EXERCISE OF MACHINE LEARNING USING SOME PYTHON TOOLS AND TECHNIQUES

Ertan Mustafa Geldiev[1], **Nayden Valkov Nenkov**[2], **Mariana Mateeva Petrova**[3]

Abstract: One of the goals of predictive analytics training using Python tools is to create a "Model" from classified examples that classifies new examples from a Dataset. The purpose of different strategies and experiments is to create a more accurate prediction model. The goals we set out in the study are to achieve successive steps to find an accurate model for a dataset and preserving it for its subsequent use using the python instruments. Once we have found the right model, we save it and load it later, to classify if we have "phishing" in our case. In the case that the path we reach to the discovery of the search model, we can ask ourselves how much we can automate everything and whether a computer program can be written to automatically go through the unified steps and to find the right model? Due to the fact that the steps for finding the exact model are often unified and repetitive for different types of data, we have offered a hypothetical algorithm that could write a complex computer program searching for a model, for example when we have a classification task. This algorithm is rather directional and does not claim to be all-encompassing. The research explores some features of Python Scientific Python Packages like Numpy, Pandas, Matplotlib, Scipy and scycit-learn to create a more accurate model. The Dataset used for the research was downloaded free from the UCI Machine Learning Repository (UCI Machine Learning Repository, 2017).

UDC Classification: 004; **DOI:** http://dx.doi.org/10.12955/cbup.v6.1295

Keywords: machine learning, Predictive Analytics Training with Python, data sets.

Introduction

Mandatory elements for the study:

- The name of the **Dataset** used in the research is Website Phishing Data Set (UCI Machine Learning Repository, 2017) "In the dataset used after processing with appropriate software, the authors conclude that out of a total of 1353 analyzed web sites 548 can be considered real, 702 fraudulent - and 103 unclear status. In our case we have a classification problem and our classifier needs to predict the class of each instance "-1" - phishing, "0" suspicious, "1" - non-phishing.") from source Neda Abdelhamid. The data set is in arff format. It has 10 attributes that are most often with values {1, -1, 0} or {1, 0} where in the class result the meaning -1 is that it is fraudulent, 0 is suspicious and 1 is a legal website.
- The basic building blocks of the **Scientific Python ecosystem**:
    - Python, a computing language
    - Core numeric libraries
        - **Numpy**: is an open-source library of python allowing for scientific computations involving multi-dimensional arrays with compiled fast functions for multiple mathematical actions and operations, simulations and more
        - **Scipy**: is an open-source library containing optimization, algebric and other modules that expand numpy with regression functions, fusion transformations and many others
        - **Matplotlib**: has libraries that allow you to paint high quality figures

The successive steps made in the study are similar and some are borrowed fromIdris(2016), as follows:

A. Transforming of data set from ARFF to CSV format

B. Loading in the python program the CSV file

C. Using Descriptive statistics to explain the basic features of data. Quantitative data analysis can be formed by descriptive static data formed by graphs and their characteristics (Segui, 2017).

D. Visualizations

E. Preparing the Input

F. Feature Selection

G. Evaluate Machine Learning Algorithms

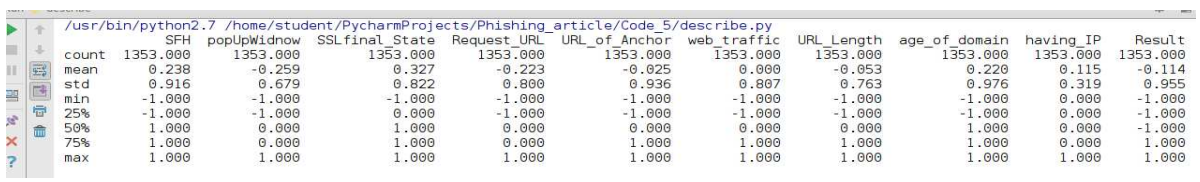
[1] Varna University of Management, PhD student in University of Shumen, Bulgaria, geldiev3@gmail.com
[2] Konstantin Preslavsky University of Shumen, n.nenkov@shu.bg
[3] "St.Cyril and St.Methodius" University of VelikoTarnovo, Bulgaria, petrova@uni-vt.bg

