	
20	1.779424e-04	3.318277e-04	1.506381e-02	3.918535e-03	4.296512e-03	...	
21	7.618232e-07	3.458847e-07	6.112891e-07	1.211327e-06	2.520794e-06	...	
22	4.320997e-06	2.566660e-06	1.011021e-05	3.383079e-06	4.983981e-06	...	
23	2.807775e-03	2.080651e-05	2.771132e-05	2.081996e-04	1.382526e-07	...	
24	1.251993e-04	7.442829e-05	4.433568e-05	2.231370e-05	3.571988e-05	...	
25	1.746314e-09	1.512730e-07	3.830464e-07	2.473554e-08	7.275623e-09	...	

26	1.591877e-07	1.972618e-05	4.262366e-05	2.273526e-05	2.324434e-07	...
27	2.694117e-08	9.042442e-09	1.428535e-07	1.658383e-07	8.510216e-09	...
28	4.460278e-06	2.980093e-06	1.637607e-05	3.803910e-08	4.341255e-06	...

	19	20	21	22	23	\
0	8.249751e-08	9.991845e-07	3.600156e-09	4.968473e-09	1.689670e-10	
1	1.276742e-07	1.034730e-08	5.010020e-09	4.773042e-11	1.690832e-10	
2	9.204837e-34	1.526660e-34	2.502032e-35	4.721847e-33	1.717097e-34	
3	6.767803e-07	6.523881e-12	5.990599e-11	1.545981e-11	1.995496e-10	
4	9.004097e-06	9.444393e-07	5.945263e-08	2.526567e-12	1.367109e-11	
5	1.097051e-07	1.614893e-08	5.055943e-11	1.261019e-13	3.919163e-10	
6	5.603424e-08	5.632913e-06	1.033788e-08	1.112860e-08	1.063444e-09	
7	4.804091e-08	2.481117e-10	2.615332e-09	1.407813e-10	5.936845e-11	
8	9.560363e-09	1.789840e-09	6.954507e-09	7.889435e-11	3.679752e-10	
9	4.753034e-10	7.137935e-10	8.423430e-10	1.391215e-08	2.126364e-04	
10	1.324294e-10	9.663298e-10	3.777060e-09	3.387605e-11	4.196447e-11	
11	2.382653e-09	2.073273e-08	3.444912e-08	3.429481e-09	1.148784e-09	
12	1.816578e-08	2.252554e-09	2.489256e-09	2.964150e-11	3.853618e-11	
13	6.189153e-09	2.852007e-07	3.444569e-11	7.581026e-10	5.450170e-10	
14	1.093911e-08	3.193275e-07	6.875747e-11	1.067830e-09	6.229535e-10	
15	1.298103e-06	2.173598e-08	1.534130e-08	9.828805e-12	1.125841e-12	
16	4.189495e-06	7.196013e-08	1.698665e-08	1.157792e-09	5.215613e-11	
17	1.068438e-04	3.694275e-06	1.351751e-06	3.400906e-09	8.797114e-10	
18	1.403500e-07	1.184395e-08	3.732375e-09	1.443447e-10	4.631208e-13	
19	4.008114e-09	1.160615e-08	2.957696e-09	9.080154e-11	3.726919e-15	
20	5.248635e-07	6.521627e-09	5.953313e-09	1.632091e-10	2.648247e-12	
21	7.451144e-06	3.077120e-03	4.216307e-07	3.549587e-11	8.849231e-11	
22	8.262758e-07	7.319908e-08	1.929094e-09	5.696922e-11	8.720806e-10	
23	3.358715e-07	3.257448e-09	5.788268e-09	2.937437e-10	8.232928e-12	
24	5.127375e-09	3.225077e-09	6.656565e-10	6.373314e-12	1.255764e-11	
25	4.334128e-08	2.229831e-11	2.810799e-09	2.603010e-04	1.132898e-08	
26	6.523659e-06	2.431256e-07	1.680771e-09	6.342923e-07	1.076294e-12	
27	2.014277e-07	4.420103e-07	2.649291e-03	3.007652e-10	5.165638e-11	
28	4.317540e-03	6.497492e-06	3.227731e-08	5.388785e-09	6.271764e-11	

	24	25	26	27	28
0	2.154717e-10	2.005710e-10	2.009643e-12	1.726235e-14	0.0
1	1.807939e-12	2.800528e-11	1.660399e-12	1.399654e-14	0.0
2	1.801542e-34	3.553921e-35	2.596002e-35	1.075980e-34	0.0
3	7.821016e-12	1.003194e-10	6.390845e-13	7.422607e-14	0.0
4	1.478005e-09	1.084783e-09	1.073755e-11	7.456393e-13	0.0
5	2.356858e-11	1.318868e-11	9.294032e-13	9.276174e-15	0.0
6	1.718285e-11	1.477769e-08	3.141483e-10	2.259612e-13	0.0
7	1.682058e-12	3.018156e-11	1.193446e-11	8.485540e-16	0.0
8	1.156820e-11	1.130250e-12	3.685302e-12	4.251557e-15	0.0
9	6.471276e-10	6.512393e-10	1.249010e-11	1.370898e-15	0.0
10	1.494034e-06	5.504390e-07	4.802002e-05	3.381144e-10	0.0
11	1.739726e-07	1.025996e-04	2.916450e-07	3.883807e-12	0.0

