Tecnológico de Monterrey

Data Science and Big Data Analytics (Group 1)

Dr. Raúl Valente Ramírez-Velarde

Project 03
Chevron Equipment Maintenance Data

Team 06:

Maximiliano Contreras A00825377

Alejandro Bañuelos A01244596

Aline Martínez A00826215

Marco Ortiz A00823250

Monterrey, Nuevo León, México. November 28th, 2021

Project Report

First of all, we clean the data to drop the columns that would not provide us with useful information, such as columns with a single unique value or very detailed descriptions that would not contribute to the model.

After cleaning, we end up with the next 30 variables: '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'.

```
[ ]: 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')
```

We focused on the types of equipment to carry out the total cost analysis on each of them.

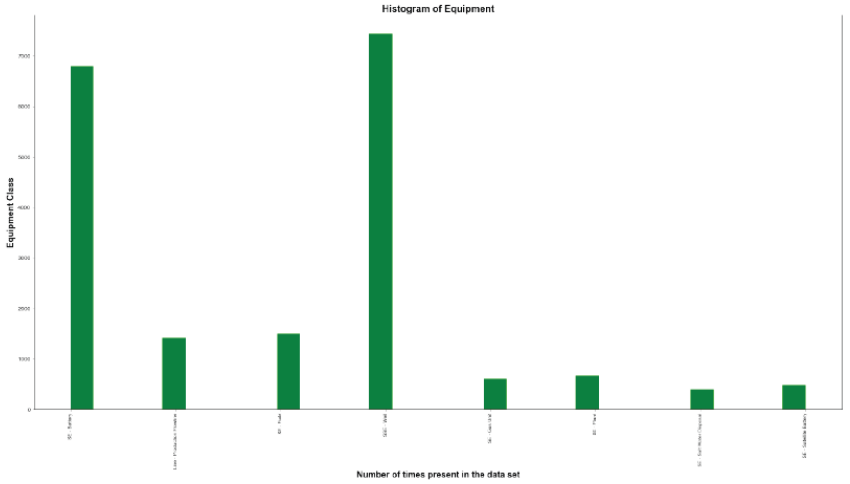
```
[ ]: ## In this case we will separate the dataset by the type of equipment ##

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

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

Equipment Type = System

We created a histogram to see how many times the equipment class was repeating in our data.
By selecting the equipment class that was repeated more than 200 times, we obtained the following.

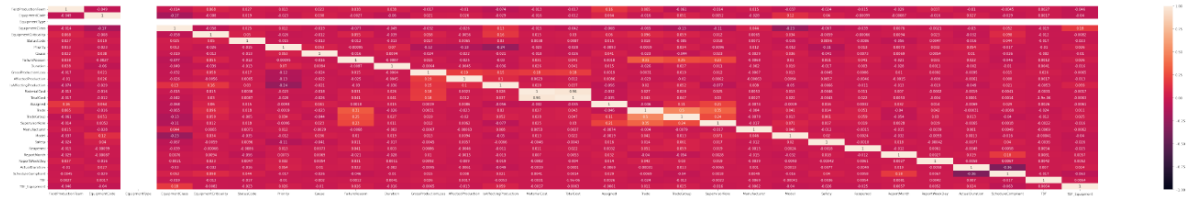
```
[ ]: createGraphs(dfSystem)
```



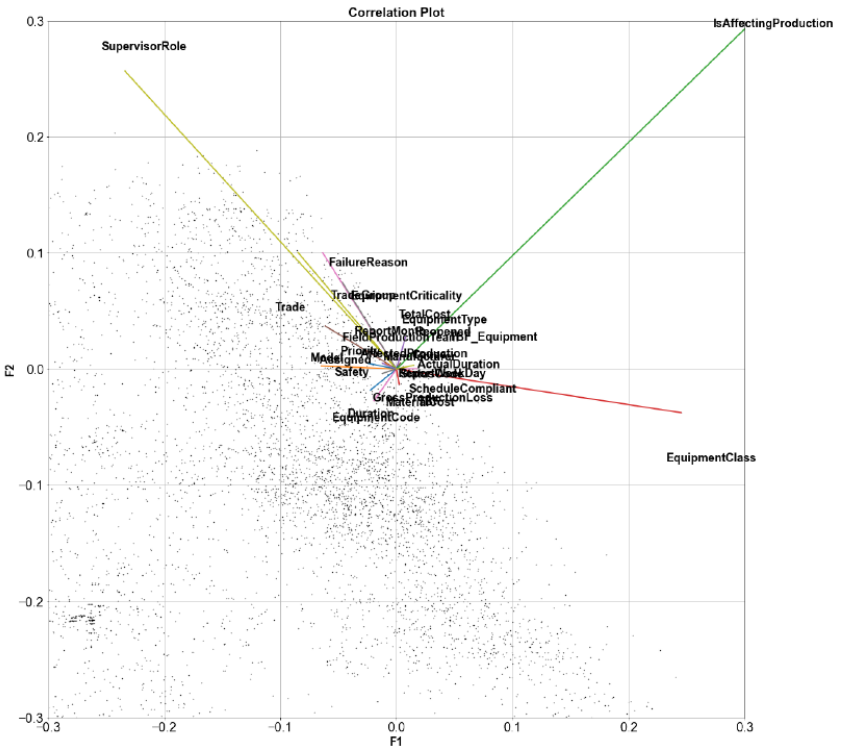
These are the equipment classes we will use for the following analysis since they're the ones with more information.

After standardizing the data, we plotted a correlation matrix for this system equipment class.

