/content/drive/MyDrive/Ficheros de Vueling/2022-09-20 14:39:04.960091 FLT_2022.xlsx

```
import numpy as np
import matplotlib.pyplot as plt

import pandas as pd
import seaborn as sns

%matplotlib inline


from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split, cross_val_score
```

```
# Configuració warnings
# ============================================================================
import warnings
warnings.filterwarnings('ignore')
```

```
# Activo Google Drive

from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

Obrir fitxer creat amb Calculo Vueling 2022 con codigos Rev D.ipynb

```
#@title Obrir fitxer creat amb Calculo Vueling 2022 con codigos Rev D.i
#path='/content/drive/MyDrive/2022.06.03 2022_delay + cierre puertas CO
Hoja = 'FLT_2022'
                                                    Drive/Ficheros de Vueling/2022-09-20 14:39:04
                                                    Drive/Ficheros de Vueling/2022-10-10 07:25:00
df = pd.read_excel(path) #, sheet_name=Hoja)

```

Creando una copia...

```
# Para crear una "logistic regresion":
# 10 minuts és el temps que s'accepta com arribar a l'hora

df['Tard1'] = np.where(df['Puntualidad1'] <10, 0, 1)
```

```
df['Tard2'] = np.where(df['Puntualidad2'] <10, 0, 1)
df['Tard3'] = np.where(df['Puntualidad3'] <10, 0, 1)
```

```
df.info()
```

```
 41  open cargo/pax door2            11094 non-null  object
 42  Secuencia                       12358 non-null  int64
 43  MES                             12358 non-null  int64
 44  Setmana                         12358 non-null  int64
 45  DiaSetmana                      12358 non-null  int64
 46  DiaSetmanaName                  12358 non-null  object
 47  Aeropuerto_Key                  12358 non-null  object
 48  AeropuertoKey1                  12358 non-null  object
 49  t_ClosePax                      12247 non-null  float64
 50  t_Close_Cargo_Door              12247 non-null  float64
 51  t_Entre_Puertas                 12247 non-null  float64
 52  Retardo_Abrir_Puerta_Pax        11093 non-null  float64
 53  Trayecto                        12358 non-null  object
 54  E_Despegue                      12358 non-null  int64
 55  lag_STD                         12358 non-null  datetime64[ns]
 56  lag_ATD                         12358 non-null  datetime64[ns]
 57  lag_STA                         12358 non-null  datetime64[ns]
 58  lag_ATA                         12358 non-null  datetime64[ns]
 59  lag_ACT PAX                     12358 non-null  int64
 60  lag_Secuencia                   12358 non-null  int64
 61  lag_REG                         12358 non-null  object
 62  T_teoricoTierra1                12358 non-null  int64
 63  T_RealTierra1                   12357 non-null  float64
 64  E_tierra1                       12357 non-null  float64
 65  Puntualidad1                    12357 non-null  float64
 66  Total_PAX_Boarding              12358 non-null  int64
 67  T_Medio_Boarding                12356 non-null  float64
 68  Taxi_Despegue                   12358 non-null  int64
 69  Taxi_Aterrizaje                 12357 non-null  float64
 70  DuracionVueloTeorico            12358 non-null  int64
 71  DuracionVueloReal               12357 non-null  float64
 72  E_Duracion_Vuelo                12357 non-null  float64
 73  E_Despegue2                     12358 non-null  int64
 74  E_Despegue3                     12357 non-null  float64
 75  E_Despegue4                     12358 non-null  int64
 76  E_Duracion_Vuelo2               12358 non-null  int64
 77  E_Duracion_Vuelo3               12357 non-null  float64
 78  E_Duracion_Vuelo4               12358 non-null  int64
 79  E_tierra2                       12357 non-null  float64
 80  [illegible]                     12358 non-null  int64
 81  [illegible]                     12358 non-null  int64
 82  [illegible]                     12358 non-null  object
 83  Aeropuerto_Key3                 12358 non-null  object
 84  Aeropuerto_Key4                 12358 non-null  object
 85  Puntualidad2                    12358 non-null  int64
 86  Puntualidad3                    12358 non-null  int64
 87  Puntualidad4                    12358 non-null  int64
 88  E_Despegue_Total                12357 non-null  float64
 89  E_Duracion_Vuelo_Total          12356 non-null  float64
 90  E_tierra_Total                  12356 non-null  float64
 91  E_acumulado_Total               12356 non-null  float64
 92  Trajecto                        12343 non-null  object
```

```
 93  retardoUperturaPuertaAterrizaje  11093 non-null  float64
 94  Tard1                            12358 non-null  int64
 95  Tard2                            12358 non-null  int64
 96  Tard3                            12358 non-null  int64
dtypes: datetime64[ns](18), float64(19), int64(26), object(34)
memory usage: 9.1+ MB
```