12	1.751204e-11	6.908815e-10	1.232916e-13	3.260158e-14	0.0
13	6.256768e-05	1.115395e-07	3.495051e-07	4.875184e-07	0.0
14	6.909503e-05	8.948678e-08	1.842902e-07	4.451790e-07	0.0
15	2.673531e-12	2.999199e-13	3.937537e-13	5.259496e-15	0.0
16	1.366727e-10	6.186968e-11	1.189806e-12	6.957082e-16	0.0
17	6.178088e-10	1.994377e-09	1.119537e-10	5.321283e-13	0.0
18	9.460299e-11	2.213109e-10	7.500926e-13	5.235441e-15	0.0
19	2.459881e-11	3.161510e-11	3.543295e-12	1.342723e-15	0.0
20	8.901063e-11	3.494268e-11	1.994376e-12	6.074896e-14	0.0
21	2.639012e-08	5.962833e-10	1.421604e-11	1.114435e-14	0.0
22	3.713447e-10	1.979744e-14	1.228538e-12	2.423658e-16	0.0
23	1.573832e-11	5.891993e-13	5.531172e-13	6.188983e-16	0.0
24	4.800885e-13	4.043012e-12	1.557842e-13	2.206505e-15	0.0
25	7.739996e-10	1.360505e-09	7.530212e-12	1.740360e-14	0.0
26	5.831348e-10	3.428871e-11	3.782284e-13	2.907165e-13	0.0
27	3.319932e-11	1.459574e-09	4.284233e-11	2.911455e-14	0.0
28	3.990296e-10	1.588654e-10	1.581146e-12	1.009963e-13	0.0