```
[ ]: ##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)
```



After creating the PCA, we got something like this:



Where we can see that we got a lot of variables highly correlated to each other.

We created a dataframe with the Principal Components factors for each variable in our data.

```
[ ]: # Se combierte 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
```

In this way, we can somehow identify similarities or similar behaviors in the variables that we want to analyze, in this case the Total Cost.

We created a function to get the maximum components affecting each variable. We can then call this function with the dataframe and the variable of our interest and the function will return the PCs that are equal to or higher than 8 times the mean value.

```
[ ]: 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)
```

We created a python dictionary to obtain the maximum PCs affecting the TBF_Equipment, ActualDuration, TotalCost and IsAffectingProduction variables.

```
[ ]: 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]}
```

Then we looked for the 3 maximum and the 3 minimum PC values for each of the 4 variables mentioned above.

```
[ ]: 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])
```

Looking at our variable of interest (TotalCost), we found that the PC 0, PC 1 and PC 27 we obtained from the Principal Component Analysis are the ones that are affecting the TotalCost variable the most.

```
TotalCost
[0, 1, 27]
```

In the PC 0, the variables TotalCost, MaterialCost and Trade are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the EquipmentClass, Priority and ReportMonth variables.

```
0
---Highest---
TotalCost
MaterialCost
Trade
---Lowest---
EquipmentClass
Priority
ReportMonth
```

In the PC 1, the variables MaterialCost , TotalCost and GrossProductionLoss are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the Trade, TradeGroup and SupervisorRole variables.

```
1
---Highest---
MaterialCost
TotalCost
GrossProductionLoss
---Lowest---
Trade
TradeGroup
SupervisorRole
```

In the PC 27, the variables IsAffectingProduction, ScheduleCompliant and EquipmentCriticality are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the Priority, ActualDuration and MaterialCost variables.

```
27
---Highest---
IsAffectingProduction
ScheduleCompliant
EquipmentCriticality
---Lowest---
Priority
ActualDuration
MaterialCost
```

This makes sense to us, because the PCA results tell us that the Total Cost variable is affected by the cost of the material required for the work order, the gross loss of production it caused, the compliance with the originally agreed dates and the production it affected.

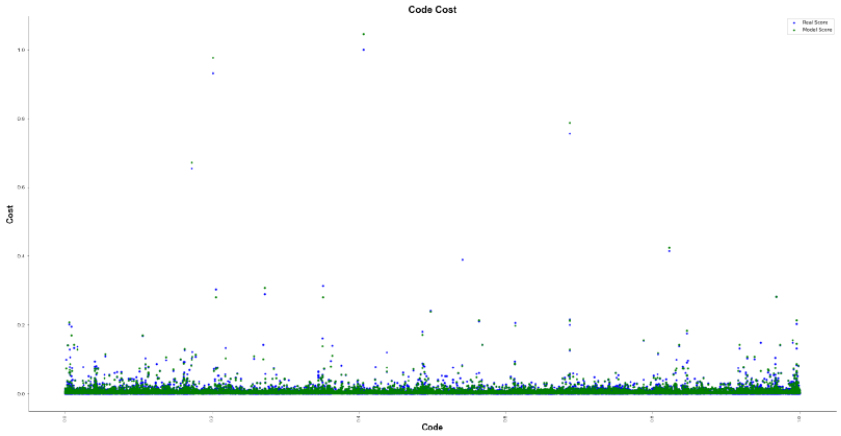
This could give us an idea that failures that cause losses in production, that take longer than expected to solve and whose materials are more expensive, will affect the total cost of this system's equipment.

Once we identified this, we created a linear regression model to predict the Total Cost. Our target is the Total Cost variable and we used the rest of the variables as independent variables. We used 80% of the data for training the model and 20% of the data to test it.

