University of Essex

Department of Mathematical Sciences

Part 1: Pilot Study

Submitted as part of the requirements for:

CE802 Machine Learning and Data Mining

Name: Divya Arora

Registration number:1901423

Supervisor: Luca Citi

Date of submission (January 15 2020)
Word count:719

Abstract: *This pilot study is to investigate the feasibility of machine learning procedures in predicting the fake news and misleading information affecting the image of social media company.*

Keywords:*social media,false information,problematic accounts,machine learning*

Introduction

A social media company needs to find out the false and misleading information affecting the image of the company. In consideration of bad media reputation, the manager decides to use data analytics to identify problematic accounts that are posting false information. Furthermore, historical data for all accounts are available to do the analysis.

1 Main Body

Four main points discussed below are predictive task, selective informative features, learning procedures and evaluation.

1.1 *Type of Predictive Task*

First, we need to classify the machine learning paradigm. As it has been analyzed that labeled data is given which indicates that it is a supervised learning algorithm. The accounts are divided into posting false information or not. Supervised learning algorithms divided into further two categories classification and regression. When it comes to the difference between classification and regression, classification technique is used when the output variable is categorical consists of two classes such as yes-no, true-false, male-female while regression is used for predicting continuous variables. Therefore, Classification is the type of predictive task for identifying problematic accounts posting false information as the output data is divided into categories (true or false)[1].

1.2 *Selective Informative Features*

The research illustrates that source of information, content of information and user responses are three primary characteristics for detecting any false information found on social media [3]. It is very easy, convenient and inexpensive nowadays to create bots and register new accounts. It is very necessary to classify the information content whether it is true or fake. Moreover,

according to [2], review related features and user-related features should be considered by taking into account review-id, review-date, review-text, star rating and user-name, user-id, number of friends, number of reviews they have written, how long they have been associated with social media company. False information has a larger impact on social media because of the sharing content at a large scale. User response are more useful than the information content because they are harder to manipulate as well as user engagements like shares, replies or comments contain path of the information flow.

1.3 *Learning Procedures*

There are various algorithms in machine learning, the most important task is to find out the best among them used to solve the specific problem. The selection of the algorithm based on the domain and subject of the problem being analyzed. There is not any generalized algorithm available to deal with all sorts of problems. So, the solution is to try different algorithms and choose the best among them.[4] Machine Learning Classification algorithm can be divided into two types linear and non-linear models. One widely used non-linear classification algorithm is a decision tree, it is a supervised learning technique that can be used both for classification and Regression problems, but mostly it is preferred for solving Classification problems [5].So, decision tree is preferred as it follows the same process as humans follow while making any decision and logic behind decision tree can be understood easily because of its tree-like structure. Naive Bayes is an algorithm that can be used for dealing with binary classification problems[6]. Furthermore, another helpful linear classification algorithm is logistic regression which is used for predicting the categorical dependent variable using a given set of independent variables. Its outcome variable could be categorical or discrete value and the account in this case consists of two values true and false [7]. SVM is used because of its ability to best divide classes into two parts so that it is easy to align new observation in the correct category. Training and Testing of models are done on these three algorithms[8].

1.4 *Evaluation approach*

The performance of the algorithms can be evaluated based on confusion matrix, accuracy, precision and recall. It is necessary to evaluate the performance of the model after carrying out predictions on the test set by undergoing a division of data set into seventy percent training and thirty percent testing. Confusion matrix is used to find out number of correct and incorrect predictions. Accuracy is one of the most important parameter for evaluating the performance of the model. However, only accuracy is not enough there are other parameters also precision and recall for evaluating the performance of the model. Precision is the proportion of correct Predictions while Recall is the ability of classifier to find all positive instances or relevant cases in the data set[9].

1.5 *References*

[1]https://www.javatpoint.com/supervised-machine-learning

[2] https://csce.ucmss.com/cr/books/2018/LFS/CSREA2018/IKE4299.pdf

[3] https://arxiv.org/pdf/1901.06437.pdf

[4] https://www.guru99.com/machine-learning-tutorial.html

[5] https://www.javatpoint.com/logistic-regression-in-machine-learning

[6] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[7] https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[8] https://www.javatpoint.com/machine-learning-naive-bayes-classifier

[9] https://www.javatpoint.com/data-preprocessing-machine-learning

University of Essex

Department of Mathematical Sciences

Part 2: Comparative Study

Submitted as part of the requirements for:

CE802 Machine Learning and Data Mining

Name: Divya Arora

Registration number:1901423

Supervisor: Luca Citi

Date of submission (January 15 2020)

Word count:855

Introduction

Initially,there should a clear understanding of the problem and datain order to produce an effective results.This Comparative study is used to investigate which machine learning algorithm is best among four for solving the problem.