[29 rows x 29 columns]

```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
plt.grid()

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

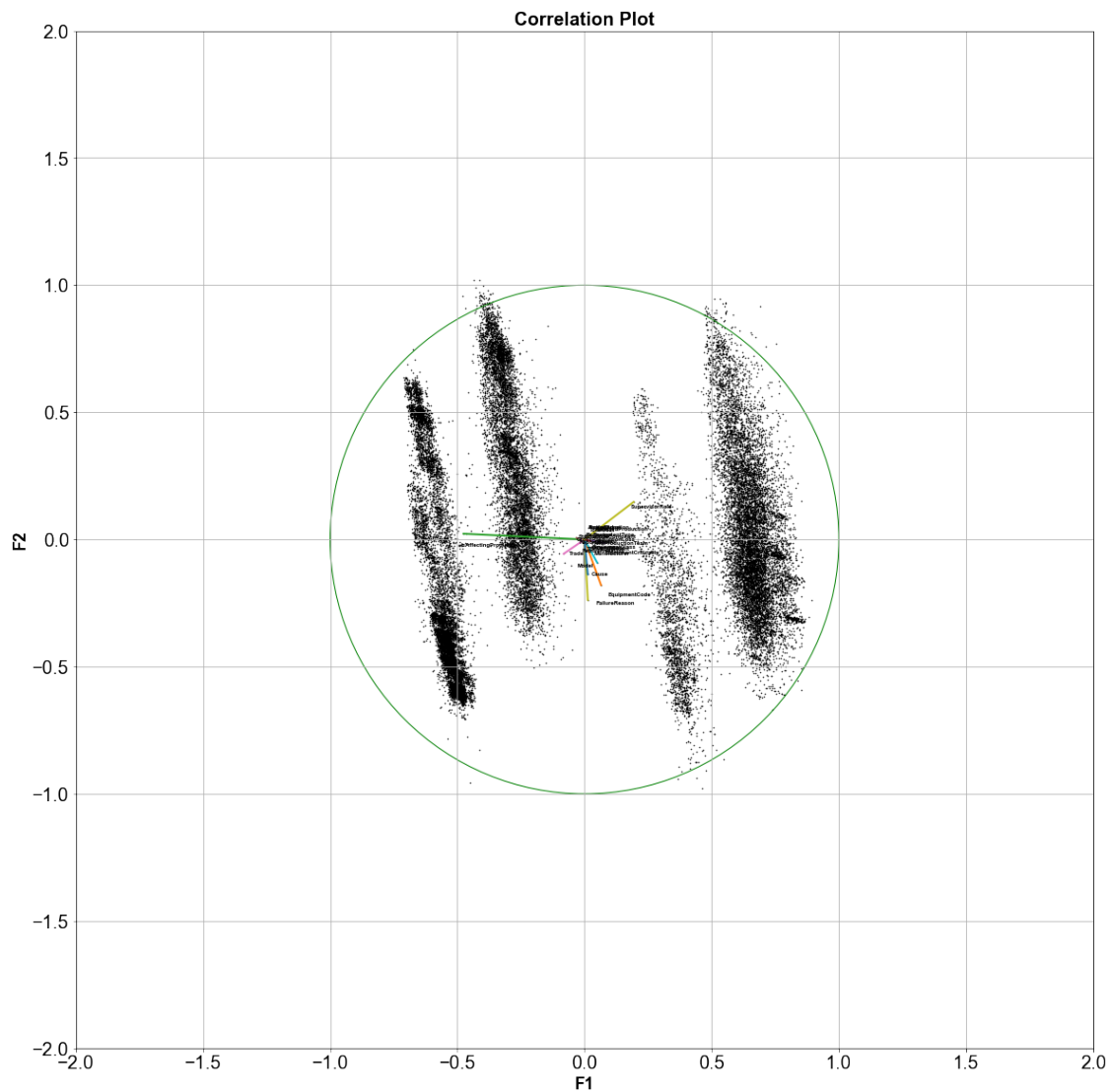
for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
↪05),L[row,1]+random.uniform(-0.05,0.05)),**fontL2)

plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-2,2)
plt.ylim(-2,2)
```

```
plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)
plt.show()
```



```
[ ]: import random

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
```

```

plt.grid()

circ=plt.Circle((0,0), radius=1, color='g', fill=False)
ax.add_patch(circ)
ax.set_aspect('equal')

plt.scatter(L[:,0],L[:,1],marker=".", c = "k" , s = 1)
plt.scatter(F[:,0],F[:,1],marker=".", c = "k" , s = 1)

for row in range(0,len(s)):
    plt.plot([0, L[row,0]], [0,L[row,1]], linewidth=2,label='X1')

for row in range(0,len(s)):
    #ax.annotate('X'+str(row+1), (L[row,0]+0.01,L[row,1]+0.01))
    ax.annotate(header_row[row], (L[row,0]+random.uniform(-0.05,0.
↪05),L[row,1]+random.uniform(-0.05,0.05)),**fontL)

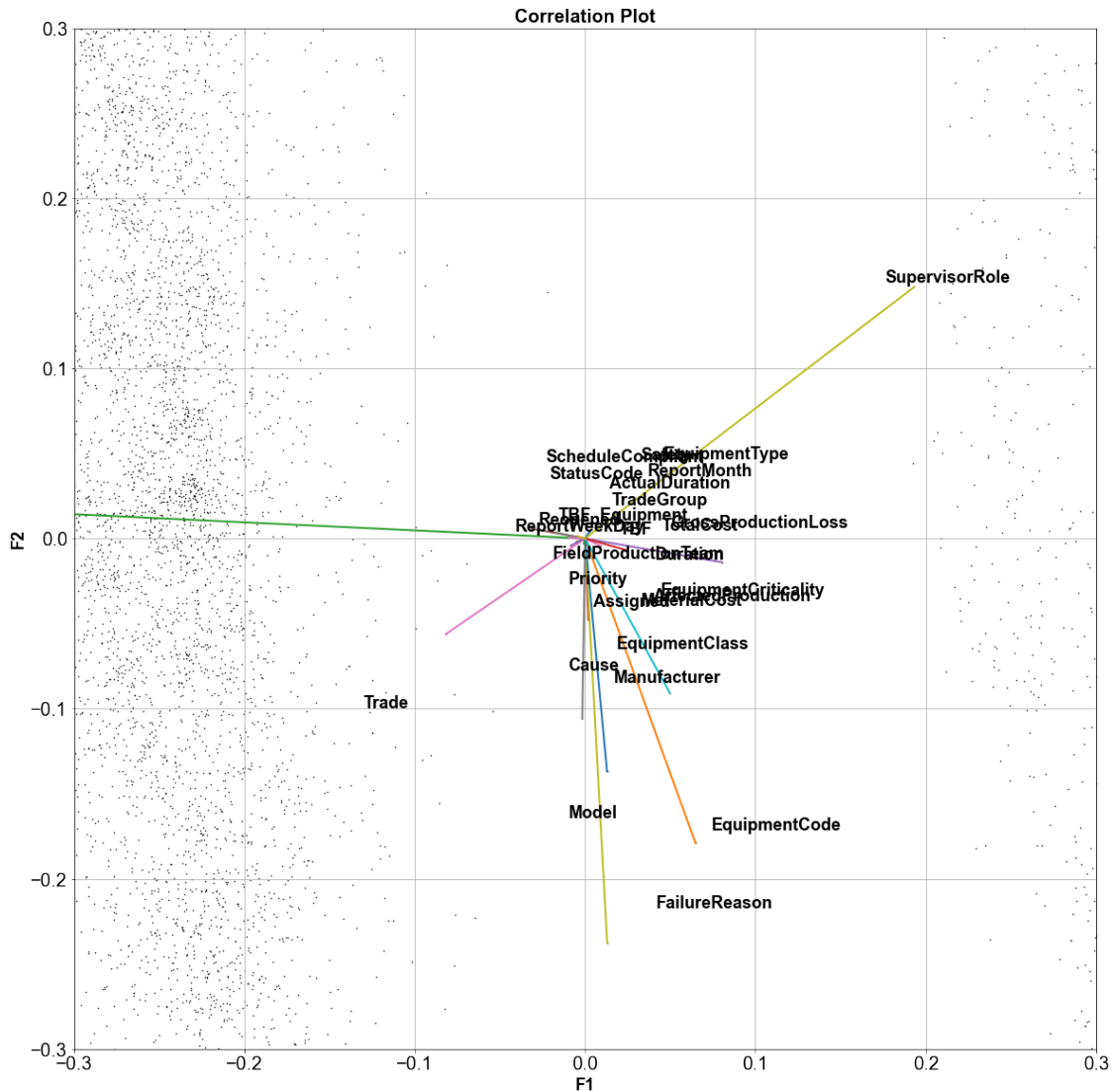
plt.ylabel('F2',**fontL)
plt.xlabel('F1',**fontL)

plt.xlim(-0.3,0.3)
plt.ylim(-0.3,0.3)

plt.xticks(fontsize = 20 , family = "Arial")
plt.yticks(fontsize = 20 , family = "Arial")

plt.title('Correlation Plot',**fontT)
fig.set_size_inches(30, 20)
fig.savefig('PCA_Correlation_Plot.jpg', dpi=300)

```



```
[ ]: pca_pipe = make_pipeline(StandardScaler(), PCA())
pca_pipe.fit(dfPosition)
```

```
modelo_pca = pca_pipe.named_steps['pca']
```

```
[ ]: PCNames = []
for i in range(0, len(columnSystem)):
    nombre = "PC " + str(i)
    PCNames.append(nombre)
```

```
[ ]: # Se convierte el array a dataframe para añadir nombres a los ejes.
pca_table = pd.DataFrame(
```



```

data      = modelo_pca.components_,
columns   = dfSystem.columns,
index     = PCNames
)

print(pca_table)

