# ocorrencias_aeronauticas

August 1, 2021

## 1 Ocorrências Aeronáuticas

### 1.1 Informações Gerais

A base de dados de ocorrências aeronáuticas é gerenciada pelo Centro de Investigação e Prevenção de Acidentes Aeronáuticos (CENIPA). Constam nesta base de dados as ocorrências aeronáuticas notificadas ao CENIPA nos últimos 10 anos e que ocorreram em solo brasileiro.

URL: https://dados.gov.br/dataset/ocorrencias-aeronauticas-da-aviacao-civil-brasileira

BASES DE DADOS UTILIZADAS: * OCORRÊNCIA.csv - Informações sobre as ocorrências. * AERONAVE.csv - Informações sobre as aeronaves envolvidas nas ocorrências.

### 1.2 Identificação do problema a ser abordado

Cliente deseja saber se entre todas as ocorências notificadas (acidente, incidente leve, incidente grave), se é possivel prever a chance de um Acidente ocorrer.

### 1.3 Importação dos pacotes necessários

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import␣
 ↪classification_report,confusion_matrix,accuracy_score
from xgboost import XGBClassifier
from xgboost import plot_importance
from sklearn.ensemble import VotingClassifier

from google.colab import files
```

[2]:
```
#configurações para ajudar na visualização dos dados
np.set_printoptions(threshold=None, precision=2)
pd.set_option('display.max_columns', 500)
pd.set_option('display.max_rows', 500)
pd.set_option('precision', 2)
```

### 1.4 Leitura e análise da base principal

[3]:
```
#carregando a base principal

uploaded = files.upload()
df_ocorrencias = pd.read_csv('ocorrencia_2010_2020.csv', sep=';')
```

```
<IPython.core.display.HTML object>


Saving ocorrencia_2010_2020.csv to ocorrencia_2010_2020 (11).csv
```

[4]:
```
#analise inicial da base

print("\nDimensões:\n{0}\n".format(df_ocorrencias.shape))
print("\nCampos:\n{0}\n".format(list(df_ocorrencias.keys())))
print("\nTipos dos dados:\n{0}\n".format(df_ocorrencias.info()))
```

```
Dimensões:
(5752, 22)


Campos:
['codigo_ocorrencia', 'codigo_ocorrencia1', 'codigo_ocorrencia2',
'codigo_ocorrencia3', 'codigo_ocorrencia4', 'ocorrencia_classificacao',
'ocorrencia_latitude', 'ocorrencia_longitude', 'ocorrencia_cidade',
'ocorrencia_uf', 'ocorrencia_pais', 'ocorrencia_aerodromo', 'ocorrencia_dia',
'ocorrencia_hora', 'investigacao_aeronave_liberada', 'investigacao_status',
'divulgacao_relatorio_numero', 'divulgacao_relatorio_publicado',
'divulgacao_dia_publicacao', 'total_recomendacoes',
'total_aeronaves_envolvidas', 'ocorrencia_saida_pista']

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5752 entries, 0 to 5751
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   codigo_ocorrencia               5752 non-null   int64
 1   codigo_ocorrencia1              5752 non-null   int64
 2   codigo_ocorrencia2              5752 non-null   int64
 3   codigo_ocorrencia3              5752 non-null   int64
 4   codigo_ocorrencia4              5752 non-null   int64
 5   ocorrencia_classificacao        5752 non-null   object
 6   ocorrencia_latitude             4187 non-null   object
 7   ocorrencia_longitude            4187 non-null   object
 8   ocorrencia_cidade               5752 non-null   object
```

```
 9   ocorrencia_uf                   5752 non-null   object
 10  ocorrencia_pais                 5752 non-null   object
 11  ocorrencia_aerodromo            5752 non-null   object
 12  ocorrencia_dia                  5752 non-null   object
 13  ocorrencia_hora                 5751 non-null   object
 14  investigacao_aeronave_liberada  5411 non-null   object
 15  investigacao_status             5412 non-null   object
 16  divulgacao_relatorio_numero     4887 non-null   object
 17  divulgacao_relatorio_publicado  5752 non-null   object
 18  divulgacao_dia_publicacao       1494 non-null   object
 19  total_recomendacoes             5752 non-null   int64
 20  total_aeronaves_envolvidas      5752 non-null   int64
 21  ocorrencia_saida_pista          5752 non-null   object
dtypes: int64(7), object(15)
memory usage: 988.8+ KB