```
df[['Puntualidad1','Tard1']][:3]
```

	Puntualidad1	Tard1
0	15.0	1
1	-22.0	0
2	3.0	0

Només treballaré en els factors importants que crec afecten l'Error en puntualitat del 3r salt.

---

crec el DataFrame df1

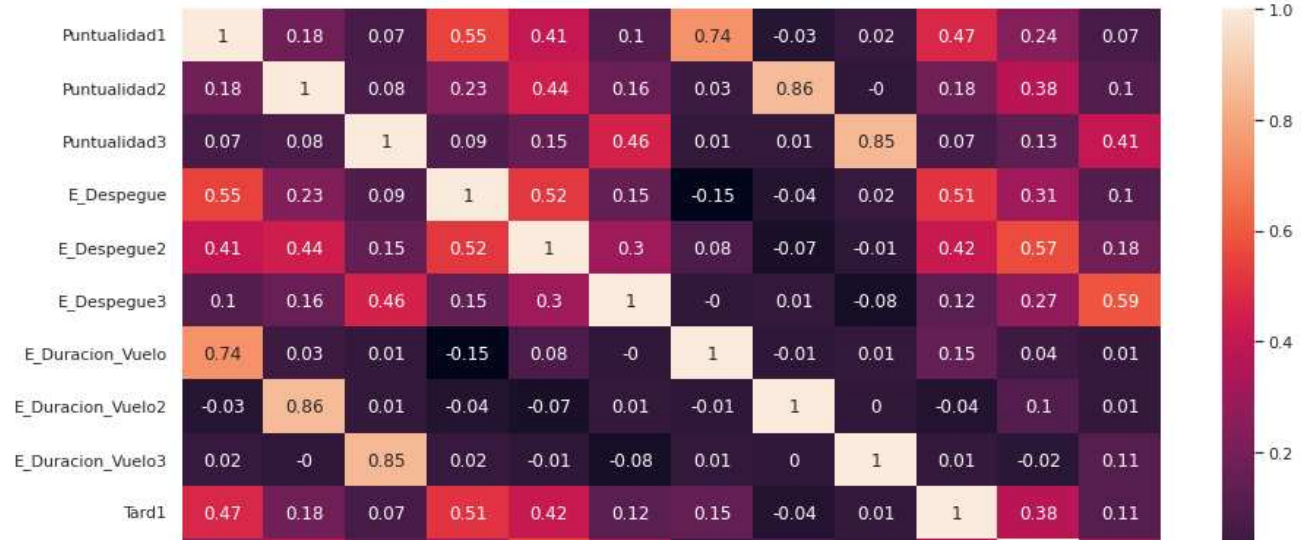
```
#@title crec el DataFrame df1
df1=df[['Puntualidad1', 'Puntualidad2', 'Puntualidad3',
        'E_Despegue','E_Despegue2','E_Despegue3',
        'E_Duracion_Vuelo','E_Duracion_Vuelo2','E_Duracion_Vuelo3', 'Ta
```

Crec la matriu de correlació entre tots els factors importants.

```
#@title Crec la matriu de correlació entre tots  els factors importants

correlation_matrix = df1.corr().round(2)
sns.set(rc = {'figure.figsize':(15,8)})
sns.heatmap(data=correlation_matrix, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff4cdf46650>
```




```
#@#@title Estandarditzar les característiques eliminant la mitjana i es

from sklearn_pandas import DataFrameMapper
from sklearn.preprocessing import StandardScaler

mapper = DataFrameMapper([(df1.columns, StandardScaler())])
scaled_features = mapper.fit_transform(df1.copy(), 4)
df1_StdScaler= pd.DataFrame(scaled_features, index=df1.index, columns=d

df1_StdScaler.head()
```

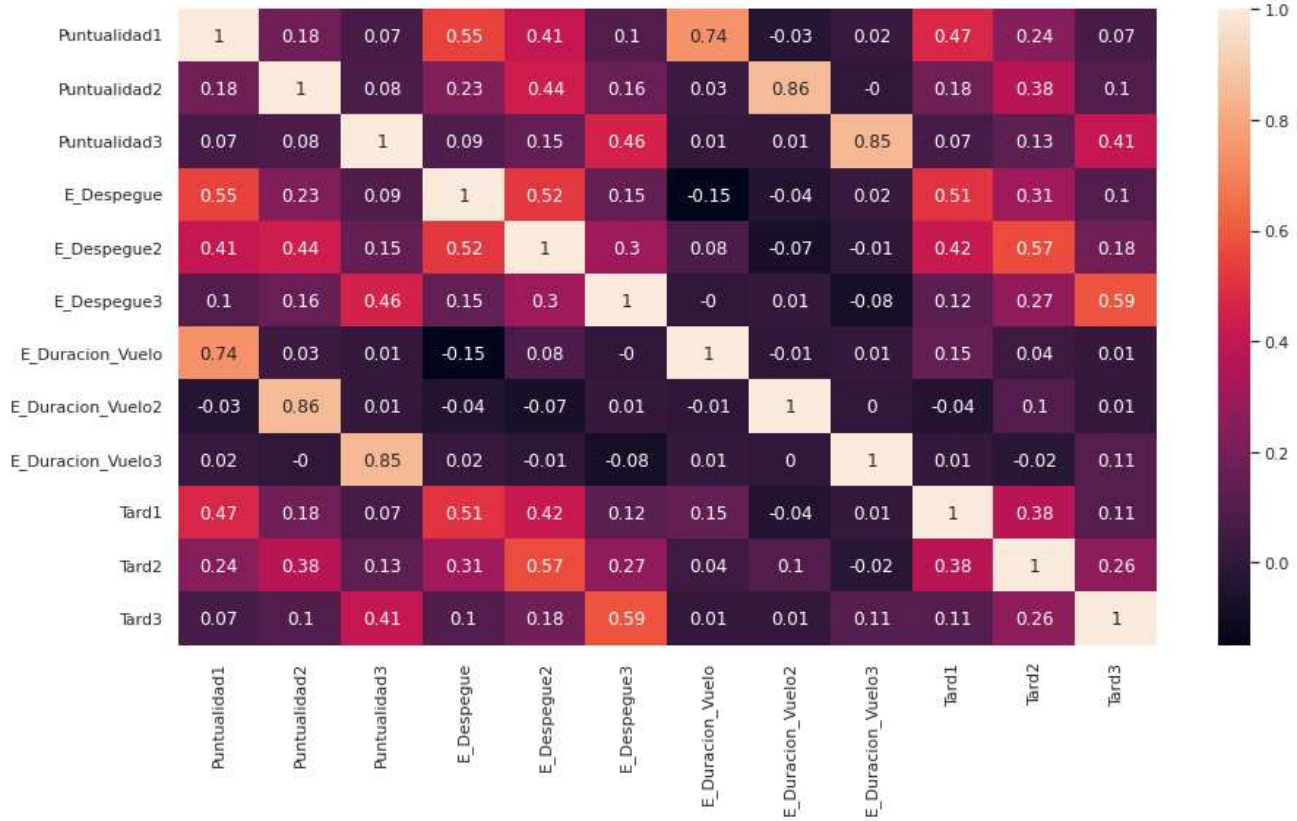
	Puntualidad1	Puntualidad2	Puntualidad3	E_Despegue	E_Despegue2	E_Despegue3	E
0	0.661875	0.158984	-0.020942	1.006091	0.398530	-0.039738	
1	-0.620447	-0.405877	-0.020942	-0.939568	-0.840450	-0.655431	
2	0.245987	0.204173	-0.169253	0.340471	0.133034	-0.095710	
3	0.800504	0.000823	-0.525200	1.159696	0.575527	-0.543487	
4	0.176672	-0.134744	-0.050604	-0.222746	-0.530705	-0.263626	

Creando una copia...

ació entre tots els factors importants despr