```

	FieldProductionTeam	EquipmentCode	EquipmentType	EquipmentClass	\
PC 0	3.746909e-02	2.989850e-01	-1.110223e-16	2.720072e-02	
PC 1	4.732885e-02	-2.940602e-01	1.110223e-16	-2.125873e-01	
PC 2	-1.673168e-02	-3.493927e-02	-8.326673e-17	4.865445e-02	
PC 3	-5.656887e-02	-1.064304e-01	-6.938894e-18	4.362374e-02	
PC 4	-1.995344e-01	-4.085242e-02	6.245005e-17	3.090361e-01	
PC 5	-2.422474e-01	-8.262165e-03	1.127570e-16	-4.908143e-01	
PC 6	-1.971618e-01	-2.362602e-02	8.326673e-17	7.308586e-03	
PC 7	-4.596410e-01	5.276703e-02	1.110223e-16	-3.456084e-03	
PC 8	3.868372e-01	2.217648e-01	3.330669e-16	-1.001915e-02	
PC 9	9.258335e-02	-8.618054e-02	-1.387779e-17	-2.839366e-01	
PC 10	-9.982237e-02	-1.264732e-01	-2.359224e-16	6.887984e-02	
PC 11	8.322477e-02	2.516669e-02	-9.367507e-17	1.347614e-01	
PC 12	2.685479e-03	1.507473e-02	-2.331035e-17	1.024378e-01	
PC 13	-1.925788e-02	-1.035641e-01	-5.698567e-16	-3.495843e-02	
PC 14	-6.482040e-02	2.162963e-02	2.914335e-16	2.212309e-01	
PC 15	-6.943146e-02	-1.092964e-01	1.353084e-16	5.059067e-02	
PC 16	5.948527e-01	1.485594e-01	8.326673e-17	2.538377e-02	
PC 17	-8.601766e-02	-9.412835e-02	7.695125e-17	2.875370e-01	
PC 18	-8.686984e-02	-2.042456e-01	-4.510281e-17	-1.241419e-01	
PC 19	-6.281450e-02	-1.276455e-02	-1.040834e-17	9.220955e-02	
PC 20	2.015157e-02	-2.689028e-02	3.243933e-16	-3.810279e-01	
PC 21	-2.354525e-01	7.237276e-01	2.220446e-16	-3.335978e-02	
PC 22	1.330858e-01	-2.751512e-01	-2.498002e-16	1.959772e-01	
PC 23	-1.074885e-02	1.616243e-01	-9.020562e-17	-1.711341e-01	
PC 24	7.130039e-02	7.302081e-02	1.110223e-16	2.967921e-02	
PC 25	8.626138e-02	-5.524796e-02	-1.110223e-16	-3.457194e-01	
PC 26	-2.801084e-02	4.856614e-02	2.775558e-16	-6.929620e-02	
PC 27	-2.834359e-03	-4.369648e-03	3.053113e-16	7.487925e-03	
PC 28	-2.640135e-17	1.365233e-16	-1.000000e+00	-3.001686e-16	

	EquipmentCriticality	StatusCode	Priority	Cause	\
PC 0	2.988350e-01	-3.593970e-02	8.415490e-02	1.701879e-01	
PC 1	1.229504e-01	-1.895207e-02	2.318389e-03	-3.023659e-01	
PC 2	-1.293728e-01	1.405873e-02	-2.866906e-02	3.323545e-02	
PC 3	-1.679021e-01	-5.127013e-02	6.386352e-02	-5.687986e-03	
PC 4	-4.717898e-01	-5.227980e-02	-4.546026e-01	-5.586463e-02	
PC 5	1.823489e-01	1.101707e-02	-3.761574e-01	1.235235e-01	
PC 6	1.773616e-02	-5.006889e-01	-3.022060e-02	3.866131e-02	
PC 7	1.335252e-01	2.367752e-01	-1.494848e-01	-1.331449e-01	
PC 8	1.826118e-01	-7.445758e-04	-1.128267e-01	-8.193832e-02	

PC 9	-9.923128e-02	-2.773055e-01	4.316329e-02	2.200863e-01
PC 10	-7.043046e-02	3.486867e-01	1.057507e-01	-5.300293e-03
PC 11	1.705728e-02	-4.333992e-02	4.605473e-02	-1.493590e-01
PC 12	5.116564e-02	-2.966710e-02	-4.956482e-02	-7.115692e-02
PC 13	-6.902409e-02	1.483483e-01	1.323960e-01	4.554627e-02
PC 14	4.272534e-02	2.963731e-01	5.329049e-02	-2.404015e-01
PC 15	-3.887586e-02	-4.032800e-01	1.631512e-01	-5.007006e-02
PC 16	-1.136010e-01	1.053004e-01	-9.475349e-02	2.101140e-02
PC 17	6.988585e-02	-1.924324e-01	4.283774e-01	-2.726664e-01
PC 18	-2.934762e-02	3.599702e-01	3.945387e-01	4.237580e-01
PC 19	1.196581e-02	-1.748366e-01	1.714103e-01	4.351020e-01
PC 20	5.408801e-02	-4.866261e-02	2.341373e-01	-4.057387e-01
PC 21	-1.779435e-02	-1.522836e-02	1.131419e-01	1.853639e-02
PC 22	2.472805e-01	-2.442023e-02	-2.195740e-01	1.011044e-01
PC 23	-3.225247e-01	1.088008e-02	1.602788e-01	-2.195208e-01
PC 24	-4.137126e-01	6.148664e-03	8.460270e-02	1.489621e-01
PC 25	-3.545902e-01	2.344221e-02	-5.753067e-02	-8.186661e-02
PC 26	-1.815202e-01	5.032461e-03	9.849464e-02	1.520647e-03
PC 27	-2.145258e-02	2.336303e-03	-5.899597e-03	9.020908e-04
PC 28	7.191389e-17	-2.013708e-16	1.376525e-16	4.544751e-18