Tipos dos dados:
None
```

[5]:
```
#estatistica descritiva dos dados

print(df_ocorrencias.describe())
categ = df_ocorrencias.dtypes[df_ocorrencias.dtypes == "object"].index
print("\n", df_ocorrencias[categ].describe(), sep='\n')
```

```
       codigo_ocorrencia  codigo_ocorrencia1  codigo_ocorrencia2  \
count            5752.00             5752.00             5752.00
mean            58504.21            58504.21            58504.21
std             14060.29            14060.29            14060.29
min             39115.00            39115.00            39115.00
25%             46366.50            46366.50            46366.50
50%             52692.50            52692.50            52692.50
75%             77653.75            77653.75            77653.75
max             79874.00            79874.00            79874.00

       codigo_ocorrencia3  codigo_ocorrencia4  total_recomendacoes  \
count             5752.00             5752.00              5752.00
mean             58504.21            58504.21                 0.30
std              14060.29            14060.29                 1.15
min              39115.00            39115.00                 0.00
25%              46366.50            46366.50                 0.00
50%              52692.50            52692.50                 0.00
75%              77653.75            77653.75                 0.00
max              79874.00            79874.00                23.00

       total_aeronaves_envolvidas
```

```
count                    5752.00
mean                        1.01
std                         0.11
min                         1.00
25%                         1.00
50%                         1.00
75%                         1.00
max                         3.00


       ocorrencia_classificacao ocorrencia_latitude ocorrencia_longitude  \
count                      5752                4187                 4187
unique                        3                2495                 2490
top                   INCIDENTE                 ***                  ***
freq                       3171                 726                  727

       ocorrencia_cidade ocorrencia_uf ocorrencia_pais ocorrencia_aerodromo  \
count               5752          5752            5752                 5752
unique              1099            28               1                  515
top       RIO DE JANEIRO            SP          BRASIL                 ****
freq                 321          1373            5752                 2181

       ocorrencia_dia ocorrencia_hora investigacao_aeronave_liberada  \
count            5752            5751                          5411
unique           2971             967                             3
top        18/12/2013        20:00:00                           SIM
freq                9             123                          3221

       investigacao_status divulgacao_relatorio_numero  \
count                 5412                        4887
unique                   2                        1677
top             FINALIZADA                         ***
freq                  4911                        2928

       divulgacao_relatorio_publicado divulgacao_dia_publicacao  \
count                            5752                     1494
unique                              2                      215
top                               NÃO              2017-08-01
freq                             4265                       58

       ocorrencia_saida_pista
count                    5752
unique                      2
top                       NÃO
freq                     5245
```

### 1.5 Tratamento inicial da base

Como o cliente deseja saber algo específico sobre o relacionamento entre tipo de aeronave e classificação da ocorrência, utlizaremos apenas algumas variáveis categóricas. Portanto, embora seja uma base muito rica em informações, para essa análise específica podem ser desconsiderados muitos atributos

[6]:
```
#criando copia da base original utilizando apenas os atributos desejados

colunas = ['codigo_ocorrencia', 'ocorrencia_classificacao',␣
 ↪'ocorrencia_saida_pista']
df_ocorrencias = df_ocorrencias[colunas].copy()

#Garantindo que não tenham ocorrências duplicadas

df_ocorrencias.sort_values(by=['codigo_ocorrencia'],␣
 ↪ascending=[True],inplace=True)
df_ocorrencias.drop_duplicates(['codigo_ocorrencia'], keep='first',␣
 ↪inplace=True)
```

### 1.6 Enriquecimento da base trazendo informações sobre tipo e o modelo das aviações

[7]:
```
#carregando a base aeronave já com as colunas escolhidas para analise

uploaded = files.upload()
df_aeronave = pd.read_csv('aeronave_2010_2020.csv', sep=';')
df_aeronave = df_aeronave[['codigo_ocorrencia2','aeronave_tipo_veiculo',
                           'aeronave_motor_tipo']]
df_aeronave = df_aeronave.rename(columns={'codigo_ocorrencia2':␣
 ↪'codigo_ocorrencia'})
df_aeronave.sort_values(by=['codigo_ocorrencia'], ascending=[True],inplace=True)
df_aeronave.drop_duplicates(['codigo_ocorrencia'], keep='first', inplace=True)