```
correlation_matrix = df1_StdScaler.corr().round(2)
sns.set(rc = {'figure.figsize':(15,8)})
sns.heatmap(data=correlation_matrix, annot=True)
```

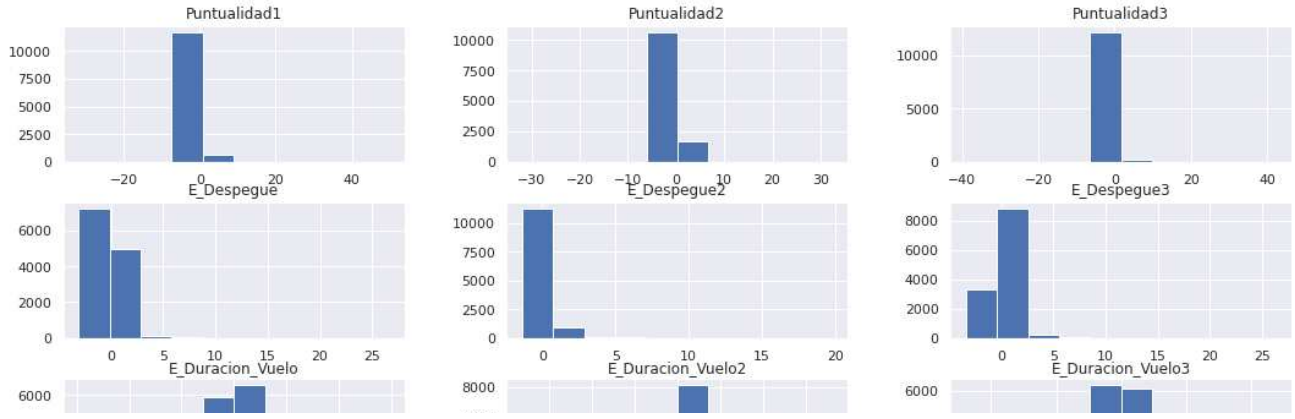
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff4ce3d2b90>
```



No hi veig diferències en les correlacions després de la normalització dels valors

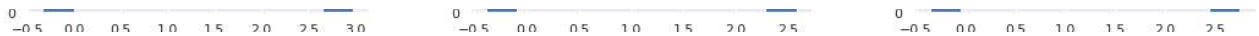
```
# Con los datos transformados veo si ha cambiado mucho la forma de cada
df1_StdScaler[df1_StdScaler.columns].hist(figsize=(18,10))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce011510>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4cdf8d110>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce660c50>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce21de10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce662790>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce0a1cd0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff4cdf59210>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce3a9410>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce3a95d0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce443490>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce42d9d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff4ce9c5650>]],
      dtype=object)
```



Veig que els outliers (> o < 30 minuts) afecten molt a la gràfica.

Trec els outliers



```
1 df1.head()
```

	Puntualidad1	Puntualidad2	Puntualidad3	E_Despegue	E_Despegue2	E_Despegue3	E
0	15.0	4	-5	20	11	0.0	
1	-22.0	-21	-5	-18	-17	-11.0	
2	3.0	6	-10	7	5	-1.0	
3	19.0	-3	-22	23	15	-9.0	
4	1.0	-9	-6	-4	-10	-4.0	

Creando una copia...

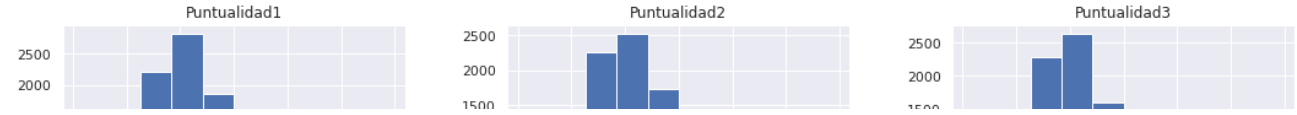
Elimino els outliers

Mostrar código

```
1 df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10708 entries, 0 to 12357
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Puntualidad1       10708 non-null  float64
 1   Puntualidad2       10708 non-null  int64  
 2   Puntualidad3       10708 non-null  int64  
 3   E_Despegue         10708 non-null  int64  
 4   E_Despegue2        10708 non-null  int64  
 5   E_Despegue3        10708 non-null  float64
 6   E_Duracion_Vuelo   10708 non-null  float64
 7   E_Duracion_Vuelo2  10708 non-null  int64  
 8   E_Duracion_Vuelo3  10708 non-null  float64
 9   Tard1              10708 non-null  int64  
 10  Tard2              10708 non-null  int64  
 11  Tard3              10708 non-null  int64  
dtypes: float64(4), int64(8)
memory usage: 1.1 MB
```

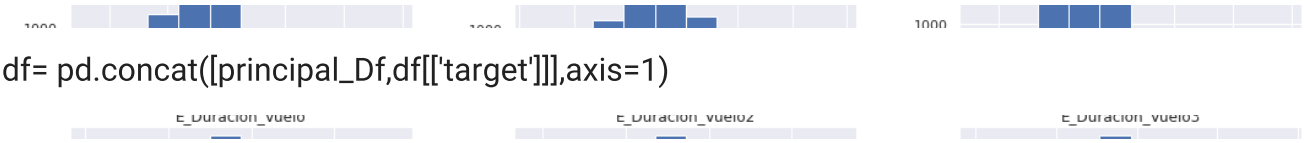
```
1 # Amb les dades transformades veig si ha canviat molt la forma de cada 
2
3 ax= df2[df2.columns[:-3]].hist(figsize=(18,10))
4
```



Jose Manuel Castaño.

desbanceado = sampling Diferencia de accuracy y precision.