H. Algorithm Evaluation Metrics
I. Finding a suitable Classification Algorithms for our dataset
J. Comparing Machine Learning Algorithms
K. Automating Machine Learning Workflows with Pipelines
L. Saving and Loading Machine Learning Models

Actions taken over the data set and results

A. Easy step in our case. ARFF or Attribute-Relation File Format has two main parts head and data. We take the data part and get the csv file.
B. Loading csv file produce shape (1353, 10) where 1353 is the number of instances and 10 is the number of attributes
C. Descriptive statistics – using Pandas, print data.head(15) gave us the first 15 rows, print data.shape gave us the dimensions of the data, data.dtypes gave us the types of data, and the describe() function of Pandas shows the results in Figure 1.

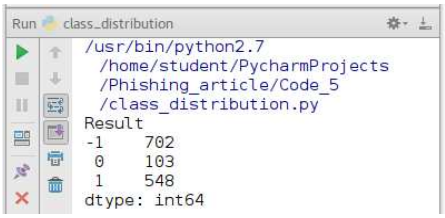
Figure 1: Descriptive Statistics



```
/usr/bin/python2.7 /home/student/PycharmProjects/Phishing_article/Code_5/describe.py
              SFH  popUpWidnow  SSLfinal_State  Request_URL  URL_of_Anchor  web_traffic  URL_Length  age_of_domain  having_IP     Result
count  1353.000     1353.000        1353.000     1353.000       1353.000     1353.000    1353.000       1353.000   1353.000  1353.000
mean      0.238       -0.259           0.327       -0.223         -0.025        0.000      -0.053          0.220      0.115    -0.114
std       0.916        0.679           0.822        0.800          0.936        0.807       0.763          0.976      0.319     0.955
min      -1.000       -1.000          -1.000       -1.000         -1.000       -1.000      -1.000         -1.000      0.000    -1.000
25%      -1.000       -1.000           0.000       -1.000         -1.000       -1.000      -1.000         -1.000      0.000    -1.000
50%       1.000        0.000           1.000        0.000          0.000        0.000       0.000          1.000      0.000    -1.000
75%       1.000        0.000           1.000        0.000          1.000        1.000       1.000          1.000      0.000     1.000
max       1.000        1.000           1.000        1.000          1.000        1.000       1.000          1.000      1.000     1.000
```

Print data.groupby('Result').size() gave us only the class. The values are shown in Figure 2.

Figure 2: Output of reviewing a class



```
Run  class_distribution
/usr/bin/python2.7
 /home/student/PycharmProjects
 /Phishing_article/Code_5
 /class_distribution.py
Result
-1    702
 0    103
 1    548
dtype: int64
```

Correlations between attributes are obtained through
data.groupby('Redata.corr(method='pearson')sult').size() and shown in Figure 3.

Figure 3: -1 negative, 1 positive correlation,or 0 no correlation

```
web_traffic          -0.187        -0.140          -0.172        -0.046          -0.092
URL_Length            0.152         0.136           0.095         0.048           0.098
age_of_domain         0.163         0.077           0.208         0.053           0.069
having_IP_Address     0.043         0.123           0.108         0.013           0.039
Result               -0.678        -0.510          -0.519        -0.272          -0.287

                     URL_Length  age_of_domain  having_IP_Address  Result
SFH                       0.152          0.163              0.043  -0.678
popUpWidnow               0.136          0.077              0.123  -0.510
SSLfinal_State            0.095          0.208              0.108  -0.519
Request_URL               0.048          0.053              0.013  -0.272
URL_of_Anchor             0.098          0.069              0.039  -0.287
web_traffic              -0.108         -0.684             -0.095   0.244
URL_Length                1.000          0.075              0.165  -0.183
age_of_domain             0.075          1.000              0.040  -0.232
having_IP_Address         0.165          0.040              1.000  -0.059
Result                   -0.183         -0.232             -0.059   1.000
```

Negative or positive gradient of attributes is visible through the skew() function of the Pandas dataframe.