# merge ocorrencias e aeronave
df_ocorrencias = pd.merge(
    df_ocorrencias,
    df_aeronave,
    how='left',
    on='codigo_ocorrencia'
)
```

```
<IPython.core.display.HTML object>


Saving aeronave_2010_2020.csv to aeronave_2010_2020 (11).csv
```

### 1.7 Preparando os dados para o modelo

[8]:
```
df_ocorrencias.head(5)
```

[8]:
```
   codigo_ocorrencia ocorrencia_classificacao ocorrencia_saida_pista  \
0              39115                ACIDENTE                    NÃO
1              39155               INCIDENTE                    NÃO
2              39156         INCIDENTE GRAVE                    NÃO
3              39158               INCIDENTE                    NÃO
4              39176               INCIDENTE                    NÃO

  aeronave_tipo_veiculo aeronave_motor_tipo
0                 AVIÃO             PISTÃO
1                 AVIÃO          TURBOÉLICE
2                 AVIÃO          TURBOÉLICE
3                 AVIÃO                JATO
4                 AVIÃO                JATO
```

[9]:
```
#tratando os valores que estavam faltando "***"" e passando para "NÃO␣
 ↪INFORMADO"
df_ocorrencias['aeronave_tipo_veiculo'] =␣
 ↪df_ocorrencias['aeronave_tipo_veiculo'].replace({'***':'NÃO INFORMADO'})
df_ocorrencias['aeronave_motor_tipo'] = df_ocorrencias['aeronave_motor_tipo'].
 ↪replace({'***':'NÃO INFORMADO'})

#Criando coluna acidente a partir da ocorrencia_classificacao com os valores␣
 ↪SIM ou NÃO para predição
conditions = [
        (df_ocorrencias['ocorrencia_classificacao'].str.contains('ACIDENTE')),
        (df_ocorrencias['ocorrencia_classificacao'].str.contains('INCIDENTE'))]

values = ['SIM','NÃO']

df_ocorrencias['acidente'] = np.select(conditions, values).copy()

#limpando  o df
del df_ocorrencias["codigo_ocorrencia"]
del df_ocorrencias["ocorrencia_classificacao"]
```

[10]:
```
df_ocorrencias
```

[10]:
```
     ocorrencia_saida_pista aeronave_tipo_veiculo aeronave_motor_tipo acidente
0                       NÃO                AVIÃO             PISTÃO      SIM
1                       NÃO                AVIÃO          TURBOÉLICE      NÃO
2                       NÃO                AVIÃO          TURBOÉLICE      NÃO
3                       NÃO                AVIÃO                JATO      NÃO
4                       NÃO                AVIÃO                JATO      NÃO
...                     ...                  ...                 ...      ...
5747                    NÃO                AVIÃO                JATO      NÃO
```

```
5748                    NÃO         NÃO INFORMADO                JATO      NÃO
5749                    NÃO                AVIÃO             PISTÃO      SIM
5750                    NÃO          HELICÓPTERO          TURBOEIXO      NÃO
5751                    NÃO         NÃO INFORMADO       NÃO INFORMADO      SIM