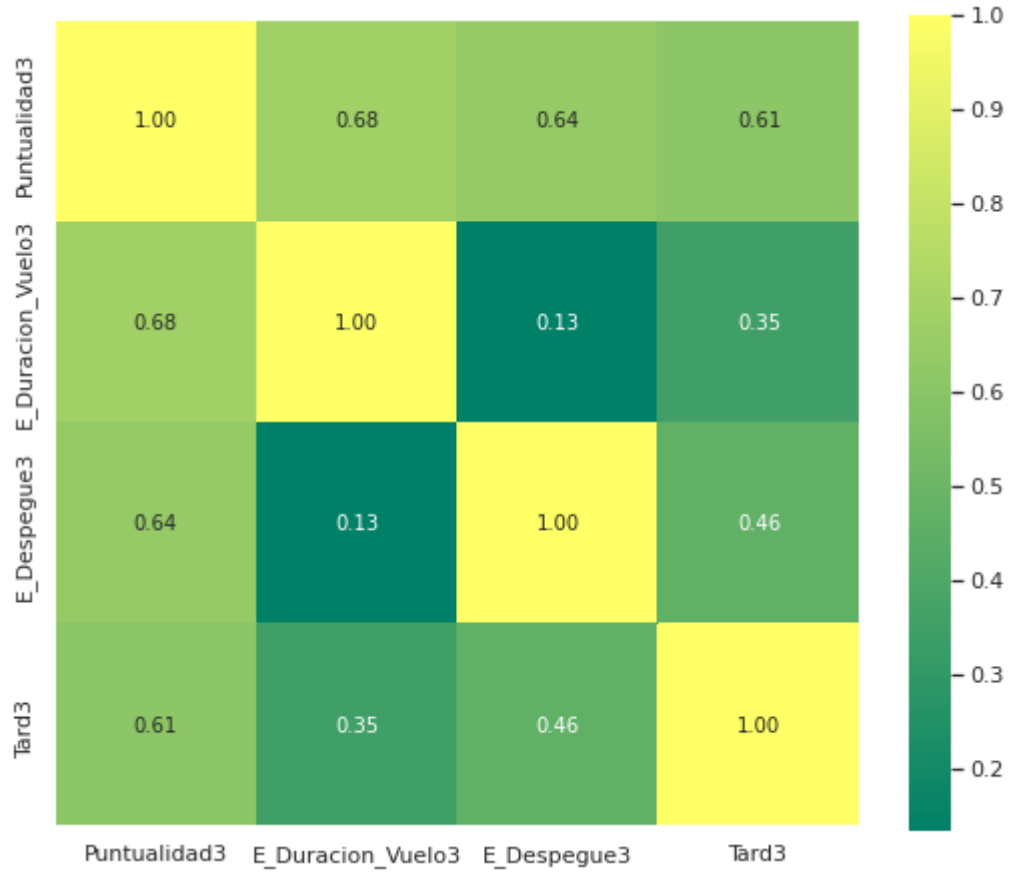
RobustScaler dummies gradien boosting clasificaier aplicar el cross validation random forest e sbueno para los desbalanceados



finaldf= pd.concat([principal_Df,df[['target']]],axis=1)

- quines variables tenen més correlació respecte al target (tard 3)

Mostrar código



Creando una copia...

as:

```
2
3 #Importo scikit-learn metrics module para el calculo
4 from sklearn import metrics
5 from sklearn.metrics import mean_squared_error
6 from sklearn.metrics import r2_score
7 metricasEjercicio2=[]
8 ##########################################################################
9
```

```
def MetricasModelo(metodo, Y_real, Y_calculado, scores):
  print('Metodo: ', metodo)

  #Precisión del modelo: ¿con qué frecuencia es correcto el clasificado
  #accuracy = metrics.accuracy_score(Y_real, Y_calculado)
  #print("Accuracy:", accuracy)

  rmse = np.sqrt(mean_squared_error(Y_real, Y_calculado,))
  print("RMSE: %f" % (rmse))


  R_squared = r2_score(Y_real, Y_calculado,)

  print("R-Squared: ", np.round(R_squared, 2))

  print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() *
  accuracy = "Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std

  # Guardo metricas
  metricasEjercicio2.append([metodo, rmse, R_squared, accuracy])


  cnf_matrix_SVM = metrics.confusion_matrix(Y_real, Y_calculado)

  print('\n\nMatriz de Confusión')
  print(cnf_matrix_SVM)


  # Creamos la Matriz de Confusion:

  class_names=[0,1] # name  of classes

  fig, ax = plt.subplots()
  tick_marks = np.arange(len(class_names))
  plt.xticks(tick_marks, class_names)
  plt.yticks(tick_marks, class_names)

  Creando una copia...

  sns.heatmap(pd.DataFrame(cnf_matrix_SVM), annot=True, cmap="gist_ncar
  ax.xaxis.set_label_position("top")


  plt.tight_layout()
  plt.title('Matriu de confusió', y=1.1)
  plt.ylabel('Valor Actual')
  plt.xlabel('Valor predit')
```

```
57   plt.Text(1.5,257.44,'Predicció')
58
```

Regresión logística con Python

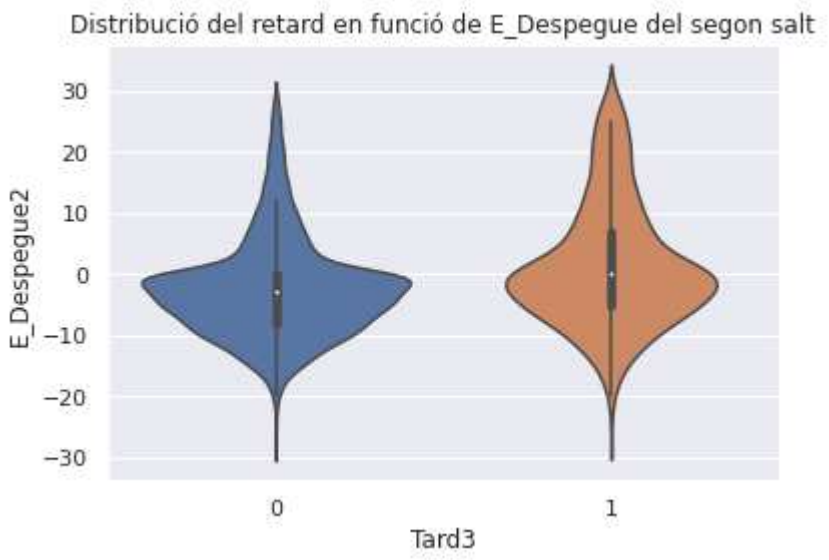
https://www.cienciadedatos.net/documentos/py17-regresion-logistica-python.html

```
1 # Número de obsercaciones por clase
2 # ==============================================================================
3 df2['Tard3'].value_counts().sort_index()
```

```
0    9950
1     758
Name: Tard3, dtype: int64
```