D. Visualizations

Univariate Plots - Histograms – function hist(), ; Density Plots Figure1 - plot(kind='density', ...), ; Box and Whisker Plots plot(kind='box', ...)

A Correlation Matrix Plot is shown in Figure 5. When a correlation between attributes is high, algorithms with regressions have worse productiveness. The relationship between the variables, their connection direction and the correlation between the attributes can be seen through the correlation matrix. (Brownlee, 2017).

Figure 4: Density plots and Multivariate Plots

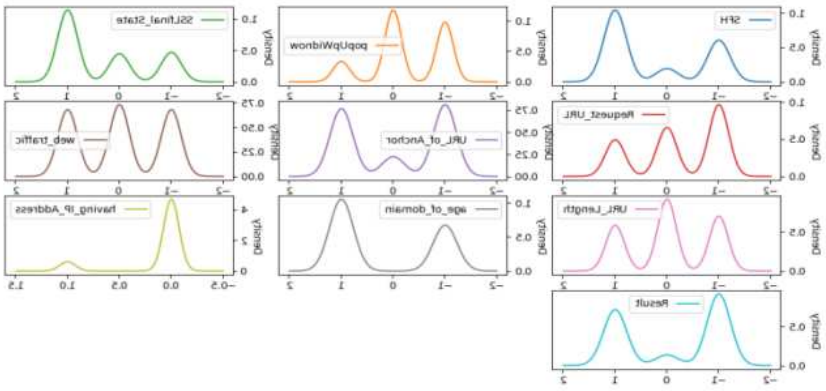


Figure 5: Correlation matrix plot

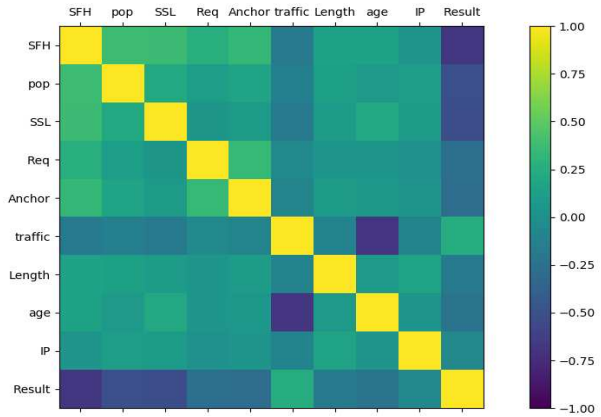
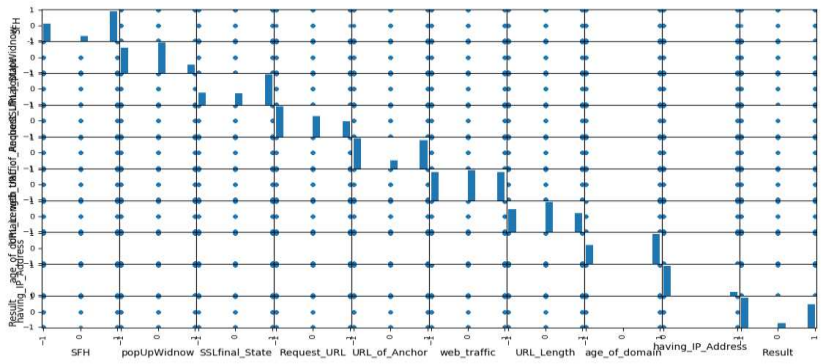


Figure 6: Scatter plot matrix



E. Preparing Data for Machine Learning

The preparation of the input real data supports the successful results in their subsequent analysis. According to the documentation from Guido and et.(2016), we have several basic data transformation capabilities like - standardization, mean removal and variance scaling, non-linear transformation, normalization, binarization, encoding categorical feature, the imputing of missing values, and the generation of polynomial features.