	FailureReason	Duration	...	Manufacturer	Model \
PC 0	1.857244e-01	-4.470657e-03	...	3.560189e-01	2.869432e-01
PC 1	-4.299808e-01	-4.744999e-03	...	-2.732157e-01	-3.880734e-01
PC 2	8.729729e-02	6.086941e-03	...	-8.159080e-02	8.046022e-03
PC 3	-2.599926e-02	-1.544153e-02	...	-4.496529e-03	5.628876e-02
PC 4	-1.618257e-01	-5.952166e-02	...	2.122258e-01	1.866434e-01
PC 5	3.038512e-02	-1.223019e-02	...	5.652277e-02	-5.169204e-02
PC 6	5.593807e-02	-6.672213e-02	...	4.826431e-04	3.094444e-03
PC 7	1.521395e-01	2.962313e-01	...	-1.531474e-02	-1.396420e-01
PC 8	7.091775e-02	-8.221141e-02	...	-4.726280e-02	-7.517876e-02
PC 9	-6.597777e-02	1.929459e-01	...	1.136830e-01	3.975563e-03
PC 10	-3.342051e-02	3.205987e-01	...	8.392534e-03	5.216019e-02
PC 11	7.322225e-02	5.600884e-01	...	-1.584890e-02	-2.124194e-02
PC 12	4.954671e-02	4.196622e-01	...	1.071209e-02	1.684834e-02
PC 13	-4.570875e-02	1.828908e-01	...	3.321936e-02	9.730681e-02
PC 14	8.196888e-02	-4.476265e-01	...	-3.656212e-02	-4.850011e-02
PC 15	-2.229886e-02	4.966279e-02	...	-2.442687e-02	5.396028e-02
PC 16	-8.272824e-02	1.485332e-01	...	-4.071127e-02	-8.591280e-03
PC 17	-2.731991e-02	-8.022218e-03	...	-9.443655e-02	1.624546e-01
PC 18	-1.042084e-01	-6.569594e-02	...	3.760001e-02	1.228528e-01
PC 19	1.672487e-01	-1.548419e-02	...	-2.574516e-01	-2.818034e-01
PC 20	-1.218322e-01	-3.304292e-02	...	4.568232e-01	1.754289e-01
PC 21	-4.298803e-01	4.931273e-02	...	-2.912263e-01	6.073645e-02
PC 22	1.450900e-02	1.074654e-02	...	-5.357062e-02	1.218063e-01
PC 23	5.978480e-01	-1.938131e-02	...	1.797458e-02	-3.530054e-01
PC 24	-2.491467e-01	1.417230e-03	...	3.577489e-01	-3.684148e-01
PC 25	1.589008e-01	-1.465510e-02	...	-4.690464e-01	4.903984e-01

PC 26	1.713449e-02	-6.630742e-04	...	-4.327298e-02	1.085908e-01
PC 27	2.600000e-03	-7.100493e-05	...	-9.207183e-04	-7.687605e-03
PC 28	8.883472e-17	-3.013774e-17	...	-4.382864e-17	8.854597e-17

	Safety	Reopened	ReportMonth	ReportWeekDay	\
PC 0	7.502744e-02	3.837622e-03	3.866607e-02	-2.295918e-02	
PC 1	4.851702e-02	-3.008843e-02	1.222294e-02	-1.874381e-02	
PC 2	5.273279e-02	3.010743e-02	-1.557926e-02	2.466999e-02	
PC 3	6.380606e-02	-9.959361e-03	-1.224694e-01	4.266941e-02	
PC 4	1.944024e-02	9.384636e-03	-4.636073e-02	-8.464960e-02	
PC 5	2.393778e-01	1.113427e-01	2.132537e-02	-9.152017e-02	
PC 6	4.554355e-02	1.943830e-02	5.056397e-01	1.715229e-02	
PC 7	1.294691e-01	1.750941e-01	-4.084931e-02	-1.979896e-01	
PC 8	-3.343183e-01	-1.518957e-01	-2.376128e-01	-8.177282e-03	
PC 9	-3.779383e-01	6.369499e-01	-3.821026e-02	2.370607e-01	
PC 10	-1.971118e-01	-1.485389e-01	3.916303e-01	4.914571e-01	
PC 11	-4.392247e-02	2.786017e-01	-1.909350e-01	-2.709965e-01	
PC 12	2.351907e-01	-1.442553e-01	-6.413087e-02	4.082512e-01	
PC 13	-3.997680e-01	-1.509988e-01	3.453496e-01	-5.948811e-01	
PC 14	-3.785579e-02	5.196764e-01	1.864571e-01	1.004520e-01	
PC 15	1.756874e-01	-9.470201e-02	-1.682124e-01	-3.635781e-02	
PC 16	5.149063e-01	2.082550e-01	3.863885e-01	-1.393080e-01	
PC 17	1.413869e-01	1.760978e-01	-5.191170e-02	-9.841511e-02	
PC 18	2.567771e-01	7.235770e-02	-2.439751e-01	-5.654952e-02	
PC 19	6.063601e-02	-8.734084e-02	1.941664e-01	-5.093312e-02	
PC 20	8.298777e-02	-1.086813e-01	1.147429e-01	1.441375e-02	
PC 21	-4.081229e-02	3.218085e-02	-1.712311e-02	4.422518e-02	
PC 22	-2.278402e-02	3.678738e-02	-1.036436e-01	8.233363e-03	
PC 23	2.126177e-02	-3.718600e-02	-5.241065e-02	1.738501e-02	
PC 24	3.586218e-02	-2.077961e-02	-1.050324e-01	-3.206704e-02	
PC 25	1.336599e-02	-1.235616e-02	-4.221743e-02	3.359310e-04	
PC 26	1.063966e-02	6.190111e-03	2.849890e-04	2.893258e-03	
PC 27	-7.722107e-04	-2.116975e-04	-1.043994e-03	1.756251e-03	
PC 28	9.303113e-17	4.308706e-17	1.454851e-17	-1.348181e-16	