Gráfic distribució del retard-3 en funció de E_Despegue del segon salt

Mostrar código



T-test entre classes

Creando una copia...

```
2 # ==============================================================================
3 from statsmodels.stats.weightstats import ttest_ind
4
5
6 res_ttest = ttest_ind(
7                 x1 = df2[df2['Tard3'] == 0],
8                 x2 = df2[df2['Tard3'] == 1],
9                 alternative='two-sided'
```

```
10             )
11 #print(f"t={res_ttest[0]}, p-value={res_ttest[1]}")
12
13 print('t_test  =', round(res_ttest[1].mean(),3))
```

```
    t_test  = 0.001
```

Tant el gràfic com el t-test mostren evidències que hi ha una diferència entre l'error en el retard i l'error en l'enlairament. Aquesta informació és útil per considerar l'error en l'enlairament com un bon predictor per al model.

Considerem només una variable (Error en l'enlairament de l'anterior salt)

[ ] ↳ *12 celdas ocultas*

3.- Gestió unbalance dataset

https://elitedatascience.com/imbalanced-classes

Regressió Logística Balanceada

[ ] ↳ *11 celdas ocultas*

Regresión logística múltiple

```
1
2 X_train, X_test, y_train, y_test = train_test_split(
3                                         datos_x,
4                                        datos_y,
5                                         train_size   = 0.8,
6                                         random_state = 1234,
7                                         shuffle      = True
8                                     )
```

Creando una copia...

```
1 # Creació del model utilitzant matrius com a scikitlearn
2 # ============================================== ==================
3 # A la matriu de predictors cal afegir una columna de 1s per a l'interc
4
```

```
import statsmodels.api as sm

# Divisió de les dades en train i test
# ==============================================================================
datos_x= df2_sobremostrejat.loc[:, ['Puntualidad1', 'Puntualidad2',
        'E_Despegue','E_Despegue2','E_Despegue3',
        'E_Duracion_Vuelo','E_Duracion_Vuelo2','E_Duracion_Vuelo3',]].v

X_train, X_test, y_train, y_test = train_test_split(
                                        datos_x,
                                       datos_y,
                                        train_size   = 0.8,
                                        random_state = 1234,
                                        shuffle      = True
                                    )


```

```
#X_train = sm.add_constant(X_train, prepend=True)
modelo = sm.Logit(y_train, X_train,)
modelo = modelo.fit()
print(modelo.summary())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.146195
         Iterations: 35
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                14360
Model:                          Logit   Df Residuals:                    14354
Method:                           MLE   Df Model:                            5
Date:                Mon, 10 Oct 2022   Pseudo R-squ.:                  0.7872
Time:                        07:27:42   Log-Likelihood:                -2099.4
converged:                      False   LL-Null:                       -9865.6
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0661        nan        nan        nan         nan         nan
x2             0.0273        nan        nan        nan         nan         nan
x3             0.0230        nan        nan        nan         nan         nan
x4            -0.0491      0.005     -9.516      0.000      -0.059      -0.039
                        7     49.128      0.000       0.354       0.383
                      n        nan        nan         nan         nan
x7                    nan        nan        nan         nan         nan
x8             0.3760      0.008     46.538      0.000       0.360       0.392
==============================================================================
/usr/local/lib/python3.7/dist-packages/statsmodels/base/model.py:568: ConvergenceWarr
  ConvergenceWarning)
```

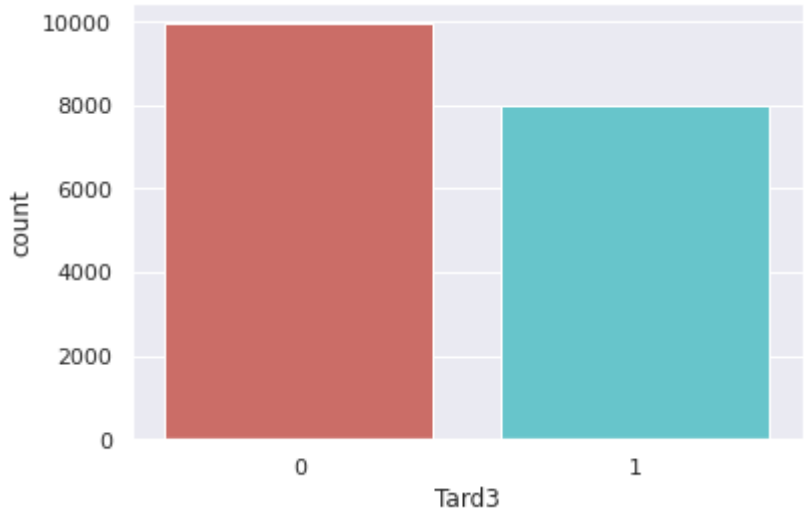
Creando una copia...

3.- Regresion Logistica Balanceada Binomial

```
1 df2_sobremostrejat.shape
```

```
(17950, 12)
```

```
1 fig, ax = plt.subplots(figsize=(6, 4))
2 sns.countplot(x='Tard3',data=df2_sobremostrejat, palette='hls')
3 plt.show()
```




```
1 datos_x= df2_sobremostrejat.loc[:, ['Puntualidad1', 'Puntualidad2',
2         'E_Despegue','E_Despegue2','E_Despegue3',
3         'E_Duracion_Vuelo','E_Duracion_Vuelo2','E_Duracion_Vuelo3',]]
4
5 datos_y = df2_sobremostrejat.loc[:,['Tard3']]
```

https://www.statsmodels.org/stable/index.html

Creem el model de regressió logística.(totes les variables) per veure quines variables afecten

```
1 #@title Creem el model de regressió logística.(totes les variables) per
2
```