More about these definitions can be read in the paper by Han and Kamber(2006). Rescale Date is used in algorithms that use distance measures like k-Nearest Neighbors and are useful for optimization algorithms used in the core of machine learning algorithms like gradient descent. For example if we binarize our data we will use the function Binarizer() and the result can be seen in Figure 7.

Figure 7: Binarizing or thresholding of 15 rows of our data

```
[1 0 0 0 0 0 0 1 0 1]
[1 0 1 0 0 0 1 1 0 0]
[0 0 1 0 0 0 0 1 0 1]
[0 0 1 0 0 1 0 0 0 1]
[1 0 0 1 0 0 0 1 0 0]
[1 0 1 1 0 0 0 1 1 0]
[0 0 0 0 0 0 0 1 0 0]
[0 0 0 0 1 1 0 0 0 1]
[0 0 0 0 0 0 1 0 0 0 1]
[1 0 1 1 1 0 1 1 0 0]
[1 0 0 0 1 0 1 1 0 1]
[1 0 1 0 0 1 0 0 0 0]
[0 0 0 1 0 0 0 1 0 1]]
```

It is possible, after classification and comparison of the results according to the accuracy of the classified results, that there is no need for prior data processing. Data mining experts are recommended to explore multiple data views and transformations, and then search for appropriate algorithms for them.

F. Feature Selection

The main feature choice is to reduce data overlap, exercise time and increase accuracy. The selection of the relevant elements of the data sets or the accuracy of the evaluators is achieved by using the classes of the sklearn.feature_selection module.

- Univariate feature selection - The SelectKBest class from the scikit-learn library allows you to select functions that are highly related to the output variable, and the determination of these functions is done through statistical tests(Brownlee, 2017).
- Recursive feature elimination (RFE - Feature ranking) RFE recursively removes attributes, leaving only those with high rank, and builds the model with them. For example if we use some model in Figure 8 DecisionTreeClassifier() then RFE class select top attributes. It is interesting to note that if another algorithm for a model is chosen, for example a Logistic regression, the selected 3 elements do not match. According to the algorithm, we get different values of the features.

Figure 8: RFE

```
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_8/recursive_feature_elimination.py
Num Features: 4
Selected Features: [False False False False False  True  True  True  True]
Feature Ranking: [7 6 5 4 3 2 1 1 1 1]
```

- Principal component analysis -Figure 9 (PCA).

Figure 9: PCA

```
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_8/pca.py
Explained Variance: [ 0.34055855  0.17858507  0.12391395]
[[-0.4771223  -0.23149528 -0.31398803 -0.20122731 -0.30060376  0.26559809
  -0.13066557 -0.323868   -0.02338785  0.54430451]
 [ 0.23177007  0.1112422   0.00087279  0.21327735  0.30351422  0.51482906
   0.01395328 -0.70810173 -0.0069288  -0.17204421]
 [-0.1361874  -0.15745477 -0.41515519  0.41040455  0.69587664 -0.14102743
  -0.03132859  0.18127598 -0.01190021  0.2788514811
```

- Feature Importance Bagged decision trees like Random Forest and Extra Trees can be used to estimate the importance of features (Brownlee, 2017). Figure10.

Figure 10: Feature Importance

```
feature_importance | correlation_matrix | correlation_matrix | recursive_feature_elimination
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_8/feature_importance.py
[  1.22422815e-01   5.87577048e-02   2.33549151e-02   1.40660672e-02
   1.84049724e-02   9.43993565e-03   1.09045330e-02   4.58123349e-03
   6.43802114e-04   7.37424022e-01]
```

G. Evaluating Machine Learning Algorithm

- Train and Test sets "We evaluate the percentage error of the classifier with real data or with data that has not yet been used. We assume that these data are a representative set of tests and that they represent the underlying problem." (Nenkov, Dyachenko, Y., Petrova, M., Bondarenko, G.,&Pustovit, 2017) We split our Dataset in two parts 67% train and 33% test and we evaluated the precision of two models in a Logistic regression and Decision tree. For logistic regression, the

probability or response coefficients are modeled on a combination of values taken from predictors (Seguí, 2017).The results are shown in Figure 11.