2 Main Body

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	...	F12	F13	F14	F15	F16	F17	F18	F19	F20	Class
0	0	0	16	2.02	0.52	-2.35	-1.98	-0.70	85	6	...	-0.07	1.08	15	-0.63	-3.49	-1.68	0.02	15.3	0.074146	True
1	0	0	86	-0.90	2.75	0.14	0.83	-0.06	107	1	...	0.17	1.06	-8	-1.21	0.34	0.36	0.61	10.1	0.074146	True
2	1	1	165	0.73	1.05	0.10	2.57	-1.65	41	5	...	0.04	0.42	-6	-0.46	-0.62	1.67	2.60	11.0	1.550000	False
3	1	1	191	-1.50	0.79	0.33	1.24	1.35	17	2	...	1.74	1.74	15	0.47	0.63	0.08	0.19	6.3	0.950000	False
4	1	1	13	0.25	-1.19	-0.90	2.67	0.22	12	8	...	-0.39	1.25	25	-0.09	-2.41	-0.53	-0.77	10.5	0.074146	True

5 rows × 21 columns

Figure 1: Imputation by mean

Data Pre Processing is one of the significant step while using machine learning algorithms for training purposes.It involves importing dataset,splitting the dataset into training and test set,deal with the missing values.Missing values is present in our data in 20th column only with a large number of NAN values. It is one the important step to handle the missing values before going further with training the model. There are three ways to handle missing values. The most common approach to deal with missing value is to either do the deletion or imputation using mean or median. Mean imputation has been carried out here to deal with the missing value which is one of the most justified approaches.

2.1 *Decision Tree with pruning*

Decision tree is a classifier where predictions are carried out by doing recursive splits in a smaller number of iterations. It is a hierarchical tree structure where internal nodes represent feature, branch represent decision rule and leaf represents outcome. There is no need of large computation in this algorithm. It is easy to understand and mainly used for solving decision related problems. It is often the case when the tree becomes large and complex,it is necessary to prun it for training the data. Gini Index and Entropy are the two criterions for using the decision tree algorithm but the most used among the two is Entropy. Based on the value of information gain, splitting of nodes carried out.[1][2]

Few steps are carried out for training the data with decision tree classifier.

1: Create a decision tree classifier object for both the criterion.

2: Fit the objects into the training set.

3: Prediction on test set by using both the criterion.

4: Evaluating test set accuracy for both the criterion.

5: There is not much difference in both the criterion but somehow entropy is still leading.So ,it is preferred to move on to other hyperparameter which are fixed inside the model.

Confusion matrix of Entropy		
	Actual Positive	Actual Negative
Predicted Positive	70	11
Predicted Negative	53	16

Confusion matrix of Gini Index		
	Actual Positive	Actual Negative
Predicted Positive	70	11
Predicted Negative	55	14

Evaluating the Performance		
Parameters	Entropy	Gini Index
Total No. of Correct Predictions	86	84
Total No. of Incorrect Predictions	64	66
Accuracy	57.333333333333336	56.00000000000001
Precision	59.25925925925925	56.00000000000001
Recall	23.18840579710145	20.28985507246377

6: Cross-validation has been carried out with the grid search approach for carrying out with the process of pruning.[3][4]

a: Defining grid of hyperparameters

b: Fitting grid to the training set

c: Calculate Best CV accuracy, Best Hyper Parameter, Best Estimator and its accuracy.

Best Hyper Parameters	
	Outcome
Max Depth	3
Min Impurity Decrease	0.01
Min Samples Leaf	5
Min Samples Split	2

Grid Search with CV	
	Outcome
Best CV accuracy	59.14285714285714
Accuracy of best estimator	57.333333333333336

2.2 *Logistic Regression*

Logistic Regression is one of the widely used algorithm used for predicting the outcome variable which must contain categorical values.Logistic Regression and Linear Regression are very common but with the fact that Linear regression is not used for classification problems but instead used in the regression. There are three approaches in logistic regression, but we strictly go for binomial one with the advent of having only two possible values.[4][5]

For using logistic regression classifier,need to follow some steps:

1:import logistic regression from sklearn linear model library.