Creando una copia...

```
5 logit_model=sm.Logit( datos_y, datos_x)
6 result=logit_model.fit()
7 print(result.summary2())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.147017
         Iterations: 35
                         Results: Logit
=================================================================
```

```
Model:              Logit            Pseudo R-squared: 0.786
Dependent Variable: Tard3            AIC:              5289.9023
Date:               2022-10-10 07:27 BIC:              5336.6744
No. Observations:   17950            Log-Likelihood:   -2639.0
Df Model:           5                LL-Null:          -12336.
Df Residuals:       17944            LLR p-value:      0.0000
Converged:          0.0000           Scale:            1.0000
No. Iterations:     35.0000
------------------------------------------------------------------
                    Coef.  Std.Err.    z    P>|z|  [0.025 0.975]
------------------------------------------------------------------
Puntualidad1        0.0653      nan     nan    nan    nan    nan
Puntualidad2        0.0276      nan     nan    nan    nan    nan
E_Despegue          0.0240      nan     nan    nan    nan    nan
E_Despegue2        -0.0475      nan     nan    nan    nan    nan
E_Despegue3         0.3673   0.0067 54.9488 0.0000 0.3542 0.3804
E_Duracion_Vuelo    0.0413      nan     nan    nan    nan    nan
E_Duracion_Vuelo2   0.0751      nan     nan    nan    nan    nan
E_Duracion_Vuelo3   0.3770   0.0072 52.1905 0.0000 0.3629 0.3912
==================================================================

/usr/local/lib/python3.7/dist-packages/statsmodels/base/model.py:568: ConvergenceWarr
  ConvergenceWarning)
```

Qué diferencia hay entre print(results.summary()) y print(results.summary2())

```
1 print(result.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Tard3   No. Observations:                17950
Model:                          Logit   Df Residuals:                    17944
Method:                           MLE   Df Model:                            5
Date:                Mon, 10 Oct 2022   Pseudo R-squ.:                  0.7861
Time:                        07:27:43   Log-Likelihood:                -2639.0
converged:                      False   LL-Null:                       -12336.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Puntualidad1          0.0653        nan        nan        nan        nan        nan
Puntualidad2          0.0276        nan        nan        nan        nan        nan
E_Despegue            0.0240        nan        nan        nan        nan        nan
E_Despegue2          -0.0475        nan        nan        nan        nan        nan
E_Despegue3           0.3673      0.007     54.949      0.000      0.354      0.380
E_Duracion_Vuelo      0.0413        nan        nan        nan        nan        nan
E_Duracion_Vuelo2     0.0751        nan        nan        nan        nan        nan
E_Duracion_Vuelo3     0.3770      0.007     52.190      0.000      0.363      0.391
==============================================================================
```

Creando una copia...

Veiem que només l'error en l'enlairament de l'avió i l'Error en el temps de vol, afecten com factors a la variable de sortida Error en l'arribada.

---

model amb 'E_Despegue3', 'E_Duracion_Vuelo3' només.

```
#@title model amb 'E_Despegue3', 'E_Duracion_Vuelo3' només.

cols=['E_Despegue3', 'E_Duracion_Vuelo3']
#cols=['E_Despegue3'] # Per crear una mica error
X=datos_x[cols]
y=datos_y


logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.194775
         Iterations 8
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: 0.717
Dependent Variable: Tard3            AIC:              6996.4355
Date:               2022-10-10 07:27 BIC:              7012.0262
No. Observations:   17950            Log-Likelihood:   -3496.2
Df Model:           1                LL-Null:          -12336.
Df Residuals:       17948            LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     8.0000
-----------------------------------------------------------------
                    Coef.  Std.Err.    z    P>|z|  [0.025 0.975]
-----------------------------------------------------------------
E_Despegue3         0.2857   0.0047 61.0477 0.0000 0.2765 0.2948
E_Duracion_Vuelo3   0.3126   0.0056 56.2014 0.0000 0.3017 0.3235
=================================================================
```

Aquest model està explicat amb un 71%

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3


                                   in,X_train)
print(result.summary2())
```

Creando una copia...

```
Optimization terminated successfully.
         Current function value: 0.193130
         Iterations 8
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: 0.719
Dependent Variable: Tard3            AIC:              4857.3689
```

```
Date:                2022-10-10 07:27 BIC:              4872.2463
No. Observations:    12565            Log-Likelihood:   -2426.7
Df Model:            1                LL-Null:          -8626.8
Df Residuals:        12563            LLR p-value:      0.0000
Converged:           1.0000           Scale:            1.0000
No. Iterations:      8.0000
----------------------------------------------------------------
                     Coef.  Std.Err.    z    P>|z|  [0.025 0.975]
----------------------------------------------------------------
E_Despegue3          0.2866   0.0056 51.3298 0.0000 0.2756 0.2975
E_Duracion_Vuelo3    0.3214   0.0069 46.7525 0.0000 0.3080 0.3349
================================================================
```

```
# Creación del modelo utilizando matrices como en scikitlearn
# ==============================================================================

# A la matriz de predictores se le tiene que añadir una columna de 1s pa

X_train = sm.add_constant(X_train, prepend=True)
modelo = sm.Logit(endog=y_train, exog=X_train,)
modelo = modelo.fit()
print(modelo.summary())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.000152
         Iterations: 35
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Tard3   No. Observations:                12565
Model:                          Logit   Df Residuals:                    12562
Method:                           MLE   Df Model:                            2
Date:                Mon, 10 Oct 2022   Pseudo R-squ.:                  0.9998
Time:                        07:27:43   Log-Likelihood:                -1.9054
converged:                      False   LL-Null:                       -8626.8
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              -101.8221     13.932     -7.309      0.000    -129.128     -74.516
E_Despegue3          10.8262      1.460      7.418      0.000       7.966      13.687
E_Duracion_Vuelo3    10.8301      1.461      7.415      0.000       7.967      13.693
=====================================================================================

Creando una copia...                          ion: A fraction 0.94 of observations can be
                                            indicate that there is complete
                                            some parameters will not be identified.
/usr/local/lib/python3.7/dist-packages/statsmodels/base/model.py:568: ConvergenceWarr
  ConvergenceWarning)
```

```
# Creación del modelo utilizando matrices como en scikitlearn
# ==============================================================================
# A la matriz de predictores se le tiene que añadir una columna de 1s pa
X_train = sm.add_constant(X_train, prepend=True)
```