Figure 11: Accuracy in Decision tree is 100%

```
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_9/train
Accuracy in LogisticRegression: 97.763%
Accuracy in DecisionTreeClassifier: 100.000%
```

- K-fold Cross-Validation of Figure11 with k=10 "In the K-fold cross validation methodology, the cross-checking of the data is subdivided into sub-sets k. These subsets k are used for tests and the other k-1 subunits are pooled to form a training set. Estimation of errors is averaged over all k to obtain the model's performance. This reduces bias and reduces distraction"(Schneider and etc., 1997).

Figure 12: 10-fold validation using models LogisticRegression and Decision tree classifier

```
cross_validation
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_9/cross_valic
Accuracy in LogisticRegression: 97.416% (1.369%)
Accuracy in DecisionTreeClassifier: 100.000% (0.000%)
```

H. Algorithm Evaluation Metrics

We have classification the problem with numeric input values in our Phishing dataset and this means that we need to use classification metrics:

- Classification Accuracy - After using the "cross_val_score" function (Guido, 2016) and using two models in our example, we got the ratio of the right predictions. Figure 13.

Figure 13:Classification Accuracy

```
classification_accuracy
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_10/classification_accur
Accuracy with LogisticRegression: 0.974 (0.014)
Accuracy with Decision Tree: 1.000 (0.000)
```

- Logarithmic Loss: The efficiency of a classification model can also be obtained through "Log Loss", where the predicted input has a probability value between 0 and 1. In principle, the purpose of the models is to minimize this value (Guido , 2016)Figure 14.

Figure 14:Logarithmic Loss with Logistic regression is better than Decision tree log loss

```
classification_logloss
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_10/classification_logl
Logloss with LogisticRegression: -0.339 (0.042)
Logloss with DecisionTree: -2.898 (0.700)
```

- Confusion Matrix: The Confusion Matrix (Hamilton, 2009) is a table that can be used to calculate the performance of a classification model on a dataset. The correct predictions are in the diagonal of the table and it is easy to visually check the predictive error table. Figure 15

Figure 15:Confusion Matrix

```
classification_confusion_matrix
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_10/classification_confus
Confusion matrix with Logistic Regresion
[[240   0   0]
 [  4  20   6]
 [  0   0 177]]
Confusion matrix with Decision tree
[[240   0   0]
 [  0  30   0]
 [  0   0 177]]
```

- Classification Report - "The classification_report() function from the scikit-learn library shows a text report with the basic classification metrics" (Brownlee, 2017).Figure 16

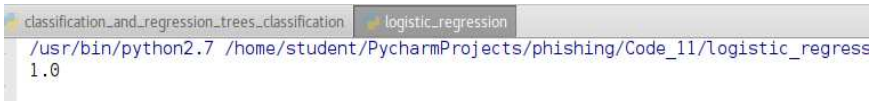
Figure 16:Classification Report

```
Logistic regression report:
             precision    recall  f1-score   support

         -1       0.98      1.00      0.99       240
          0       1.00      0.67      0.80        30
          1       0.97      1.00      0.98       177

avg / total       0.98      0.98      0.98       447

Decision tree report:
             precision    recall  f1-score   support

         -1       1.00      1.00      1.00       240
          0       1.00      1.00      1.00        30
          1       1.00      1.00      1.00       177

avg / total       1.00      1.00      1.00       447
```

I. Finding suitable Classification Algorithms for our dataset

- Logistic Regression: Logistic regression somewhat resembles Linear regression in the sense that their goal is to evaluate the values of the parameters so that at the end of the training model we have a function that best describes the relationship between the known input and output values. Unlike linear regression, the output estimate is transformed using a nonlinear function called a logistic function (Wittenand, 2017).

Figure 17:Logistic regression using LogisticRegression class (Guido, 2016)



- Linear Discriminant Analysis: "When using LDA, it is intended to produce a set of data in a lower-sized space and reduce computational costs. In our example, average predictive accuracy is printed" (Guido, 2016).Figure 18.