```
[ ]: # 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])
```

We got a R^2 score of 0.9432 which means that the model can explain 94.32% of the variability of the Total Cost variable. Also, we got a Mean Squared Error of 1.92e-05, which means that the results of the model are very close to reality.

We got this, where the blue dots are the real cost and the green dots are the predicted cost.

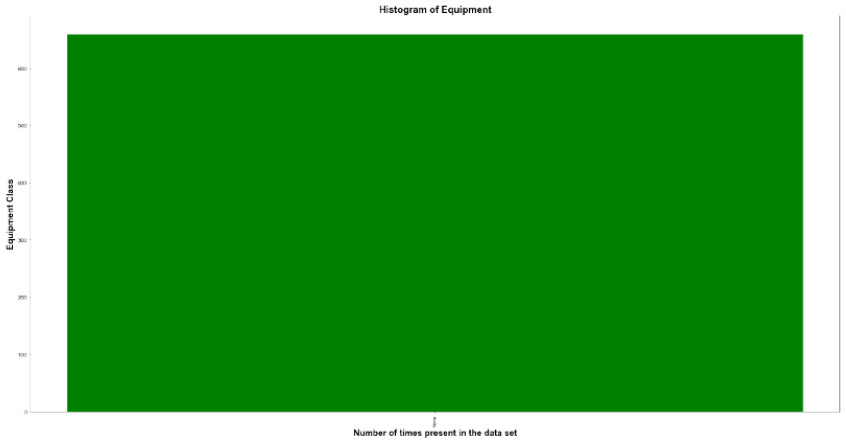


Equipment Type = Location

Basically, for the Location equipment type we did the same procedure as for the System equipment type.

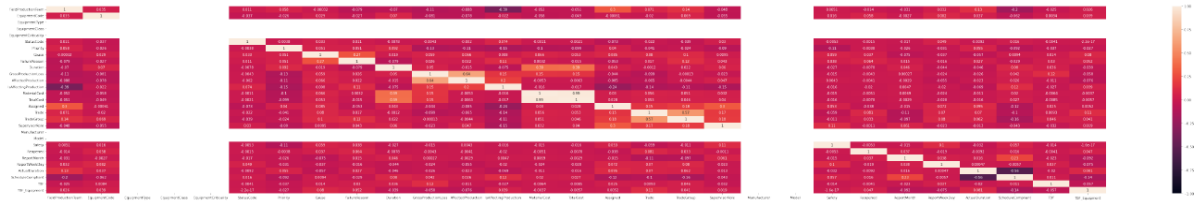
We created a histogram to see how many times the equipment class was repeating in our data. Surprisingly, the only class that was repeated more than 200 times was the "None" class.

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

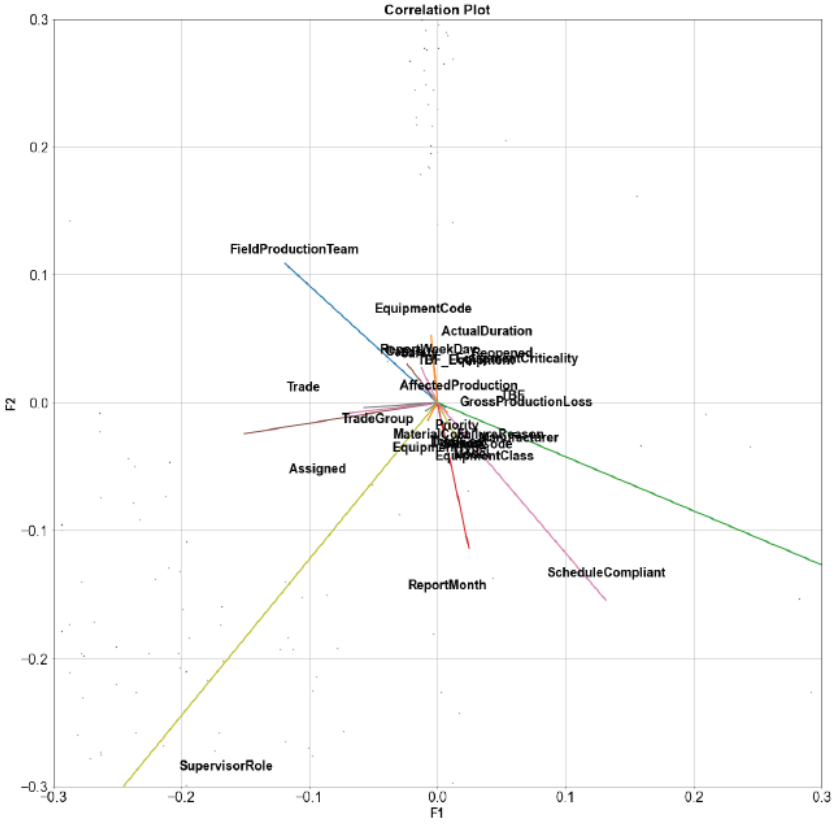


Still, the data was encoded and the blanks were filled with the mean of the column data to continue with the analysis of this type of equipment.

We created a correlation matrix:



And carried out the Principal Component Analysis.



We created a dataframe with the Principal Components factors for each variable in our data.