```
modelo = sm.Logit(endog=y_train, exog=X_train,)
modelo = modelo.fit()
print(modelo.summary())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.000152
         Iterations: 35
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Tard3   No. Observations:                12565
Model:                          Logit   Df Residuals:                    12562
Method:                           MLE   Df Model:                            2
Date:                Mon, 10 Oct 2022   Pseudo R-squ.:                  0.9998
Time:                        07:27:43   Log-Likelihood:                -1.9054
converged:                      False   LL-Null:                       -8626.8
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const           -101.8221     13.932     -7.309      0.000    -129.128     -74.516
E_Despegue3       10.8262      1.460      7.418      0.000       7.966      13.687
E_Duracion_Vuelo3    10.8301      1.461      7.415      0.000       7.967      13.693
==============================================================================

Possibly complete quasi-separation: A fraction 0.94 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
/usr/local/lib/python3.7/dist-packages/statsmodels/base/model.py:568: ConvergenceWarn
  ConvergenceWarning)
```

```
# Predicciones con intervalo de confianza
# ==============================================================================
predicciones = modelo.predict(exog = X_train)

# Clasificación predicha
# ==============================================================================
clasificacion = np.where(predicciones<0.5, 0, 1)
clasificacion
```

```
array([0, 0, 0, ..., 0, 1, 0])
```

```
# Accuracy de test del modelo
# ==============================================================================
X_test = sm.add_constant(X_test, prepend=True)
predicciones = modelo.predict(exog = X_test)
clasificacion = np.where(predicciones<0.5, 0, 1)
accuracy = accuracy_score(
            y_true    = y_test,
            y_pred    = clasificacion,
            normalize = True
           )
```

Creando una copia...

```
print("")
print(f"El accuracy de test es: {100*accuracy}%")
```

```
El accuracy de test es: 100.0%
```

Matriz de confusión de las predicciones de test

```
#@title Matriz de confusión de las predicciones de test
# ========================================================================
confusion_matrix = pd.crosstab(
    y_test.values.ravel(),
    clasificacion,
    rownames=['Real'],
    colnames=['Predicción']
)
confusion_matrix
```

Predicción	0	1
Real		
0	2948	0
1	0	2437

Busquem la correlacion entre variables 'E_Despegue3' i 'E_Duracion_Vuelo3', i el tamany que és en funció del temps de retard.

Regresión logística simple. 'E_Despegue3','E_Duracion_Vuelo3'

```
#@title Regresión logística simple. 'E_Despegue3','E_Duracion_Vuelo3'
datos_x= df2_sobremostrejat.loc[:, ['Puntualidad1', 'Puntualidad2',
        'E_Despegue','E_Despegue2','E_Despegue3',
        'E_Duracion_Vuelo','E_Duracion_Vuelo2','E_Duracion_Vuelo3',]]

                                    at.loc[:,['Tard3']]

cols=['E_Despegue3','E_Duracion_Vuelo3']
#cols=['E_Despegue3'] # Per crear una mica error
X=datos_x[cols]
y=datos_y


logit_model=sm.Logit(y,exog = X)
```

```
result=logit_model.fit()
print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.194775
         Iterations 8
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: 0.717
Dependent Variable: Tard3            AIC:              6996.4355
Date:               2022-10-10 07:27 BIC:              7012.0262
No. Observations:   17950            Log-Likelihood:   -3496.2
Df Model:           1                LL-Null:          -12336.
Df Residuals:       17948            LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     8.0000
-----------------------------------------------------------------
                    Coef.  Std.Err.    z    P>|z|  [0.025 0.975]
-----------------------------------------------------------------
E_Despegue3         0.2857   0.0047 61.0477 0.0000 0.2765 0.2948
E_Duracion_Vuelo3   0.3126   0.0056 56.2014 0.0000 0.3017 0.3235
=================================================================
```

```
y_pred = result.predict(X)
```

Predicció:

Un cop entrenat el model, es poden obtenir prediccions per a noves dades. Els models de regressió logística de statsmodels tornen la probabilitat de pertànyer a la classe de referència.

```
# Predicción de probabilidades
# ==============================================================================
predicciones = result.predict(exog = X)
predicciones[:4]
```

```
0    0.173249
1    0.219781
2    0.043163
3    0.001313
dtype: float64
```

Creando una copia...

```
# Clasificación predicha
# ==============================================================================
clasificacion = np.where(predicciones<0.5, 0, 1)
clasificacion
```

```
array([0, 0, 0, ..., 1, 1, 1])
```

Es veu clarament que les probabilitats d'arribar a l'hora és molt alta.

Dibuixem la regressió logística només per E_Despegue3 (error en l'enlairament)