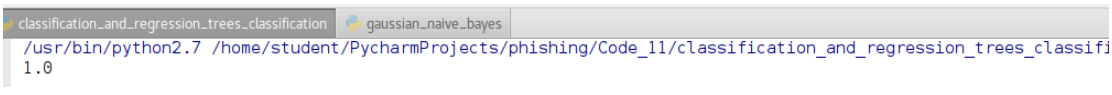
Figure 18: Linear Discriminant Analysis

  - Nonlinear Machine Learning Algorithms

Figure 19: k-Nearest Neighbors KneighborsClassifier()(Guido, 2016)

Figure 20: Naive Bayes GaussianNB()(Guido, 2016)

Figure 21: Classification and Regression Trees DecisionTreeClassifier()(Guido, 2016)



J.Comparing Machine Learning Algorithms and choosing a Machine Learning Model Figure 22

We compared six different classification algorithms with our dataset - Logistic Regression; Linear Discriminant Analysis; k-Nearest Neighbors; Classification and Regression Trees; Naive Bayes; and Support Vector Machines(Brownlee, 2017).

Figure 22: Support Vector Machines SVC()(Guido, 2016)

```
support_vector_machines_classification   gaussian_naive_bayes
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_11/support_vector_machines_classific
1.0
```

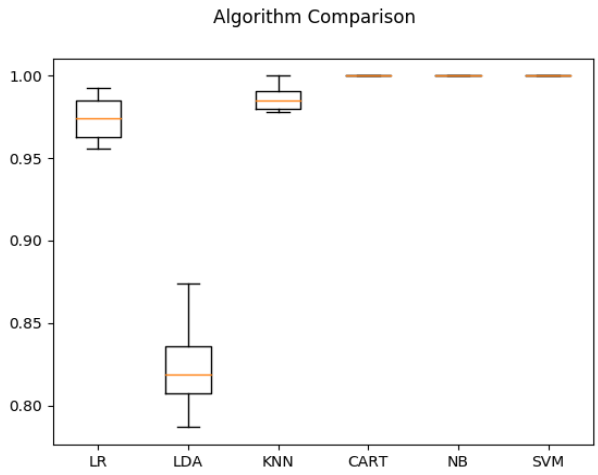
Figure 23: The mean accuracy and the standard deviation accuracy

```
race_algorithms   race_algorithms   race_algorithms
/usr/bin/python2.7 /home/student/PycharmProjects/phishing/Code_13/race_algorithms.py
LR: 0.974156 (0.013687)
/usr/lib/python2.7/dist-packages/sklearn/discriminant_analysis.py:389: UserWarning: Variables are collinear.
  warnings.warn("Variables are collinear.")
LDA: 0.821912 (0.024564)
KNN: 0.985953 (0.006968)
CART: 1.000000 (0.000000)
NB: 1.000000 (0.000000)
SVM: 1.000000 (0.000000)
```

K.Automating Machine Learning Workflows with Pipelines

"In machine learning, work processes occur to overcome common problems. Python scikit-learn provides a pipeline that is simply an abstraction rather than an algorithm that allows you to perform a sequence of different transformations (finding and generating functions, selecting specific features) from primary data before applying evaluators. It is possible to construct a chain of evaluators using a pipeline (Nenkov et al., 2016). With the pipeline, network search is allowed through a set of parameters for each step by placing transformers, evaluators and predictors for which there are corresponding installation rules set in the scikit-learn classes.(Guido and etc., 2016)"

Figure 24: It seems that Classification and Regression Trees, Naive Bayes and Support Vector Machines are more appropriate for the next study.