	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
PC 0	4.512260e-02	-8.955784e-02	2.481955e-02	-6.513816e-02
PC 1	1.937892e-02	-3.731481e-02	-1.514680e-02	6.174954e-03
PC 2	-1.038783e-02	1.883606e-02	-2.099662e-03	6.045744e-02
PC 3	5.762490e-01	-6.423385e-01	2.794816e-02	3.098645e-01
PC 4	-4.109680e-02	3.010564e-02	1.781449e-02	3.821188e-02
PC 5	-9.093564e-02	3.604990e-02	8.614806e-02	3.215808e-02
PC 6	9.770180e-02	1.078639e-03	9.885679e-02	6.603926e-02
PC 7	3.156138e-01	-1.112263e-01	-1.180535e-01	-3.356397e-01
PC 8	1.740016e-01	-1.463702e-01	7.523048e-03	-1.081903e-01
PC 9	5.793327e-02	-5.044105e-02	-1.889136e-01	-1.397739e-01
PC 10	7.739591e-02	1.424491e-03	3.751275e-01	-2.103083e-01
PC 11	-1.739308e-01	6.209134e-02	4.404473e-01	4.179279e-01

PC 12	-1.101889e-01	5.237641e-02	-6.445214e-01	2.688290e-01
PC 13	3.421787e-02	1.023627e-02	-3.933599e-01	2.799259e-02
PC 14	-2.970672e-02	5.284319e-02	-1.303290e-01	2.330692e-01
PC 15	-1.431290e-02	1.007900e-03	4.229707e-02	-3.326628e-01
PC 16	1.157667e-01	-9.116047e-02	-1.217374e-02	-1.719556e-01
PC 17	-1.874977e-02	4.679300e-02	-6.085619e-02	-2.698699e-01
PC 18	3.610351e-03	4.022067e-02	1.829495e-02	4.133311e-02
PC 19	7.484687e-02	7.050720e-02	2.036379e-02	2.950810e-01
PC 20	1.900329e-01	1.787415e-01	4.669745e-02	2.814962e-01
PC 21	1.884760e-01	1.806465e-01	4.885697e-03	7.516825e-02
PC 22	5.767764e-01	5.903781e-01	1.419470e-02	3.298186e-02
PC 23	1.516297e-01	1.885267e-01	-7.374275e-03	-5.001630e-02
PC 24	1.182101e-01	2.156868e-01	-1.881972e-02	-9.041542e-02
PC 25	3.677543e-02	9.755776e-02	-8.712742e-03	-2.359883e-03
PC 26	5.821620e-03	8.786221e-02	-7.352495e-03	-2.273337e-02
PC 27	3.403533e-04	8.395672e-03	-1.121081e-03	-2.799504e-03
PC 28	-1.447270e-17	-2.271280e-18	-5.343692e-17	8.949664e-17

[29 rows x 29 columns]

```
[ ]: dfPosition.columns
```

```
[ ]: Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
          'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
          'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
          'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
          'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
          'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
          'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
          'TBF_Equipment'],
          dtype='object')
```

```
[ ]: commonPCs = []
for i in range(0, len(maxPCvars)):
    name = maxPCvars[i]
    maxPCAit = getMaxComponent(pca_table, name)
    commonPCs.append(maxPCAit)

dictPCs = dict(zip(maxPCvars, commonPCs))

print(dictPCs)
```

```
{'TBF_Equipment': [3, 11, 12, 14, 19, 20], 'ActualDuration': [], 'TotalCost':
[2, 27], 'IsAffectingProduction': [2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16,
17, 18, 20, 21, 22, 23, 26, 27, 28]}
```

```
[ ]: import heapq
```

```

for i in range(0,len(dictPCs)):
    name = maxPCvars[i]
    print(name)
    PC_num = dictPCs[name]
    print(PC_num)
    for j in range(0,len(PC_num)):
        print(PC_num[j])
        PCval = pca_table.iloc[j]
        mean = PCval.mean()
        largest = heapq.nlargest(3, enumerate(PCval), key=lambda x: x[1])
        lowest = heapq.nsmallest(3, enumerate(PCval), key=lambda x: x[1])
        print("---Highest---")
        for k in range(0,len(largest)):
            indexTabla = largest[k][0]
            print(pca_table.columns[indexTabla])
        print("---Lowest---")
        for k in range(0,len(lowest)):
            indexTabla = lowest[k][0]
            print(pca_table.columns[indexTabla])

```

```

TBF_Equipment
[3, 11, 12, 14, 19, 20]
3
---Highest---
Manufacturer
EquipmentCode
EquipmentCriticality
---Lowest---
IsAffectingProduction
Trade
TradeGroup
11
---Highest---
SupervisorRole
TotalCost
MaterialCost
---Lowest---
FailureReason
Model
Cause
12
---Highest---
MaterialCost
TotalCost
Trade
---Lowest---
SupervisorRole
EquipmentCriticality

```

Manufacturer
 14
 ---Highest---
 ActualDuration
 TBF_Equipment
 Priority
 ---Lowest---
 ScheduleCompliant
 TradeGroup
 EquipmentCriticality
 19
 ---Highest---
 EquipmentClass
 SupervisorRole
 AffectedProduction
 ---Lowest---
 EquipmentCriticality
 Priority
 Trade
 20
 ---Highest---
 AffectedProduction
 Safety
 GrossProductionLoss
 ---Lowest---
 EquipmentClass
 Priority
 TradeGroup
 ActualDuration
 []
 TotalCost
 [2, 27]
 2
 ---Highest---
 Manufacturer
 EquipmentCode
 EquipmentCriticality
 ---Lowest---
 IsAffectingProduction
 Trade
 TradeGroup
 27
 ---Highest---
 SupervisorRole
 TotalCost
 MaterialCost
 ---Lowest---
 FailureReason