```
[ ]: # Se combierte 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
```

We used the getMaxComponent function again to get the maximum components affecting each variable.

```
[ ]: 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)
```

Then we looked for the 3 maximum and the 3 minimum PC values for each of the TBF_Equipment, Actual_Duration, TotalCost and IsAffectingProduction variables.

```
[ ]: 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])
```

Looking at our variable of interest (TotalCost), we found that the PC 1 and PC 23 we obtained from the Principal Component Analysis are the ones that are affecting the TotalCost variable the most.

```
TotalCost
[1, 23]
```

In the PC 1, the variables ScheduleCompliant, GrossProductionLoss and IsAffectingProduction are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the FieldProductionTeam, Assigned and ActualDuration variables.

```
1
---Highest---
ScheduleCompliant
GrossProductionLoss
IsAffectingProduction
---Lowest---
FieldProductionTeam
Assigned
ActualDuration
```

In the PC 23, the variables MaterialCost, TotalCost and Duration are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the IsAffectingProduction, ScheduleCompliant and ReportMonth variables.

```
23
---Highest---

MaterialCost
TotalCost
Duration
---Lowest---
IsAffectingProduction
ScheduleCompliant
ReportMonth
```

These results that we obtained are telling us that for Location type equipment, the total cost is affected by schedule problems, by loss of production, the cost of the materials needed for the order and, in some way, the duration. corrective work.

Once we identified this, we created a linear regression model to predict the Total Cost.

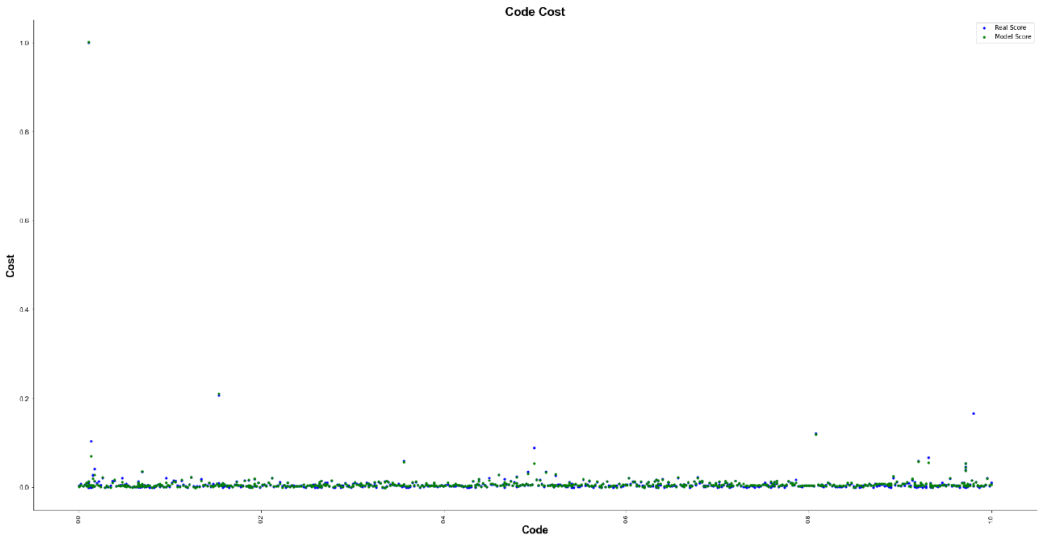
[ ]:
```
# importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
# fitting the training data
LR.fit(x_train,y_train)
```

[ ]:
```
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
```

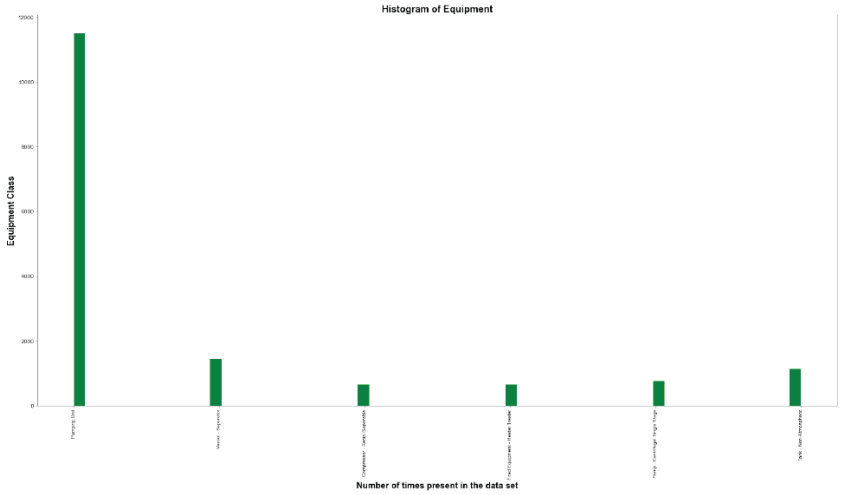
We got a R^2 score of 0.8438 which means that the model can explain 84.38% of the variability of the Total Cost variable. Also, we got a Mean Squared Error of 1.62e-05, which means that the results of the model are very close to the real ones.

We got this, where the blue dots are the real cost and the green dots are the predicted cost.

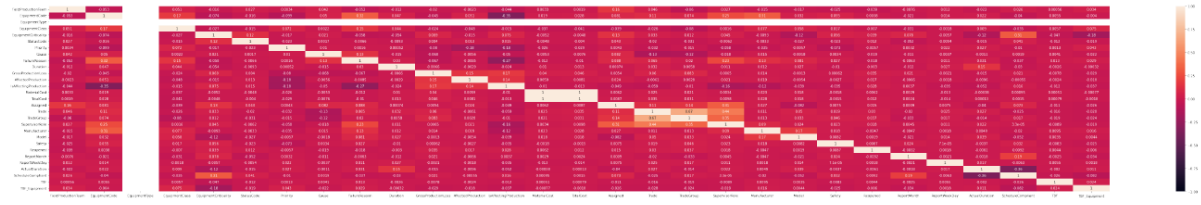


Equipment Type = Asset

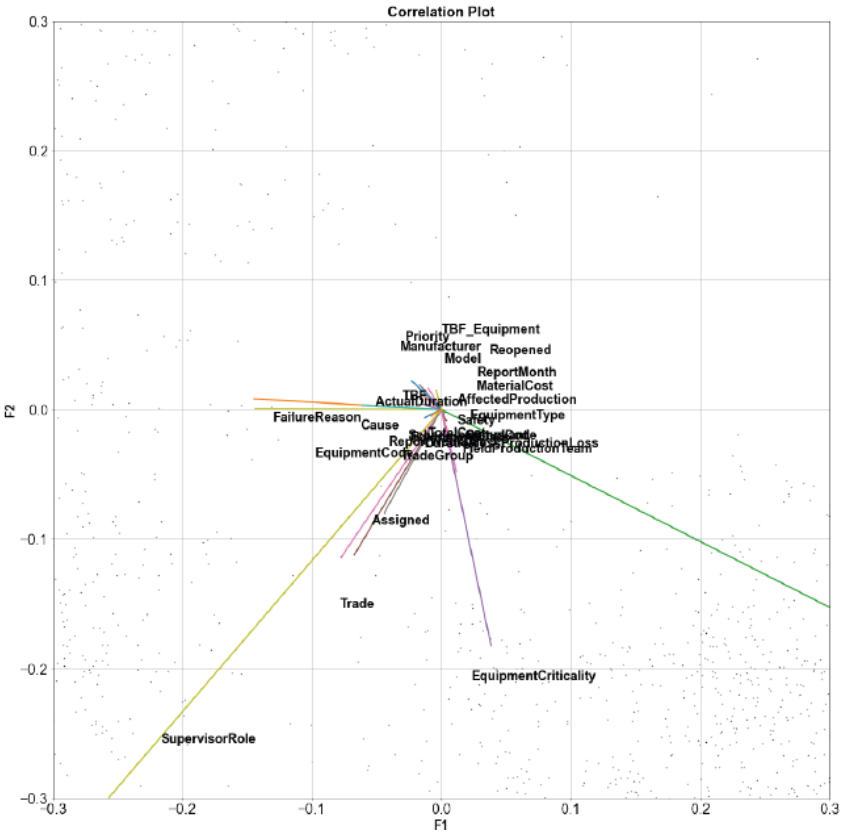
We created a histogram to see how many times the equipment class was repeating in our data. We got 6 classes that were repeated more than 500 times. These are the classes we will use for the following analysis since they're the ones with more information.



We created the correlation matrix:



And carried out the Principal Component Analysis.



We created a dataframe with the Principal Components factors for each variable in our data.