- Data Preparation and Modeling Pipeline

L. Saving and Loading Machine Learning Models

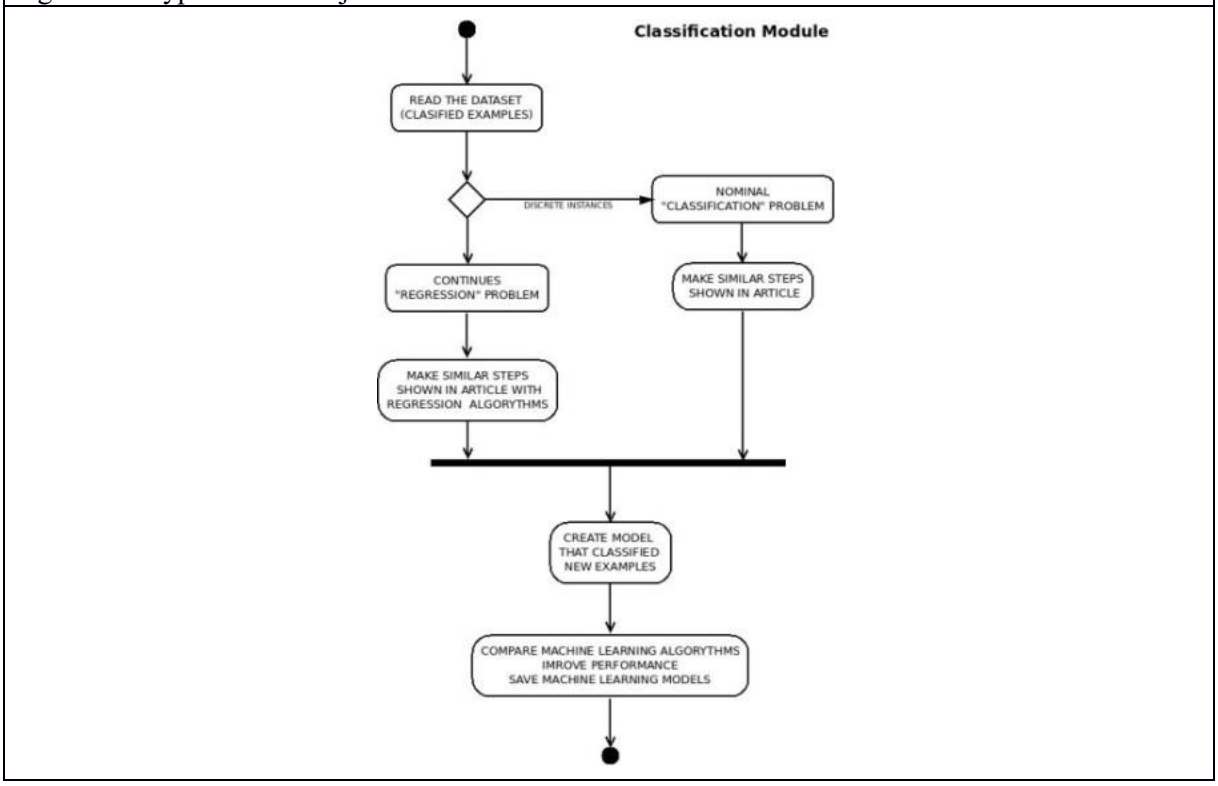
- To finalize our model with pickle "To save our model, we can use the Python pickle module that allows objects to be converted into a byte stream (serialization) and eventually transformed into a primary form if necessary (Brownlee, 2017)."
- To finalize our model with Joblib library from SciPi (Idris, 2016). We use Joblib to create Python objects that can be used in NumPy data structures multiple times.

Sample algorithm for writing a computer program to find and look for accurate models in Data Mining

If we need to write a general computer program to search for knowledge, it should have the options presented below and maybe more. Most likely, this would be done in time . In terms of volume, such as programming work and funding, this is a serious challenge. Perhaps a lighter option would be to create smaller modules that solve specific subtasks.

INPUT or Data Preparation.

Figure 26: Hypothetical Project “Classification module"



OUTPUT "Understand the Problem, Knowledge representation and reasoning (Tables, Linear models, Trees, Rules, Instances-Based, Clusters and others), Mining Frequent Patterns, Associations, and Correlations Feature Selection For Machine Learning (Hamilton, 2009), Classification and Prediction and others described in the literature.

A block diagram for a classification module on which a python computer program can be implemented is shown in Figure 26.