2:Instantiate the model with a linear regression classifier.

3:Fit the classifier with training data.

4:Do the predictions on the testing data and illustrate the performance of the model.

The Table below shows the outcomes of the evaluation.

Confusion matrix		
	Actual Positive	Actual Negative
Predicted Positive	63	18
Predicted Negative	37	32

Evaluating the Performance	
Parameters	Outcome
Total No. of Correct Predictions	95
Total No. of Incorrect Predictions	55
Accuracy	63.33333333333333
Precision	64.0
Recall	46.3768115942029

2.3 *Support Vector Machines*

SVM is one of the widely used algorithm used for classification related problems. It is known for its ability to separate the classes efficiently. There are two types of support vector machine linear and non-linear. As per the consideration of the situation, linear SVM should be used as data has been classified into two classes only.[9]

For using this classifier, need to follow the given steps:

1: Import support vector classifier by using linear Kernel.

2: Fit the classifier to the training set.

3: predicting test set result.

4: Compare actual and predicted response values.

The Table below shows the performance of the model.

Confusion matrix		
	Actual Positive	Actual Negative
Predicted Positive	68	13
Predicted Negative	39	30

Evaluating the Performance	
Parameters	Outcome
Total No. of Correct Predictions	108
Total No. of Incorrect Predictions	52
Accuracy	65.33333333333333
Precision	69.76744186046511
Recall	43.47826086956522

2.4 *Naïve Bayes Algorithm*

Naïve Bayes is the Proficient Classifier used in machine learning with a base concept based on Bayes' Theorem. It is one of the most undemanding and quick algorithms for the prediction purpose on account of which it does not need much training data. It is capable of handling both continuous and discrete values. It is ok with Gaussian, Multinomial and Bernoulli are the three possible options in naïve Bayes algorithm. As per the data, Gaussian approach should be used since each feature in the dataset are continuous and normally distributed.[7][8] The model has been built by carrying out with the following steps:

1: Import Gaussian Naïve Bayes from Sklearn Naïve Bayes library.

2: Instantiate the model with the Gaussian approach.

3: Fit the Gaussian Naïve Bayes classifier with the training set.

4: Make predictions on the testing set and evaluate the performance of the model.

Table shows Confusion Matrix and evaluated performance of the model.

Confusion matrix		
	Actual Positive	Actual Negative
Predicted Positive	62	19
Predicted Negative	47	22

Evaluating the Performance	
Parameters	Outcome
Total No. of Correct Predictions	84
Total No. of Incorrect Predictions	66
Accuracy	56.00000000000001
Precision	53.65853658536586
Recall	31.88405797101449

2.5 *Conclusion*

After Comparing all the three algorithm parameters, it has been concluded that Support Vector is leading in two out of three parameters which are precision and accuracy.However,Recall value is more for Logistic Regression.After analysing the outcome ,it has been seen that Support vector is giving more correct predictions as compared to Logistic Regression. Hence,it is proved that Support Vector Machine is the best algorithm among all the four algorithms. The CSV file has been gernated based on support vector classifier.

Comparison			
Classifier	Accuracy	Precision	Recall
Pruned decision tree	57.333333333333336	59.25925925925925	23.18840579710145
Logistic Regression	63.33333333333333	64.0	46.3768115942029
Support Vector Machines	65.33333333333333	69.76744186046511	43.47826086956522
Naïve Bayes	56.00000000000001	53.65853658536586	31.88405797101449

2.6 *References*

[1] https://www.ritchieng.com/machine-learning-decision-trees/

[2] https://www.ritchieng.com/machine-learning-decision-trees/

[3] https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[4] https://dataaspirant.com/2017/02/01/decision-tree-algorithm-python-with-scikit-learn/

[5] https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python

[6] https://www.javatpoint.com/logistic-regression-in-machine-learning

[7] https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/

[8] https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html

[9] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

Appendix

#Import Packages

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split #train_test_split helps split data into train & test set