```
[ ]: # Se combierte 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
```

We used the getMaxComponent function to get the maximum components affecting each variable.

```
[ ]: 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]}
```

Then we looked for the 3 maximum and the 3 minimum PC values for each of the variables.

Looking at our variable of interest (TotalCost), we found that the PC 1 and PC 27 we obtained from the Principal Component Analysis are the ones that are affecting the TotalCost variable the most.

```
TotalCost
[1, 27]
```

In the PC 1, the variables SupervisorRole, Trade and TradeGroup are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the IsAffectingProduction, ScheduleCompliant and Priority variables.

```
1
---Highest---
SupervisorRole
Trade
TradeGroup
---Lowest---
IsAffectingProduction
ScheduleCompliant
Priority
```

In the PC 27, the variables TotalCost, MaterialCost and IsAffectingProduction are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the FailureReason, EquipmentClass and EquipmentCode variables.

```
27
---Highest---
TotalCost
MaterialCost
IsAffectingProduction
---Lowest---
FailureReason
EquipmentClass
EquipmentCode
```

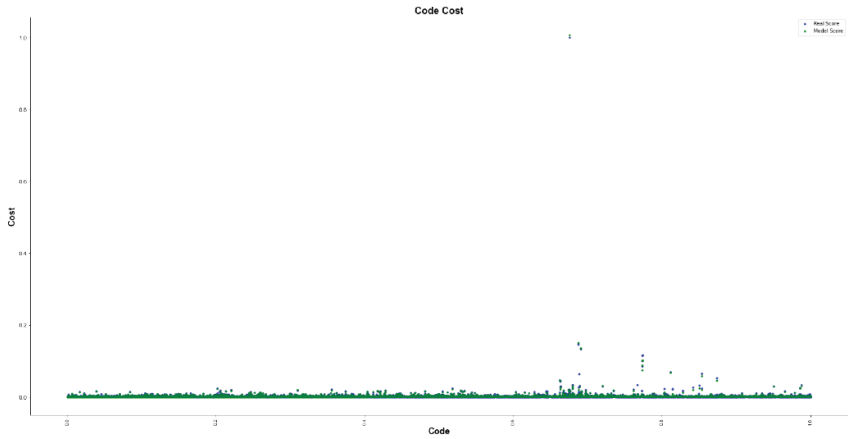
These data are telling us that for the type of Asset equipment, the total cost is affected by the situations of the trade and the trade group to which it belongs, as well as the situations that we have already seen with the previous types of equipment such as the cost of materials and the effect on production and, in this particular case, the cost could be affected depending on the class and code of the equipment.

Then we created a linear regression model to predict the Total Cost.