```
 1
 2 #@title Dibuixem la regressió logística només per E_Despegue3 (error en
 3
 4 cols=['E_Despegue3','E_Duracion_Vuelo3']
 5 cols=['E_Despegue3'] # Per crear una mica error
 6 X=datos_x[cols]
 7 y=datos_y
 8 logit_model=sm.Logit(y,exog = X)
 9
10
11 grid_X = np.linspace(
12             start = min(X.E_Despegue3),
13             stop  = max(X.E_Despegue3),
14             num   =  predicciones.shape[0] #200
15          ).reshape(-1,1)
16
17
18 grid_X=pd.DataFrame(grid_X)
19 result=logit_model.fit()
20 predicciones = result.predict(exog = grid_X)
21
22 fig, ax = plt.subplots(figsize=(6, 3.84))
23 ax.set_title("Modelo regresión logística")
24 ax=sns.scatterplot(data = df2, x = "E_Despegue3", y = "Tard3", s=100)
25 ax=sns.scatterplot( x = grid_X[0], y = predicciones, s=20, color='black
26
27 result=logit_model.fit()
28 print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.476043
         Iterations 6
Optimization terminated successfully.
         Current function value: 0.476043
         Iterations 6
                         Results: Logit
================================================================
Model:              Logit            Pseudo R-squared: 0.307
Dependent Variable: Tard3            AIC:              17091.9550
Date:               2022-10-10 07:27 BIC:              17099.7503
No. Observations:   17950            Log-Likelihood:   -8545.0
Df Model:           0                LL-Null:          -12336.
Df Residuals:       17949            LLR p-value:      nan
Converged:          1.0000           Scale:            1.0000
No. Iterations:     6.0000
----------------------------------------------------------------
                 Coef.   Std.Err.     z     P>|z|   [0.025  0.975]
----------------------------------------------------------------
E_Despegue3      0.1775    0.0028  62.6566  0.0000  0.1719  0.1830
================================================================
```

Conclusió:

Clarament, veiem que és una regressió logística, que el seu valor de correlació és baix. En aquest cas és de 0,3, i per això explica molt poc amb una sola variable.

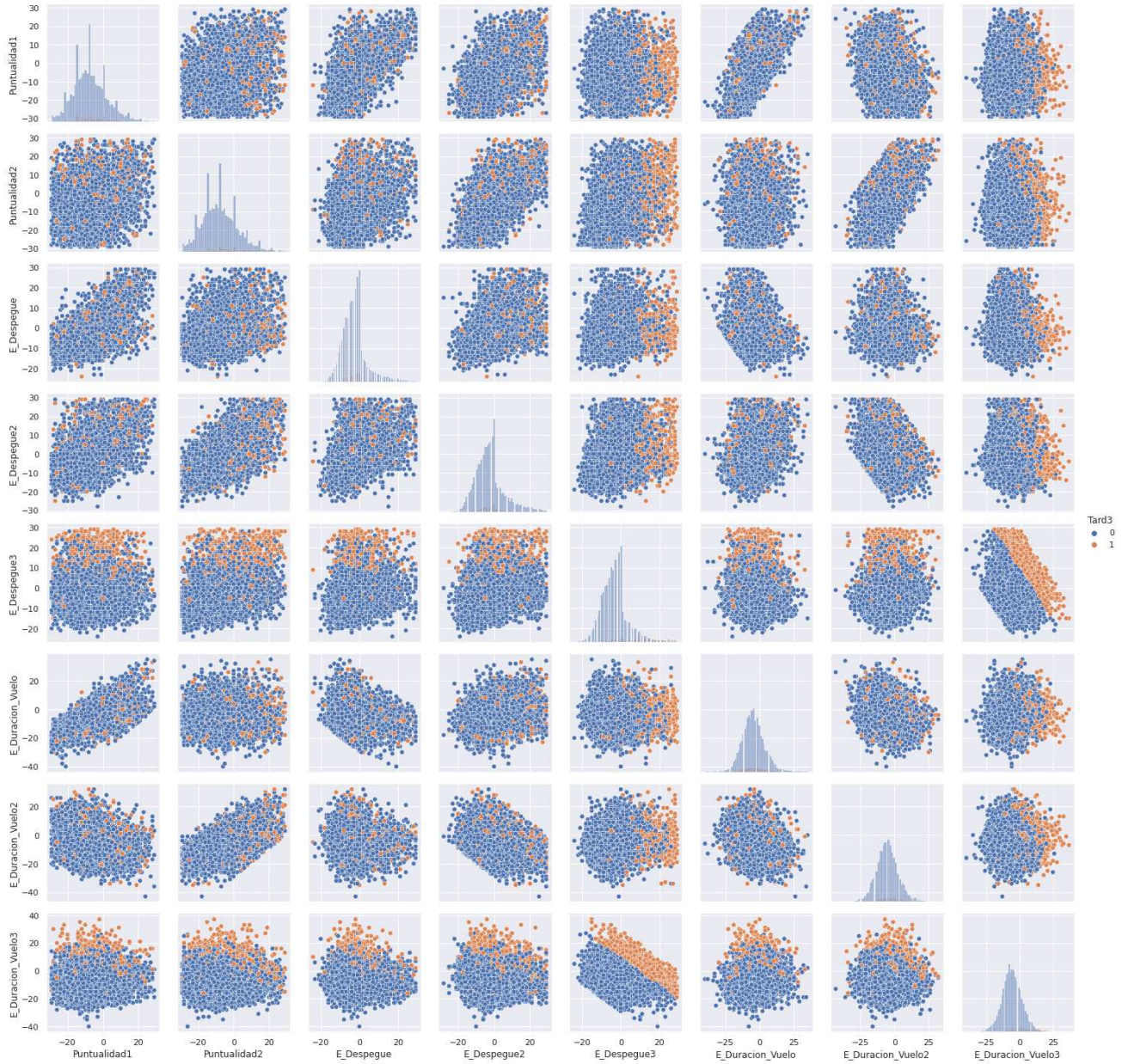
Per afirmar bé el model pel retard del 3r salt, hem de fer servir 2 variables (E_Duracion_Vuelo2, E_Despegue3) i arribaré a afirmar en un 0.72 què succeirà.

És molt poc i s'arriba a una conclusió lògica, si l'avió surt tard i volan triga més del planificat, arribarà tard. Però és molt important veure que el que ha passat en els 2 salts anteriors no afecta pràcticament res al salt 3r.

Independentment de si han surtit o volant, no han tingut una demora superior a 30 minuts que és el que considerem un vol "normal", en els 2 salts anteriors.

Important veure la correlació entre E_Duracion_Vuelo2, E_Despegue3 i com els vols que arriben tard estan en un extrem.

Correlació entre variables



Creando una copia...

Aquesta asseveració és el punt crític del projecte.

Hem demostrat que els salts són independents entre si. Un retard en un avió és degut a moltes causes, que són variables independents que afecten i que moltes són alienes a la mateixa companyia (exògenes)... però el primer salt és el que menys impacta té i que pot controlar-les,

perquè per exemple no hi ha retards amb altres interconnexions, o passatgers que han de buscar la maleta, canvis de porta que desorienten als passatgers, etc.

Llavors el que hem d'analitzar és el valor del primer salt, enlairament 1r i analitzar les causes de per què el 99% dels vols no surten a la seva hora. I sortir a l'hora és sortir a <0 minuts, no a <10 minuts.

PANDAS - Manipulacion de Datos con Python

https://www.google.com/search?q=dataframe+psar+de+numero+a+texto&rlz=1C1WPZA_esES1023ES1023&oq=dataframe+psar+de+numero+a+texto&aqs=chrome..69i57j33i10i160l2.15256j0j7&sourceid=chrome&ie=UTF-8