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score

from sklearn import tree

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

from sklearn.metrics import precision_score, recall_score

from sklearn.model_selection import GridSearchCV #Import GridSearchCV

import warnings

warnings.filterwarnings("ignore")
```

#Import training dataset

```
balance_data = pd.read_csv(

'C:\\Users\\HP\\Desktop\\coursework ml\\CE802_Ass_2019_Data.csv',sep= ',')

#import test dataset

pred_data=pd.read_csv(

'C:\\Users\\HP\\Desktop\\coursework ml\\CE802_Ass_2019_Test.csv',sep= ',')
```

#Comment Section

```
“””#import dataset by removing 20th column with nan values

#cols = list(pd.read_csv('C:\\Users\\HP\\Desktop\\coursework ml\\CE802_Ass_2019_Data.csv', nrows =1))

#balance_data = pd.read_csv(

#'C:\\Users\\HP\\Desktop\\coursework ml\\CE802_Ass_2019_Data.csv',usecols =[i for i in cols if i != 'F20'],sep= ',')
```

```
#pred_data=pd.read_csv(
#'C:\\Users\\HP\\Desktop\\coursework ml\\CE802_Ass_2019_Test.csv',usecols =[i for i in cols if i != 'F20'],sep= ',')"""
```

#missing values in training dataset

```
balance_data['F20'].fillna(balance_data['F20'].mean())
balance_data=balance_data.fillna(balance_data.mean())
```

#missing values in test dataset

```
pred_data['F20'].fillna(pred_data['F20'].mean())
pred_data=pred_data.fillna(pred_data.mean())
```

#dataset shape

```
print("Length: ", len(balance_data))                    #Length: 500
print("Dataset Shape: ", balance_data.shape)            # Dataset Shape:(500, 21)
```

#dataset observation

```
balance_data.head()
```

#data slicing

```
X = balance_data.values[:,:-1]
Y = balance_data.values[:,20]
```

#cutomized code to convert response variable in 0 and 1 for the use in algorithm

```
for i in range (Y.size):
   if Y[i]:
      Y[i] = 1
   else:
      Y[i] = 0
```

```
Y=Y.astype(int)
X_pred=pred_data.values[:,0:20]

#split the dataset into 70% training and 30% testing set
X_train, X_validate, y_train, y_test = train_test_split( X, Y, test_size = 0.3, random_state = 100)

######Decision tree pruning #########
#Creating decision tree classifier object with gini index
clf_gini = DecisionTreeClassifier(criterion = "gini", random_state = 0,
                                  max_depth=3, min_samples_leaf=5)
#Perform Training
clf_gini.fit(X_train, y_train)

#Pridiction on test set with gini index
y_pred = clf_gini.predict(X_validate)
#print(y_pred)

#Calculating Test set Accuracy with gini index
print("Confusion Matrix:\n ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Precision:",metrics.precision_score(y_test, y_pred)*100)
print("Recall:",metrics.recall_score(y_test, y_pred)*100)
print("Report : \n",classification_report(y_test, y_pred))

#Creating decision tree classifier object with information gain
clf_entropy = DecisionTreeClassifier(criterion = "entropy", random_state = 100,
 max_depth=3, min_samples_leaf=5)
```

```
#Perform Training
clf_entropy.fit(X_train, y_train)

#Pridiction on test set with information gain
y_pred_en = clf_entropy.predict(X_validate)
#print(y_pred_en)

#Calculating Test set Accuracy with information gain
print("Confusion Matrix: \n ",
    confusion_matrix(y_test,y_pred_en))
print("Accuracy: ", accuracy_score(y_test,y_pred_en)*100)
print("Precision:",metrics.precision_score(y_test, y_pred_en)*100)
print("Recall:",metrics.recall_score(y_test, y_pred_en)*100)
print("Report :\n ",classification_report(y_test,y_pred_en))

#Results of both criteria are nearly same
#Difficult to choose which is better
from sklearn.metrics import precision_score, recall_score
#print hyperparameters
print(clf_gini.get_params())


#pruning with Grid search CV
#Defining grid of hyperparameters
parameter_grid = {
'max_depth': [1,3,5,7,9,20],
'min_samples_split': [2,3, 4, 5],
'min_impurity_decrease': [0.01 , 0.012,0.004 , 0.008],
'min_samples_leaf':[2,5,6,8,10]
```

