**LITHOPY INSTRUCTION GUIDE**

**LANGUAGE MAIN:** Python

**INSTALL PACKAGES:** pip install requirements.txt

**DESCRIPTION OF THE FOLDERS:**

- input-test-data: folder that contains the files for input

- source code: folder containing the source codes of the algorithms

- summary-output-file: folder containing the output summary file / algorithm application results

**SOFTWARE AND MAIN LIBRARIES**

- Anaconda (>=5.3) with Jupyter Notebook (>=5.2)

- sklearn (>=0.20), NumPy (>=1.8.2), SciPy (>= 0.13.3) and Matplotlib (>=3.0.0)

**CONFIGURE CODE MODEL**

**FOR TEMPLATES 1, 2 AND 3:**

#input file:

train = pd.read\_csv("<full file location address> /template<numTemplate>\_full\_<Group>.csv",sep=",",error\_bad\_lines=False)

#------------

Configure output file:

<full file location address> - complete location of file location on computer – folder input-test-data

<numTemplate> - number the template (1, 2 and 3)

<Group> - Group code - GP, G1, G2 and G3

**Algorithm MLP:**

#conducts training:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20,random\_state=27)

classifier\_dt = MLPClassifier(solver='lbfgs', activation='relu',random\_state=27)

**Algorithm DecisionTree**

#conducts training:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20, random\_state=8)

classifier\_dt = tree.DecisionTreeClassifier(random\_state=8,max\_depth=20,criterion='entropy')

**Algorithm RandomForest**

#conducts training:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20,random\_state=42)

classifier\_dt = RandomForestClassifier(max\_depth=20, n\_estimators=1000, random\_state=42, n\_jobs=-1)

**Algorithm SVM**

#conducts training:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20, random\_state=20)

classifier\_dt = SVC()

**FOR TEMPLATE PRACTICAL:**

#output files:

files1 = <full file location address\_1>/output\_expedition<Exp>\_<Site>\_<Group>\_result\_<Algorithm>

.csv'

files2 = ‘<full file location address\_1>/output\_expedition<Ex>\_<Site>\_<Group>\_classification\_report\_<Algorithm>.csv'

#--------------

#input files:

fileTrain = "<full file location address\_2>/ templatePractical\_Train\_<Group>.csv"

fileTest = "<full file location address\_2>/importIODP<Exp>\_<Site>\_<Group>.csv"

#-------------

Configure output file:

<full file location address\_1> - complete location of file location on computer – folder input-test-data

<full file location address\_2> - complete location of file location on computer – folder summary-output-file

<Exp> - Expedition

<Site> - Site Code

<Group> - Group code - GP, G1, G2 and G3

<Algorithm> - algorithm: MLP, DecTree, RandF, SVM

**Algorithm MLP:**

- File: modelMLP.ipynb

# Add the following line at the beginning of the file

from sklearn.neural\_network import MLPClassifier

#conducts training:

classifier\_dt = MLPClassifier(solver='lbfgs', activation='relu',random\_state=8)

**Algorithm DecisionTree**

- File: modelDecisionTree. ipynb

# Add the following line at the beginning of the file

from sklearn.tree import DecisionTreeClassifier

#conducts training:

classifier\_dt = tree.DecisionTreeClassifier(random\_state=8,max\_depth=20,criterion='entropy')

**Algorithm RandomForest**

- File: modelRandomF. ipynb

# Add the following line at the beginning of the file

from sklearn.ensemble import RandomForestClassifier

#conducts training:

classifier\_dt = RandomForestClassifier(max\_depth=20, n\_estimators=1000, random\_state=8, n\_jobs=-1)

**Algorithm SVM**

- File: modelSVM. ipynb

# Add the following line at the beginning of the file

from sklearn.svm import SVC

#conducts training:

classifier\_dt = SVC()