[5752 rows x 4 columns]
```

### 1.8 Preparação

Criando variáveis independentes e dependentes

[11]:
```
y = df_ocorrencias['acidente']
X = df_ocorrencias
X = df_ocorrencias.drop('acidente',axis = 1)
```

[12]:
```
#Tratando a variavel categorica alterando valor para binário 0 e 1
le = LabelEncoder()
X['ocorrencia_saida_pista'] = le.fit_transform(X['ocorrencia_saida_pista'])
X.head(10)
```

[12]:
```
   ocorrencia_saida_pista aeronave_tipo_veiculo aeronave_motor_tipo
0                       0                AVIÃO             PISTÃO
1                       0                AVIÃO          TURBOÉLICE
2                       0                AVIÃO          TURBOÉLICE
3                       0                AVIÃO                JATO
4                       0                AVIÃO                JATO
5                       0                AVIÃO                JATO
6                       0                AVIÃO             PISTÃO
7                       0          HELICÓPTERO           TURBOEIXO
8                       0            ULTRALEVE             PISTÃO
9                       0          HELICÓPTERO           TURBOEIXO
```

[13]:
```
#Tratando as variáveis que possuem mais de dois tipos:
x_final = pd.get_dummies (data = X, columns =␣
 ↪['aeronave_tipo_veiculo','aeronave_motor_tipo'] )
x_final
```

[13]:
```
      ocorrencia_saida_pista  aeronave_tipo_veiculo_ANFÍBIO  \
0                          0                              0
1                          0                              0
2                          0                              0
3                          0                              0
4                          0                              0
...                      ...                            ...
5747                       0                              0
5748                       0                              0
5749                       0                              0
5750                       0                              0
5751                       0                              0
```

```
      aeronave_tipo_veiculo_AVIÃO  aeronave_tipo_veiculo_BALÃO  \
0                               1                            0
1                               1                            0
2                               1                            0
3                               1                            0
4                               1                            0
...                           ...                          ...
5747                            1                            0
5748                            0                            0
5749                            1                            0
5750                            0                            0
5751                            0                            0

      aeronave_tipo_veiculo_DIRIGÍVEL  aeronave_tipo_veiculo_HELICÓPTERO  \
0                                   0                                  0
1                                   0                                  0
2                                   0                                  0
3                                   0                                  0
4                                   0                                  0
...                               ...                                ...
5747                                0                                  0
5748                                0                                  0
5749                                0                                  0
5750                                0                                  1
5751                                0                                  0

      aeronave_tipo_veiculo_HIDROAVIÃO  aeronave_tipo_veiculo_NÃO INFORMADO  \
0                                    0                                   0
1                                    0                                   0
2                                    0                                   0
3                                    0                                   0
4                                    0                                   0
...                                ...                                 ...
5747                                 0                                   0
5748                                 0                                   1
5749                                 0                                   0
5750                                 0                                   0
5751                                 0                                   1

      aeronave_tipo_veiculo_PLANADOR  aeronave_tipo_veiculo_TRIKE  \
0                                  0                           0
1                                  0                           0
2                                  0                           0
3                                  0                           0
4                                  0                           0
...                              ...                         ...
```

```
5747                                0                                0
5748                                0                                0
5749                                0                                0
5750                                0                                0
5751                                0                                0

      aeronave_tipo_veiculo_ULTRALEVE  aeronave_motor_tipo_JATO  \
0                                   0                         0
1                                   0                         0
2                                   0                         0
3                                   0                         1
4                                   0                         1
...                               ...                       ...
5747                                0                         1
5748                                0                         1
5749                                0                         0
5750                                0                         0
5751                                0                         0

      aeronave_motor_tipo_NÃO INFORMADO  aeronave_motor_tipo_PISTÃO  \
0                                     0                           1
1                                     0                           0
2                                     0                           0
3                                     0                           0
4                                     0                           0
...                                 ...                         ...
5747                                  0                           0
5748                                  0                           0
5749                                  0                           1
5750                                  0                           0
5751                                  1                           0

      aeronave_motor_tipo_SEM TRAÇÃO  aeronave_motor_tipo_TURBOEIXO  \
0                                   0                             0
1                                   0                             0
2                                   0                             0
3                                   0                             0
4                                   0                             0
...                               ...                           ...
5747                                0                             0
5748                                0                             0
5749                                0                             0
5750                                0                             1
5751                                0                             0

      aeronave_motor_tipo_TURBOÉLICE
0                                   0
```

```
1                                    1
2                                    1
3                                    0
4                                    0
...                                ...
5747                                 0
5748                                 0
5749                                 0
5750                                 0
5751                                 0