```
}

# Instantiate a 10-fold CV grid search object 'grid'
grid = GridSearchCV(estimator=clf_gini,
                    param_grid=parameter_grid,
                    scoring='accuracy',
                    cv=10)
# Fit grid to the training set.
grid.fit(X_train, y_train)

# Best CV score across all parameters from 'grid'
print('Best CV accuracy: ',format(grid.best_score_*100))

#Best hyperparameters
print('Best hyerparameters: ', grid.best_params_ )

#Best Estimator
best_model = grid.best_estimator_
# Calculate accuracy
print("Accuracy of best model ",format(best_model.score(X_validate,y_test)*100))
#Accuracy is same as entropy criterion.So,precision and recall is same as information gain
```

#Logistic Regression Classification

```
#import required libraries
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
from sklearn import metrics # import the metrics class
```

```
# instantiate the model
logreg = LogisticRegression()
# fit the model with training data
logreg.fit(X_train,y_train)
#predicting with test set
y_pred_reg=logreg.predict(X_validate)

# comparing actual response values (y_test) with predicted response values (y_pred)
print("Confusion Matrix: \n ",confusion_matrix(y_test,y_pred_reg))
print("Accuracy:",metrics.accuracy_score(y_test, y_pred_reg)*100)
print("Precision:",metrics.precision_score(y_test, y_pred_reg)*100)
print("Recall:",metrics.recall_score(y_test, y_pred_reg)*100)
```

##Support Vector Machines

```
from sklearn.svm import SVC # "Support Vector Classifier"
supp_vector = SVC(kernel='linear')
# instantiate the model (using the default parameters)
# fit the model with data
supp_vector.fit(X_train,y_train)
#predicting with test set
y_pred_svc=supp_vector.predict(X_validate)

# comparing actual response values (y_test) with predicted response values (y_pred)
print("Confusion Matrix: \n ",confusion_matrix(y_test,y_pred_svc))
print("Accuracy:",metrics.accuracy_score(y_test, y_pred_svc)*100)
print("Precision:",metrics.precision_score(y_test, y_pred_svc)*100)
print("Recall:",metrics.recall_score(y_test, y_pred_svc)*100)
```

#Naive Bayes Classification

```
# training the model on training set

#import library
from sklearn.naive_bayes import GaussianNB
#This library already included in the beginning
from sklearn import metrics

#instantiate the model
gnb = GaussianNB()

#fit the model with data
gnb.fit(X_train,y_train)

# making predictions on the testing set
y_pred_bayes = gnb.predict(X_validate)

# comparing actual response values (y_test) with predicted response values (y_pred)
print("Confusion Matrix: \n ",confusion_matrix(y_test,y_pred_bayes))
print("Accuracy:",metrics.accuracy_score(y_test, y_pred_bayes)*100)
print("Precision:",metrics.precision_score(y_test, y_pred_bayes)*100)
print("Recall:",metrics.recall_score(y_test, y_pred_bayes)*100)
```

#Support Vector is the best algorithm among four.

```
best_y_pred=supp_vector.predict(X_pred)
```

#Converting class integer values into boolean values

```
a=0
b=0
```

```
for i in best_y_pred:
    if i==0:
        best_y_pred[i] = True
        a=a+1
    else:
        best_y_pred[i] = False
        b=b+1
best_y_pred = best_y_pred.astype(bool)

#Printing total number of '0' and '1' in predicted outcome
print("0: ",a)                  # 0:359
print("1: ",b)                  #1 : 141

#fill the predicted class in test dataset
pred_data['Class'] = best_y_pred
pred_data['Class'] = pred_data['Class']

#Form csv file of predicted results
pred_data.to_csv('C:\\Users\\HP\\Desktop\\coursework ml\\New
folder\\New_Version_CE802_Ass_2019_Test.csv')
```

"""Coding help taken from these websites

Decision tree:

https://www.geeksforgeeks.org/decision-tree-implementation-python/

https://www.ritchieng.com/machine-learning-decision-trees/

Logistic Regression:

https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python

Support Vector:

https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/

Naive Bayes:

https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn

Missing value:

https://jamesrledoux.com/code/imputation

CSV:

https://stackoverflow.com/questions/34864695/saving-prediction-results-to-csv ***"""***