# Appendix

Here is our code. The database can be found at [46] and [50].

import sklearn

from sklearn.model\_selection import train\_test\_split

import pandas as pd

import numpy as np

import shap

import xgboost as xgb

from matplotlib import pyplot

from sklearn.metrics import accuracy\_score

from sklearn.metrics import plot\_confusion\_matrix

from sklearn.metrics import accuracy\_score

from imblearn.over\_sampling import SMOTE

from imblearn.over\_sampling import BorderlineSMOTEIn

wildfire=pd.read\_excel('Portugal ABCDEF.xlsx')

wildfire

|  | **FFMC** | **DMC** | **DC** | **ISI** | **Temperature** | **Relative humidity** | **Wind** | **Class** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 83.0 | 23.3 | 85.3 | 2.3 | 16.7 | 20 | 3.1 | A |
| **1** | 63.5 | 70.8 | 665.3 | 0.8 | 17.0 | 72 | 6.7 | A |
| **2** | 90.1 | 108.0 | 529.8 | 12.5 | 14.7 | 66 | 2.7 | A |
| **3** | 94.8 | 227.0 | 706.7 | 12.0 | 23.3 | 34 | 3.1 | A |
| **4** | 94.8 | 227.0 | 706.7 | 12.0 | 25.0 | 36 | 4.0 | A |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **512** | 92.5 | 121.1 | 674.4 | 8.6 | 18.2 | 46 | 1.8 | E |
| **513** | 91.0 | 129.5 | 692.6 | 7.0 | 18.8 | 40 | 2.2 | E |
| **514** | 89.2 | 103.9 | 431.6 | 6.4 | 22.6 | 57 | 4.9 | E |
| **515** | 94.8 | 222.4 | 698.6 | 13.9 | 27.5 | 27 | 4.9 | F |
| **516** | 92.5 | 121.1 | 674.4 | 8.6 | 25.1 | 27 | 4.0 | F |

517 rows × 8 columns

x=wildfire.drop(['Class'],axis=1)

y=wildfire['Class']

oversampled = SMOTE(sampling\_strategy='auto',

random\_state=5,k\_neighbors = 1

)

x, y = oversampled.fit\_resample(x, y)

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.250,random\_state=10)

y\_train.value\_counts()

wildfire.info()

wildfire.isnull().any()

FFMC False

DMC False

DC False

ISI False

Temperature False

Relative humidity False

Wind False

Class False

dtype: bool

xgbc=xgb.XGBClassifier(objective='multi:softprob',

learning\_rate =0.6,

n\_estimators=800,

max\_depth=6,

min\_child\_weight=0,

gamma=0.2,

subsample=0.9,

colsample\_bytree=0.7,

nthread=40,

seed=230)

xgbc.fit(x\_train,y\_train)

predictions = xgbc.predict(x\_test)

accuracy = accuracy\_score(y\_test, predictions)

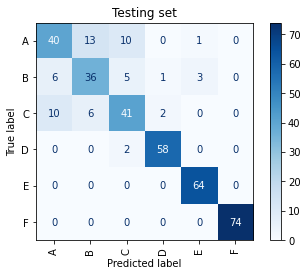
print("Accuracy: %.2f%%" % (accuracy \* 100.0))

Accuracy: 84.14%

class\_names = ['A', 'B', 'C','D','E','F']

disp = plot\_confusion\_matrix(xgbc, x\_test, y\_test, display\_labels=class\_names, cmap=pyplot.cm.Blues, xticks\_rotation='vertical')

pyplot.title('Testing set')



disp = plot\_confusion\_matrix(xgbc, x\_train, y\_train, display\_labels=class\_names, cmap=pyplot.cm.Blues, xticks\_rotation='vertical')

pyplot.title('Training set')

Calendar

Description automatically generated

shap\_values = shap.TreeExplainer(xgbc).shap\_values(x\_test)

shap.summary\_plot(shap\_values, x\_test,class\_names = class\_names, plot\_type='bar')Chart, bar chart

Description automatically generated

shap.summary\_plot(shap\_values[0], x\_test, class\_names=class\_names,show=False)

pyplot.gcf().axes[-1].set\_box\_aspect(50)

pyplot.gcf().axes[-1].set\_aspect(100)

pyplot.gcf().axes[-1].set\_box\_aspect(100)A picture containing text

Description automatically generated

shap.summary\_plot(shap\_values[1], x\_test, class\_names=class\_names,show=False)

pyplot.gcf().axes[-1].set\_box\_aspect(50)

pyplot.gcf().axes[-1].set\_aspect(100)

pyplot.gcf().axes[-1].set\_box\_aspect(100)Graphical user interface, text

Description automatically generated

shap.summary\_plot(shap\_values[2], x\_test, class\_names=class\_names,show=False)

pyplot.gcf().axes[-1].set\_box\_aspect(50)

pyplot.gcf().axes[-1].set\_aspect(100)

pyplot.gcf().axes[-1].set\_box\_aspect(100)Graphical user interface, text

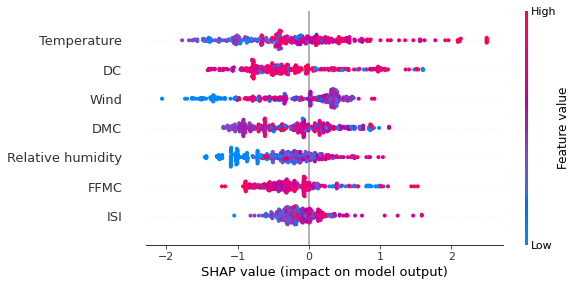
Description automatically generated

shap.summary\_plot(shap\_values[3], x\_test, class\_names=class\_names,show=False)

pyplot.gcf().axes[-1].set\_box\_aspect(50)

pyplot.gcf().axes[-1].set\_aspect(100)

pyplot.gcf().axes[-1].set\_box\_aspect(100)



shap.summary\_plot(shap\_values[4], x\_test, class\_names=class\_names,show=False)

pyplot.gcf().axes[-1].set\_box\_aspect(50)

pyplot.gcf().axes[-1].set\_aspect(100)

pyplot.gcf().axes[-1].set\_box\_aspect(100)

Graphical user interface

Description automatically generated with medium confidence

shap.summary\_plot(shap\_values[5], x\_test, class\_names=class\_names,show=False)

pyplot.gcf().axes[-1].set\_box\_aspect(50)

pyplot.gcf().axes[-1].set\_aspect(100)

pyplot.gcf().axes[-1].set\_box\_aspect(100)Graphical user interface, chart

Description automatically generated