[5752 rows x 17 columns]
```

[14]:
```
# Separando em dados teste e treinamento
x_treino, x_teste, y_treino, y_teste  = train_test_split(x_final, y, test_size␣
 ↪= 0.25, random_state = 1)
```

### 1.9 Modelo KNN

[15]:
```
#configurando o modelo
knn = KNeighborsClassifier()
knn.fit(x_treino, y_treino)
```

[15]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

[16]:
```
#matriz de confusão
resultado_knn = knn.predict(x_teste)
print (pd.crosstab(y_teste, resultado_knn, rownames = ['Real'], colnames =␣
 ↪['Predito'], margins = True))
```

```
Predito  NÃO  SIM   All
Real
NÃO      560  428   988
SIM      113  337   450
All      673  765  1438
```

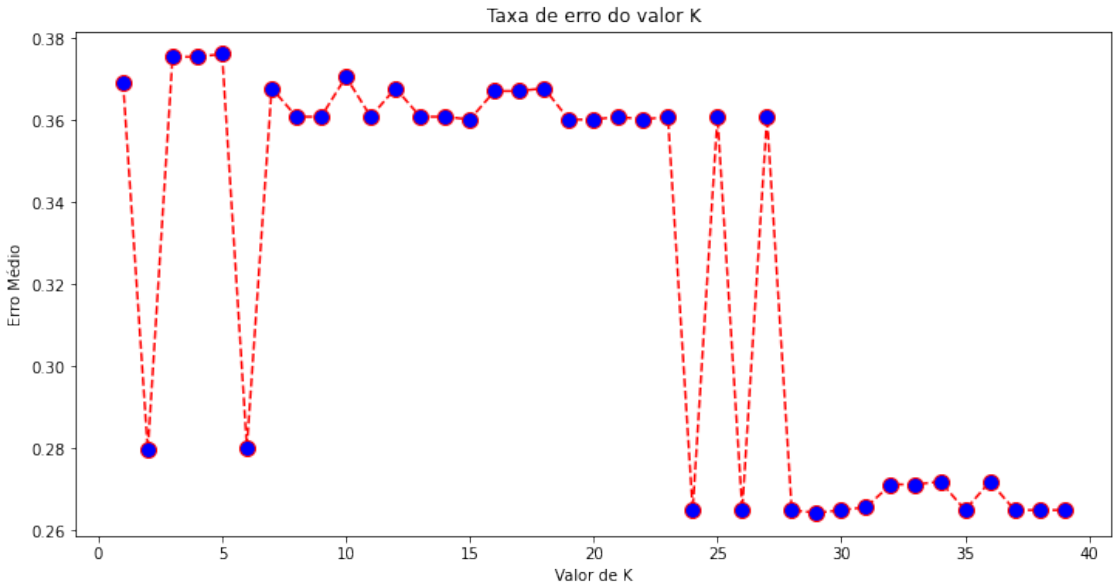
[17]:
```
#encontrando o melhor K para valores entre 1 e 40
error = []

for i in range(1, 40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_treino, y_treino)
    pred_i = knn.predict(x_teste)
    error.append(np.mean(pred_i != y_teste))

plt.figure(figsize=(12, 6))
```

```
plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Taxa de erro do valor K')
plt.xlabel('Valor de K')
plt.ylabel('Erro Médio')
```

[17]:
```
Text(0, 0.5, 'Erro Médio')
```




Observando o gráfico é possivel ver que o K=29 apresentou o menor erro médio

[18]:
```
#Utiliznado o K encontrado:
knn = KNeighborsClassifier(n_neighbors=29)
knn.fit(x_treino, y_treino)
```

[18]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=29, p=2,
                     weights='uniform')
```

[19]:
```
resultado_knn = knn.predict(x_teste)
print (pd.crosstab(y_teste,resultado_knn, rownames=['Real'],␣
 ↪colnames=['Predito'], margins=True))
```

```
Predito   NÃO  SIM   All
Real
NÃO       911   77   988
SIM       303  147   450
All      1214  224  1438
```

### 1.10 Modelo Random Forest

[20]:
```
# Treinando o modelo
rf = RandomForestClassifier (n_estimators = 1000, random_state = 42)
rf.fit (x_treino, y_treino);
```

