#### **Appendix F: Random Forest Model**

# Random Forest built using Python on Google Collaboratory

# load necessary packages

from sklearn.ensemble import RandomForestClassifier

import numpy as np

from pandas import read\_csv

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import cross\_val\_score

# load dataset

from google.colab import files

file = files.upload()

dataframe = read\_csv("**DATAFILE.csv**")

dataset = dataframe.values

# split into input (X) and output (Y) variables

X = dataset[:,0:15].astype(float)

Y = dataset[:,15]

# encode class values as integers

encoder = LabelEncoder()

encoder.fit(Y)

encoded\_Y = encoder.transform(Y)

# Build Classification Task

# Determine best value for the 'n\_estimators' (number of trees) parameter

n\_estimators = [50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 325, 350]

train\_resultsa = []

test\_resultsa = []

for estimator in n\_estimators:

foresta = RandomForestClassifier(n\_estimators=estimator,

random\_state=0)

foresta.fit(X, encoded\_Y)

train\_resultsa.append((np.mean(foresta.predict(X) == encoded\_Y)\*100))

test\_resultsa.append((round(100\*np.mean(cross\_val\_score(foresta, X, encoded\_Y, cv=10, n\_jobs=-1)),2)))

from matplotlib.legend\_handler import HandlerLine2D

import matplotlib.pyplot as plt

line1, = plt.plot(n\_estimators, train\_resultsa, "b", label="Training Accuracy")

line2, = plt.plot(n\_estimators, test\_resultsa, "r", label="Cross Validation Accuracy")

plt.legend(handler\_map={line1: HandlerLine2D(numpoints=2)})

plt.ylabel("Accuracy")

plt.xlabel("Number of Trees")

plt.show()

# Determine best value for the 'cv' (K-fold) parameter

cvs = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

train\_resultsb = []

test\_resultsb = []

for cv in cvs:

forestb = RandomForestClassifier(n\_estimators=250,

random\_state=0)

forestb.fit(X, encoded\_Y)

train\_resultsb.append((np.mean(forestb.predict(X) == encoded\_Y)\*100))

test\_resultsb.append((round(100\*np.mean(cross\_val\_score(forestb, X, encoded\_Y, cv=cv, n\_jobs=-1)),2)))

from matplotlib.legend\_handler import HandlerLine2D

import matplotlib.pyplot as plt

line2, = plt.plot(cvs, test\_resultsb, "r", label="Cross Validation Accuracy")

plt.legend(handler\_map={line1: HandlerLine2D(numpoints=2)})

plt.ylabel("Accuracy")

plt.xlabel("Number of Folds (K-values)")

plt.show()

# Run model using optimum features

forestc = RandomForestClassifier(n\_estimators=250,

random\_state=0)

forestc.fit(X, encoded\_Y)

print('Training accuracy: ', np.mean(forestc.predict(X) == encoded\_Y)\*100)

print ('Cross validate accuracy: ',round(100\*np.mean(cross\_val\_score(forestc, X, encoded\_Y, cv=13, n\_jobs=-1)),2))

importance\_vals = forestc.feature\_importances\_

print(importance\_vals)

Name = list()

for col in dataframe.columns:

Name.append(col)

# Plot Relative importance

import matplotlib.pyplot as plt

plt.title("Neural network relative importance")

plt.bar(Name[0:15], importance\_vals, label='Relative importance')

plt.legend()