Conclusion

Research using the Python programming language and some of its libraries without using methods like Ensemble Enhanced Productivity Enhancement or Algorithm Tuning, shows that data sets can be analyzed and possibly create successful models for further use.

The main conclusion that can be made is that there is a need for a thorough knowledge of the Python programming language, libraries like SciPy, NumPy, Matplotlib, Pandas, scikit-learn and others which provide all of the machine learning algorithms, tools and algorithms to achieve more accurate results in Data Mining.

The Python language and its libraries are applicable in many fields such as intelligence technologies in management, and administration of Justice (Nenkov, 2015).

In this case, with the data processed and the receipt of the model, we can ask ourselves whether all the procedures performed could be automated, to make a general program that, according to the results, chooses relevant algorithms to compare them with the real data and produce a correct model. The answer to this question is related to many practical studies and is very difficult to answer because of the variety of existing methodologies to find the right model. Perhaps the future will show.

References

Brownlee, J. (2017). Machine Learning Mastery With Python Understand Your Data, Create Accurate Model sand Work Projects End-To-End, 123-145.

Guido S., Müller A. (2016, October), Introduction to Machine Learning with Python, A Guide for scikit-learn, 123-145.

Hamilton, H. (2009). Knowledge Discovery in Databases, Retrieved from http://www2.cs.uregina.ca/~dbd/cs831/index.html

HanJ., Kamber, M.(2006). Data Mining: Concepts and Techniques. Second Edition, 78-35

Idris I. (2016). Python Data Analysis Cookbook, ISBN-10: 178528228X, 123-456.

Nenkov, N., Tasinov, T., &Petrova, M. (2017). Software system for document management at the Faculty to University, 4th *International Multidisciplinary Scientific Conference on Social Sciences and Arts, SGEM 2017*. Conference Proceedings, VOL V Science and Society. Book 3, Education & Educational Research, Pages: 457-464, DOI:10.5593/SGEMSOCIAL 2017/35/S13.060, ISBN 978-619-7408-22-5, ISSN 2367-5659

Nenkov, N., Dyachenko, Y., Petrova, M., Bondarenko, G.,&Pustovit, V. (2017). Intelligent and Cognitive Technologies in Education of International Economic Relations Students and Human Resource Development in Enterprises: Methodology in Language. *European Journal of Sustainable Development*. Publisher: European Center of Sustainable Development, Rome, Italy, Vol 6D, No.4, pp.353-360,DOI: 10.14207/ejsd.2017.v6.n4.p353

Nenkov, N., Petrova, M.,&Dyachenko, Y. (2016). Intelligence Technologies in Management and Administration of Justice, *3rd International Multidisciplinary Scientific Conference on Social Sciences and Arts, SGEM 2016.* BK 2: POLITICAL SCIENCES, LAW, FINANCE, ECONOMICS AND TOURISM CONFERENCE PROCEEDINGS, VOL V Book Series: International Multidisciplinary Scientific Conferences on Social Sciences and Arts, DOI: 10.5593/SGEMSOCIAL2016/B25/S07.050, Pages: 385-392, WOS: 000395727200050

Nenkov, N. Petrova, M. (2015). Instruments and criteria for research and analysis of Internet visibility of Bulgarian judicial institutions WEB-space, International Journal of Advanced Research in Artificial Intelligence (IJARAI), ISSN: 2165-4050(Print), 2165-4069(Online), Volume 4 (9) 2015, (DOI): http://dx.doi.org/10.14569/IJARAI.2015.040902, pp. 6-9.

Schneider, J., Moore, A. (1997). A locally Weighted Learning tutorial using Vizier 1.0.

Seguí, L. (2017). Introduction to Data Science A Python Approach to Concepts, Techniques and Applications, 5-28.

Witten, I., Frank E., Hall, M.,& Pal, Ch. (2017).Data Mining Practical Machine Learning Tools and Techniques Fourth Edition, 45-74.

UCI Machine Learning Repository. (2017). Retrieved from http://archive.ics.uci.edu/ml/index.php