```
[ ]: # 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])
```

We got a very high R^2 score (0.9755) which means that the model can explain 97.55% of the variability of the Total Cost variable. We got a Mean Squared Error of 3.7293e-07, which means that the predicted values are very close to the real ones.

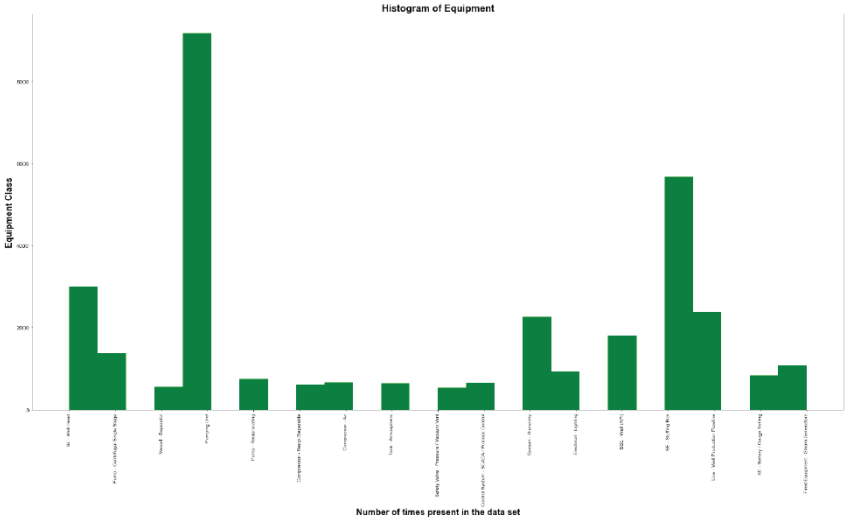
At the end we got this, where the blue dots are the real cost and the green dots are the predicted cost.



We can see and say that the predicted values are very accurate.

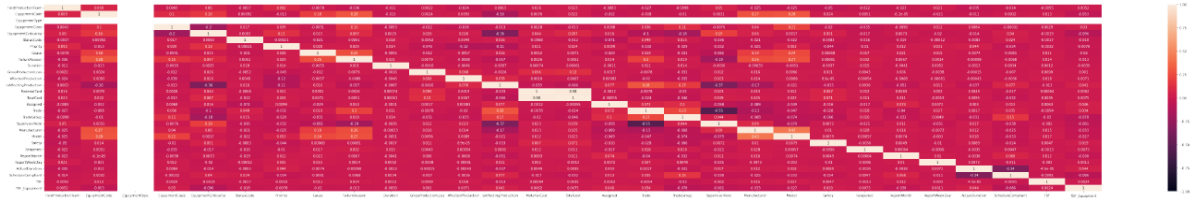
Equipment Type = Position

We plotted the histogram to see how many times the equipment class was repeating in our data. We got 17 classes that were repeated more than 400 times and these are the ones we are going to use for the analysis.

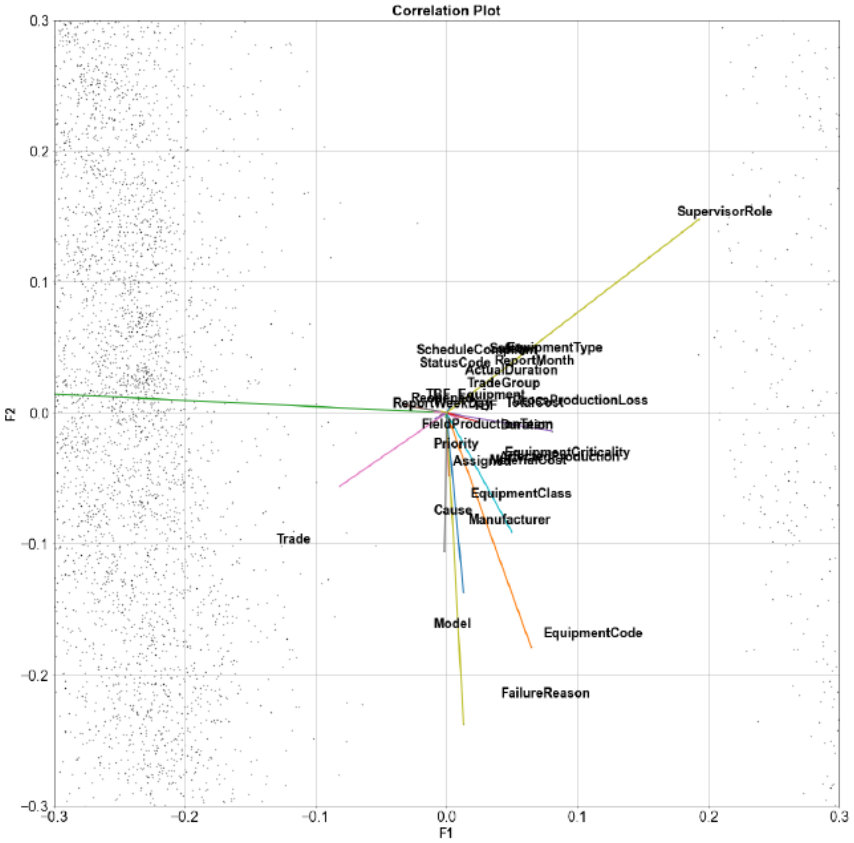


The data was encoded and the blanks were filled with the mean of the column data.

We created a correlation matrix:



And carried out the Principal Component Analysis.



We created a dataframe with the Principal Components factors for each variable in our data.

```
[ ]: # Se combierte 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
```

We used the getMaxComponent function to get the maximum components affecting each variable and then looked for the 3 maximum and the 3 minimum PC values for each of the TBF_Equipment, Actual_Duration, TotalCost and IsAffectingProduction variables in the same way we did for the previous equipment types.

Looking at the TotalCost variable, we found that the PC 2 and PC 27 we obtained from the Principal Component Analysis are the ones that are affecting the TotalCost variable the most.