Model
 Cause
 IsAffectingProduction
 [2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 26, 27, 28]
 2
 ---Highest---
 Manufacturer
 EquipmentCode
 EquipmentCriticality
 ---Lowest---
 IsAffectingProduction
 Trade
 TradeGroup
 3
 ---Highest---
 SupervisorRole
 TotalCost
 MaterialCost
 ---Lowest---
 FailureReason
 Model
 Cause
 4
 ---Highest---
 MaterialCost
 TotalCost
 Trade
 ---Lowest---
 SupervisorRole
 EquipmentCriticality
 Manufacturer
 5
 ---Highest---
 ActualDuration
 TBF_Equipment
 Priority
 ---Lowest---
 ScheduleCompliant
 TradeGroup
 EquipmentCriticality
 6
 ---Highest---
 EquipmentClass
 SupervisorRole
 AffectedProduction
 ---Lowest---
 EquipmentCriticality

Priority
 Trade
 8
 ---Highest---
 AffectedProduction
 Safety
 GrossProductionLoss
 ---Lowest---
 EquipmentClass
 Priority
 TradeGroup
 9
 ---Highest---
 Assigned
 ReportMonth
 TBF
 ---Lowest---
 StatusCode
 FieldProductionTeam
 Duration
 10
 ---Highest---
 TradeGroup
 ActualDuration
 Duration
 ---Lowest---
 FieldProductionTeam
 TBF_Equipment
 ReportWeekDay
 11
 ---Highest---
 AffectedProduction
 FieldProductionTeam
 GrossProductionLoss
 ---Lowest---
 Safety
 ReportMonth
 Reopened
 12
 ---Highest---
 Reopened
 ReportWeekDay
 Cause
 ---Lowest---
 Safety
 EquipmentClass
 StatusCode
 13


```

---Highest---
ReportWeekDay
ReportMonth
TBF
---Lowest---
TBF_Equipment
Safety
Reopened
15
---Highest---
Duration
TBF
TBF_Equipment
---Lowest---
ReportWeekDay
ReportMonth
ActualDuration
16
---Highest---
Duration
ReportWeekDay
TBF_Equipment
---Lowest---
TBF
Reopened
ActualDuration
17
---Highest---
ReportMonth
GrossProductionLoss
Duration
---Lowest---
ReportWeekDay
Safety
TBF
18
---Highest---
Reopened
GrossProductionLoss
StatusCode
---Lowest---
Duration
Cause
IsAffectingProduction
20
---Highest---
GrossProductionLoss
Safety

```

Priority
 ---Lowest---
 StatusCode
 TBF_Equipment
 AffectedProduction
 21
 ---Highest---
 FieldProductionTeam
 Safety
 ReportMonth
 ---Lowest---
 TBF_Equipment
 ReportWeekDay
 EquipmentCriticality
 22
 ---Highest---
 AffectedProduction
 Priority
 EquipmentClass
 ---Lowest---
 Cause
 TBF_Equipment
 GrossProductionLoss
 23
 ---Highest---
 Cause
 Priority
 StatusCode
 ---Lowest---
 ReportMonth
 Trade
 EquipmentCode
 26
 ---Highest---
 Cause
 TBF_Equipment
 TradeGroup
 ---Lowest---
 Assigned
 Model
 Manufacturer
 27
 ---Highest---
 Manufacturer
 TradeGroup
 TBF_Equipment
 ---Lowest---
 Cause

```

EquipmentClass
Assigned
28
---Highest---
EquipmentCode
ActualDuration
ScheduleCompliant
---Lowest---
FailureReason
Manufacturer
FieldProductionTeam

```

5 Regression

```

[ ]: column_names = dfPosition_scaled.columns
    print(column_names)

Index(['FieldProductionTeam', 'EquipmentCode', 'EquipmentType',
      'EquipmentClass', 'EquipmentCriticality', 'StatusCode', 'Priority',
      'Cause', 'FailureReason', 'Duration', 'GrossProductionLoss',
      'AffectedProduction', 'IsAffectingProduction', 'MaterialCost',
      'TotalCost', 'Assigned', 'Trade', 'TradeGroup', 'SupervisorRole',
      'Manufacturer', 'Model', 'Safety', 'Reopened', 'ReportMonth',
      'ReportWeekDay', 'ActualDuration', 'ScheduleCompliant', 'TBF',
      'TBF_Equipment'],
      dtype='object')

```

```

[ ]: ## Target ##
    y = dfPosition_scaled['TotalCost']
    y.head()

```

```

[ ]: 0    0.002273
    1    0.003385
    2    0.007779
    3    0.000952
    4    0.004194
    Name: TotalCost, dtype: float64

```

```

[ ]: ## Rest of Variables ##
    x = dfPosition_scaled[column_names.drop(['TotalCost'])]
    x.head()

```

```

[ ]:   FieldProductionTeam  EquipmentCode  EquipmentType  EquipmentClass \
0                    0.60      0.459407             0.0           0.6875
1                    0.76      0.314759             0.0           0.3750
2                    0.76      0.778964             0.0           0.3750
3                    0.90      0.541377             0.0           0.6875

```

4		0.76	0.314024		0.0	0.3750
---	--	------	----------	--	-----	--------

	EquipmentCriticality	StatusCode	Priority	Cause	FailureReason	\
0	0.333333	0.0	0.6	0.583333	0.623563	
1	0.333333	0.0	0.4	0.458333	0.221264	
2	0.333333	0.0	0.6	0.458333	0.221264	
3	0.333333	0.0	0.8	0.458333	0.813218	
4	0.166667	0.0	0.2	0.458333	0.221264	

	Duration	...	Manufacturer	Model	Safety	Reopened	ReportMonth	\
0	0.000000	...	0.582031	0.876676	0.0	0.0	1.0	
1	0.000000	...	0.593750	0.882038	0.0	0.0	0.0	
2	0.003633	...	0.335938	0.882038	0.0	0.0	1.0	
3	0.000000	...	0.582031	0.876676	0.0	0.0	1.0	
4	0.000000	...	0.992188	0.882038	0.0	0.0	0.0	