[21]:
```
y_rf = rf.predict(x_teste)
print(pd.crosstab(y_teste,y_rf, rownames=['Real'], colnames=['Predito'],␣
 ↪margins=True))
print('\n')
print(classification_report(y_teste, y_rf))
```

```
Predito   NÃO  SIM   All
Real
NÃO       916   72   988
SIM       306  144   450
All      1222  216  1438


              precision    recall  f1-score   support

         NÃO       0.75      0.93      0.83       988
         SIM       0.67      0.32      0.43       450

    accuracy                           0.74      1438
   macro avg       0.71      0.62      0.63      1438
weighted avg       0.72      0.74      0.70      1438
```

O modelo de Random Forest conseguiu melhores resultados, porém ainda com bastante falsos negativos.Tentar um modelo mais robusto e que não sofre tanta influência em amostras desbalanceadas

[22]:
```
#Analisando a importância de cada variável
pd.set_option('display.float_format', lambda x: '%.3f' % x) #configurando␣
 ↪pandas para mostrar valor inteiro ao inves notação cientifica
rf.feature_importances_

feature_importances = pd.DataFrame(rf.feature_importances_, index = x_treino.
 ↪columns, columns=['importância']).sort_values('importância',ascending=False)
feature_importances
```

[22]:

| | importância |
|---|---|
| aeronave_motor_tipo_PISTÃO | 0.378 |
| aeronave_motor_tipo_JATO | 0.216 |
| aeronave_tipo_veiculo_ULTRALEVE | 0.082 |
| aeronave_tipo_veiculo_AVIÃO | 0.081 |
| ocorrencia_saida_pista | 0.071 |
| aeronave_motor_tipo_TURBOÉLICE | 0.052 |
| aeronave_motor_tipo_TURBOEIXO | 0.039 |

```
aeronave_tipo_veiculo_HELICÓPTERO          0.030
aeronave_motor_tipo_NÃO INFORMADO          0.019
aeronave_tipo_veiculo_NÃO INFORMADO        0.009
aeronave_tipo_veiculo_ANFÍBIO              0.006
aeronave_tipo_veiculo_PLANADOR             0.005
aeronave_motor_tipo_SEM TRAÇÃO             0.005
aeronave_tipo_veiculo_TRIKE                0.004
aeronave_tipo_veiculo_BALÃO                0.001
aeronave_tipo_veiculo_HIDROAVIÃO           0.000
aeronave_tipo_veiculo_DIRIGÍVEL            0.000
```

### 1.11 Modelo XGBoost

[23]:
```
# ajuste do modelo
xgb = XGBClassifier(learning_rate =0.1,
 n_estimators=1000,
 max_depth=6,
 min_child_weight=1,
 gamma=0,
 subsample=0.8,
 colsample_bytree=0.8,
 objective= 'binary:logistic',
 nthread=4,
 scale_pos_weight=1.0,
 seed=27)
xgb.fit(x_treino, y_treino)
```