```
TotalCost
[2, 27]
```

In the PC 2, the variables Manufacturer, EquipmentCode and EquipmentCriticality are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the IsAffectingProduction, Trade and TradeGroup variables.

```
2
---Highest---
Manufacturer
EquipmentCode
EquipmentCriticality
---Lowest---
IsAffectingProduction
Trade
TradeGroup
```

In the PC 27, the variables SupervisorRole, TotalCost and MaterialCost are the ones with the highest coefficients. On the other hand, the variables with the lowest coefficients are the FailureReason, Model and Cause variables.

```
27
---Highest---
SupervisorRole
TotalCost
MaterialCost
---Lowest---
FailureReason
Model
Cause
```

This means that for the types of Position equipment, the total cost variable is mainly affected by the type of equipment it is, depending on its code, its model, the criticality of the equipment and its manufacturer, as well as of the cost of the material and, for some reason, the role of the supervisor also affects the cost.

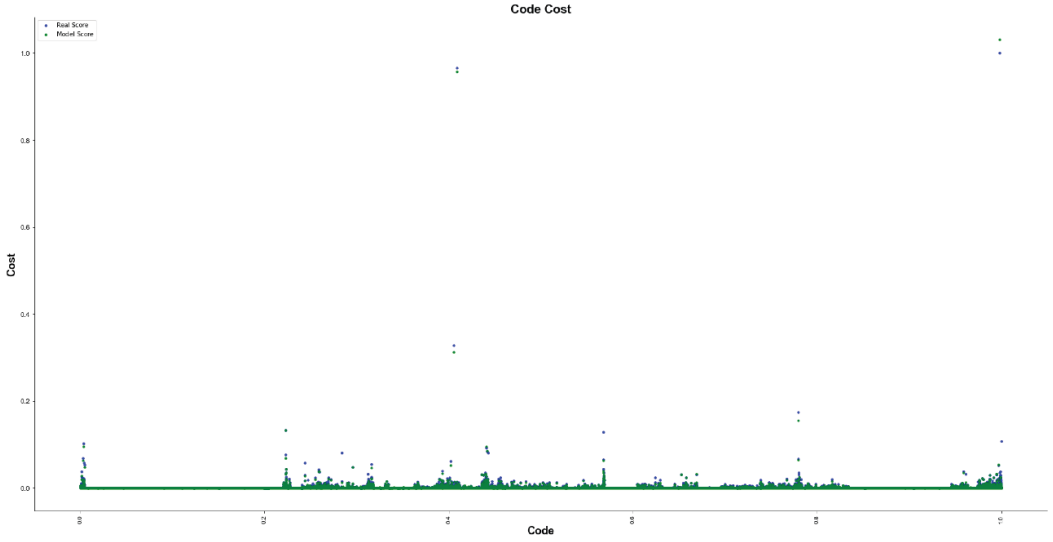
Here we can see that the total cost for this type of equipment depends a lot on the equipment and its characteristics itself, something that we had not seen in the previous types of equipment.

We created the linear regression model to predict the Total Cost.

```
[ ]: # 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])
```

We got a R^2 score of 0.7031 which means that the model can explain 70.31% of the variability of the Total Cost variable. We got a Mean Squared Error of 2.97e-06, which means that the predicted results and the real values are very close to each other.

We got this, where the blue dots are the real cost and the green dots are the predicted cost.



Conclusions

With the completion of this project, we were able to identify certain patterns or similar behaviors that affect the total corrective cost per piece of equipment. These analyzes were carried out by the 4 types of equipment that were provided to us in the data.

We can see that for the System and Location equipment types, the cost is affected by variables mainly related to the cost of the necessary materials, the loss of production generated by the failure and its impact and, in some way, for both positive and negative compliance with the schedule.
For equipment of the Asset type, we can see that, although the total cost depends on the cost of materials, it also affects the trade and commercial group to which it belongs, as well as the class and code of the equipment affect the final cost.
Finally, for the Position type equipment, we can see that the cost is mainly affected by variables of the equipment itself. Aspects such as the equipment code depends a lot on how expensive the work order will be, as well as how the model and its manufacturer play a leading role in terms of total cost. Here we can assume that for this type of equipment, it is where the cost depends more on the equipment than for the others.

For the 4 types, we created a regression model to predict the total cost variable, obtaining generally good scores and being very accurate. You can see the results of our predictions in the previous sections in this document.