	ReportWeekDay	ActualDuration	ScheduleCompliant	TBF	TBF_Equipment
0	0.833333	0.039578		0.0 0.0	0.787879
1	0.666667	0.005277		0.0 0.0	0.086997
2	0.333333	0.017150		0.0 0.0	0.894372
3	0.166667	0.018470		0.0 0.0	0.086997
4	0.833333	0.003958		0.0 0.0	0.022511

[5 rows x 28 columns]

```
[ ]: #Separate train and test data
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
↪20,random_state=0)
```

```
[ ]: print("Size of the full data set: ",x.shape)
print("Size of the training data set: ",x_train.shape)
print("Size of the test data set: ",x_test.shape)
```

Size of the full data set: (32959, 28)
Size of the training data set: (26367, 28)
Size of the test data set: (6592, 28)

```
[ ]: EquipmentCode = dfPosition_scaled["EquipmentCode"]
EquipmentCode.head()
```

```
[ ]: 0    0.459407
1    0.314759
2    0.778964
3    0.541377
4    0.314024
Name: EquipmentCode, dtype: float64
```

```
[ ]: TotalCost_real = dfPosition_scaled["TotalCost"]
TotalCost_real.head()
```

```
[ ]: 0    0.002273
     1    0.003385
     2    0.007779
     3    0.000952
     4    0.004194
     Name: TotalCost, dtype: float64
```

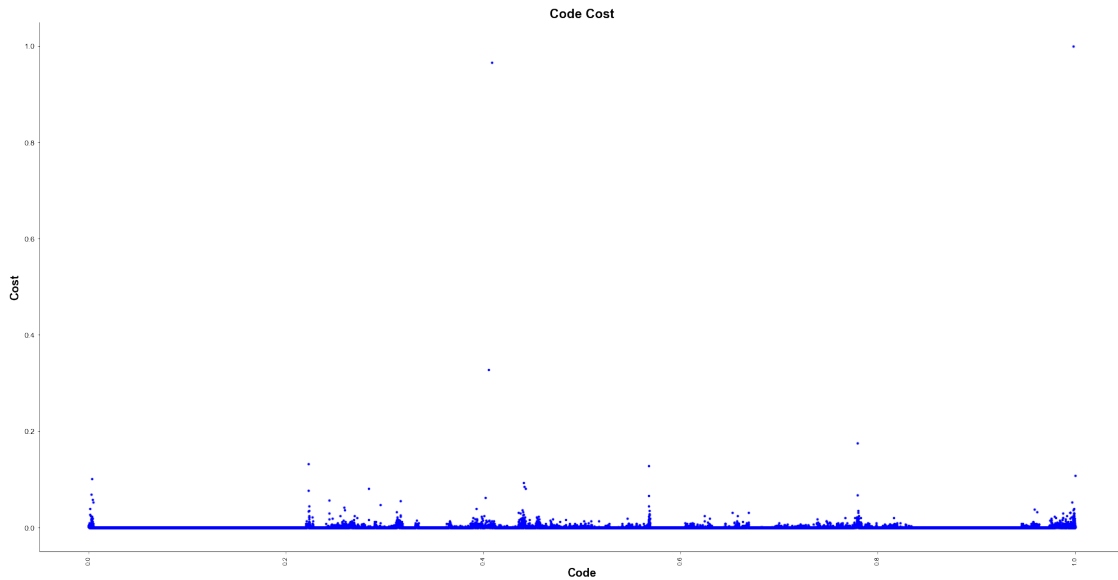
```
[ ]: df_code_cost = pd.DataFrame()

df_code_cost["Code"] = EquipmentCode
df_code_cost["Cost"] = TotalCost_real

df_code_cost.head()
```

```
[ ]:      Code      Cost
0  0.459407  0.002273
1  0.314759  0.003385
2  0.778964  0.007779
3  0.541377  0.000952
4  0.314024  0.004194
```

```
[ ]: ## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c =_
↪"blue" , s = 30)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```



```
[ ]: # importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
# fitting the training data
LR.fit(x_train,y_train)
```

```
[ ]: LinearRegression()
```

```
[ ]: y_prediction = LR.predict(x_test)
y_prediction
```

```
[ ]: array([ 2.07622540e-03, -5.83182567e-05,  5.62005330e-07, ...,
            2.39514910e-04,  4.30838412e-04,  7.99649317e-04])
```

```
[ ]: # importing r2_score module
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

score = r2_score(y_test,y_prediction)
print("R2 score is ",score)
print("Mean squared error is ",mean_squared_error(y_test,y_prediction))
print("Mean absolute error is ",mean_absolute_error(y_test,y_prediction))
print("Root mean squared error is ",np.
      ↳sqrt(mean_squared_error(y_test,y_prediction)))
```

R2 score is 0.7031787614111499

Mean squared error is 2.9787651111828477e-06
Mean absolute error is 0.00042381622117936447
Root mean squared error is 0.0017259099371586132

```
[ ]: y_prediction = LR.predict(x)

## Graph ##
plt.figure(figsize=(30,15))
plt.title("Code Cost",**fontT)
plt.xlabel("Code",**fontL)
plt.ylabel("Cost",**fontL)
plt.xticks(fontsize = 10 , family = "Arial",rotation=90)
plt.yticks(fontsize = 12 , family = "Arial")
plt.scatter(x = df_code_cost["Code"],y = df_code_cost["Cost"], marker=".", c = "blue", s = 40)
plt.scatter(x = df_code_cost["Code"],y = y_prediction, marker=".", c = "green", s = 40)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.legend(['Real Score', 'Model Score'])
plt.show()
```