[23]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.8, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=None, n_estimators=1000, n_jobs=1,
              nthread=4, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1.0, seed=27,
              silent=None, subsample=0.8, verbosity=1)
```

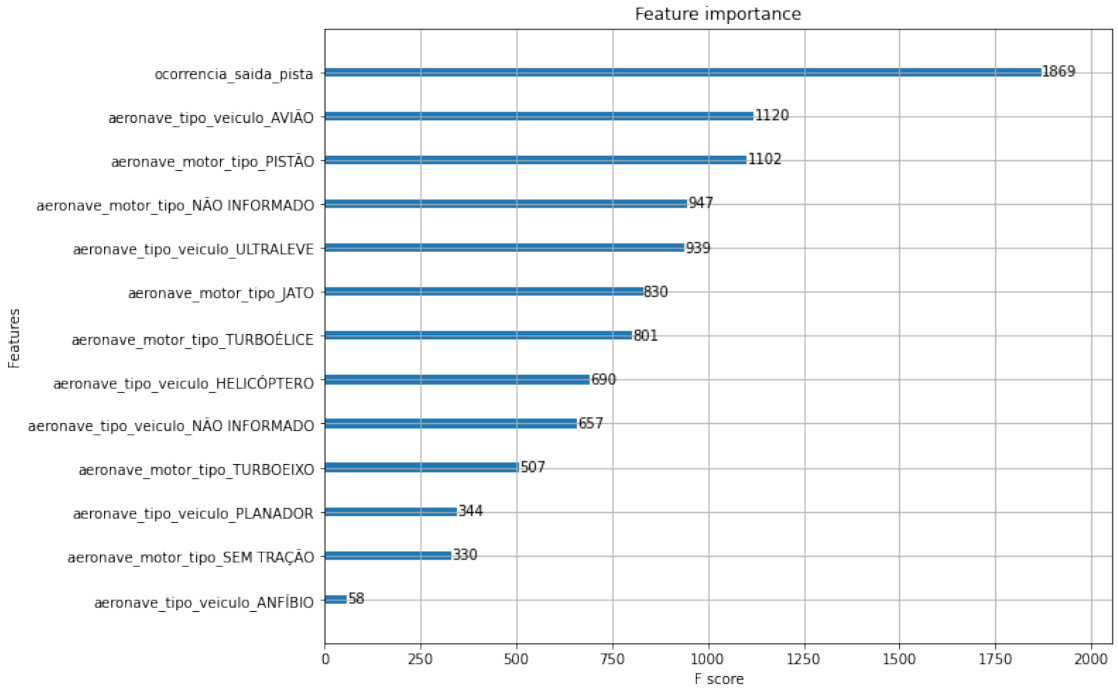
[24]:
```
#fazendo as predições no dados de teste
preditos_xgb = xgb.predict(x_teste)
print (pd.crosstab(y_teste,preditos_xgb, rownames=['Real'],␣
 ↪colnames=['Predito'], margins=True))
```

```
Predito   NÃO  SIM   All
Real
NÃO       912   76   988
SIM       303  147   450
All      1215  223  1438
```

[25]:
```
#Analisando a importância de cada variável
#from xgboost import plot_importance
```

```
fig, ax = plt.subplots(figsize=(10,8))
plot_importance(xgb, ax=ax)
```

[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2177e1a290>




## 1.12 Métodos Ensemble: combinando os dois melhores modelos, Random Forest e XGBoost.

[26]:
```
# Voting Classifier with soft voting
voto = VotingClassifier(estimators=[('rf', rf),('xgb',xgb)], voting='soft')
voto = voto.fit(x_treino,y_treino)

y_predito = voto.predict(x_teste)
print(pd.crosstab(y_teste,y_predito, rownames=['Real'], colnames=['Predito'],␣
 ↪margins=True))
print('\n')
print(classification_report(y_teste, y_predito))
```

```
Predito   NÃO  SIM   All
Real
NÃO       912   76   988
SIM       302  148   450
All      1214  224  1438
```

```
              precision    recall  f1-score   support

         NÃO       0.75      0.92      0.83       988
         SIM       0.66      0.33      0.44       450

    accuracy                           0.74      1438
   macro avg       0.71      0.63      0.63      1438
weighted avg       0.72      0.74      0.71      1438
```

Levando a probabilidade de volta para o df

[27]:
```
df_ocorrencias['probabilidade'] = voto.predict_proba(x_final[x_treino.
 ↪columns])[:,1]
df_ocorrencias
```

[27]:
```
     ocorrencia_saida_pista aeronave_tipo_veiculo aeronave_motor_tipo  \
0                       NÃO                AVIÃO              PISTÃO
1                       NÃO                AVIÃO           TURBOÉLICE
2                       NÃO                AVIÃO           TURBOÉLICE
3                       NÃO                AVIÃO                 JATO
4                       NÃO                AVIÃO                 JATO
...                     ...                  ...                  ...
5747                    NÃO                AVIÃO                 JATO
5748                    NÃO        NÃO INFORMADO                 JATO
5749                    NÃO                AVIÃO              PISTÃO
5750                    NÃO          HELICÓPTERO            TURBOEIXO
5751                    NÃO        NÃO INFORMADO       NÃO INFORMADO

     acidente  probabilidade
0         SIM          0.409
1         NÃO          0.144
2         NÃO          0.144
3         NÃO          0.015
4         NÃO          0.015
...       ...            ...
5747      NÃO          0.015
5748      NÃO          0.006
5749      SIM          0.409
5750      NÃO          0.227
5751      SIM          0.318

[5752 rows x 5 columns]
```

### 1.13 Tratando DF final

[28]:
```
df_final =␣
 ↪df_ocorrencias[['aeronave_tipo_veiculo','aeronave_motor_tipo','ocorrencia_saida_pista','pro
 ↪copy()
df_final.
 ↪sort_values(by=['probabilidade','aeronave_tipo_veiculo','aeronave_motor_tipo'],␣
 ↪ascending=[False,True,True],inplace=True)
df_final.
 ↪drop_duplicates(['probabilidade','aeronave_tipo_veiculo','aeronave_motor_tipo'],␣
 ↪keep='first', inplace=True)
df_final = df_final.rename(columns={'aeronave_tipo_veiculo': 'Tipo de Veículo',␣
 ↪'aeronave_motor_tipo': 'Tipo de Motor',
                                    'ocorrencia_saida_pista':'Acidente na saída␣
 ↪de pista', 'probabilidade':'Probabilidade de ser Acidente'})

#remove as linhas que tipo de veiculo nao foi informado
df_final = df_final[~df_final['Tipo de Veículo'].str.contains('NÃO INFORMADO')]

df_final.reset_index(drop=True, inplace=True)
```

[29]:
```
df_final
```

[29]:

| | Tipo de Veículo | Tipo de Motor | Acidente na saída de pista |
|---|---|---|---|
| 0 | TRIKE | NÃO INFORMADO | NÃO |
| 1 | PLANADOR | NÃO INFORMADO | NÃO |
| 2 | PLANADOR | SEM TRAÇÃO | SIM |
| 3 | AVIÃO | NÃO INFORMADO | SIM |
| 4 | BALÃO | SEM TRAÇÃO | NÃO |
| 5 | HELICÓPTERO | PISTÃO | NÃO |
| 6 | ANFÍBIO | TURBOÉLICE | NÃO |
| 7 | ULTRALEVE | PISTÃO | NÃO |
| 8 | ULTRALEVE | NÃO INFORMADO | NÃO |
| 9 | PLANADOR | SEM TRAÇÃO | NÃO |
| 10 | AVIÃO | PISTÃO | SIM |
| 11 | ULTRALEVE | PISTÃO | SIM |
| 12 | TRIKE | PISTÃO | NÃO |
| 13 | DIRIGÍVEL | PISTÃO | NÃO |
| 14 | HIDROAVIÃO | PISTÃO | NÃO |
| 15 | AVIÃO | NÃO INFORMADO | NÃO |
| 16 | AVIÃO | PISTÃO | NÃO |
| 17 | AVIÃO | TURBOÉLICE | SIM |
| 18 | AVIÃO | JATO | SIM |
| 19 | HELICÓPTERO | TURBOEIXO | NÃO |
| 20 | HELICÓPTERO | TURBOÉLICE | NÃO |
| 21 | ANFÍBIO | PISTÃO | NÃO |
| 22 | AVIÃO | TURBOÉLICE | NÃO |
| 23 | ANFÍBIO | PISTÃO | SIM |

| | | | | |
|---|---|---|---|---|
| 24 | AVIÃO | JATO | | NÃO |
| 25 | AVIÃO | TURBOEIXO | | NÃO |

| | Probabilidade de ser Acidente |
|---|---|
| 0 | 0.985 |
| 1 | 0.902 |
| 2 | 0.901 |
| 3 | 0.895 |
| 4 | 0.893 |
| 5 | 0.747 |
| 6 | 0.717 |
| 7 | 0.716 |
| 8 | 0.626 |
| 9 | 0.597 |
| 10 | 0.559 |
| 11 | 0.507 |
| 12 | 0.507 |
| 13 | 0.460 |
| 14 | 0.460 |
| 15 | 0.439 |
| 16 | 0.409 |
| 17 | 0.352 |
| 18 | 0.323 |
| 19 | 0.227 |
| 20 | 0.224 |
| 21 | 0.202 |
| 22 | 0.144 |
| 23 | 0.100 |
| 24 | 0.015 |
| 25 | 0.003 |