Computer Engineering Department

CMPE-255 | Data Mining | Professor David C. Anastasiu

Final Project Report

DECEIT DETECTION

Group 13

Adithya Baskaran (013006344)

Priyanka Kumar (013749866)

Smitha Eshwarahalli Ramesh (013714545)

Spring 2019

Table of Contents

Chapter 1. Introduction

Deceits has been ruling the world from the beginning of ages. But with the right technology, we can reduce its effect on our day to day life directly or indirectly. Lying requires the deceiver to keep facts straight, make the story believable, and withstand scrutiny. When individuals tell the truth, they often make every effort to ensure that other people understand. In contrast, liars attempt to manage other's perceptions. Consequently, people unwittingly signal deception via nonverbal and verbal cues.

In this project, we will be considering facial expressions (eye, nose, eyebrow, mouth) and hand gestures for our analysis. When we are stressed, bothered, disappointed, disturbed, anxious, worried, concerned, uncertain, exasperated, or mad, our bodies reveal the required information nonverbally by any number of expressions throughout the body. For example, eyes serve as a blocking mechanism, much the same way as folded hands across the chest or turning away in disagreement. The irregular breathing pattern tells us about the liar's increased anxiety level.

The objective of this project is to build a fairly accurate model to streamline the way we check violent behaviours. This can, for instance reduce the smuggling, harmful individuals, interrogations and many unseen applications and if possible, even be accepted as evidence in courts to speed up the hearings in process.

Chapter 2. System Design & Implementation details

	Module 1 (Facial Expression Dataset)	**Module 2 (Postures Dataset)**
Algorithms used	• Label Propagation (***Performed Best***) • Label Spreading	• Random Forest (***Performed Best***) • Decision Trees • Logistic Regression • MultiLayer Perceptron
Tools and techniques used	1) Google Colab 2) Packages used: Pandas, numpy,matplotlib, Scikit-Learn	1) Google Colab 2) Packages used: Pandas, numpy,seaborn,matplotlib, Scikit-Learn

Algorithms considered:

The algorithms analysed and compared to detect whether the person is being truthful or deceitful from the data sets procured from the online data banks on hand gestures and facial expression are Random forest, Decision tree, Logistic regression, and MLP for hand gestures and label propagation, and label spreading for facial expression.

Model Flow Diagram

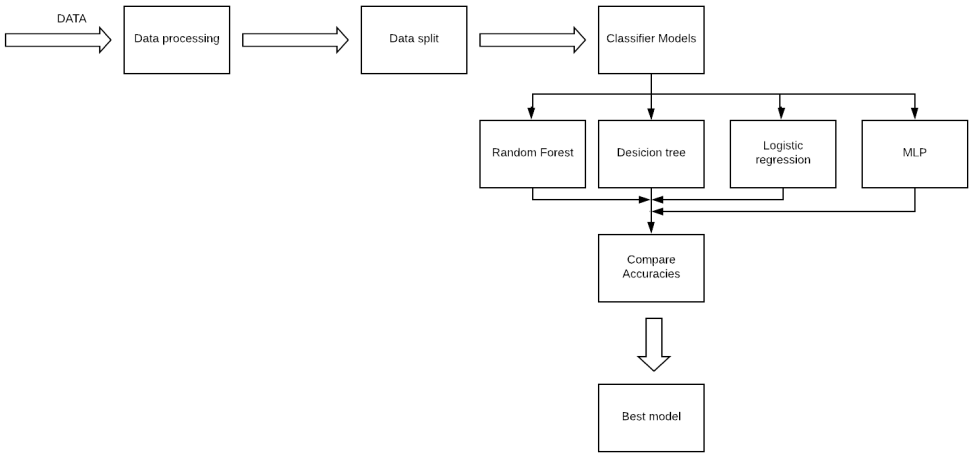


Figure 2.5 Architecture presented for hand gesture module

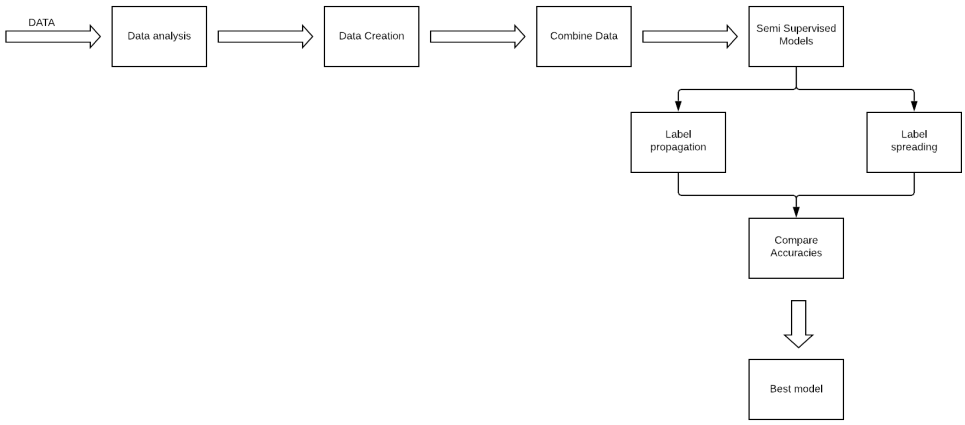


Figure 2.6 Architecture presented for All_gesture module

Chapter 3. Experiments and Proof of Concept Evaluation

Datasets:

The data set presented are the EEG record of the physical , facial expression (**Figure 3.4**) ,self generated (**Figure 3.3**) to correlate with the the fact that person is being truthful or deceitful. These datasets are from different years and libraries.These are processed separately and analysed individually but the data presented with facial expression is not adequate hence the label propagation and label spreading is taken up to analyse the data.The details of the dataset used are presented in **Table 1** along with source and Attributes.

Sources	Dataset	Attributes	Instances
UCI Machine learning Repository	Motion Capture Hand Postures (Figure 3.6)	38	78095
Kaggle	All_Gestures_Deceptive and Truthful (Figure 3.7)	41	122
Self-Generated	data_ret_re	39	1000000

Table 3.1 Details about Dataset

	OtherGestures	Smile	Laugh	Scowl	otherEyebrowMovement	Frown	Raise	OtherEyeMovements	Close-R	X-Open	...	waggle	forwardHead	downRHead
0	0	0	1	1	1	0	1	0	0	0	...	0	0	1
1	0	0	0	0	0	0	0	0	1	1	...	0	0	0
2	0	1	0	1	0	1	0	0	0	1	...	0	0	0
3	1	0	1	0	1	0	1	0	1	1	...	0	0	0
4	0	0	1	1	1	0	0	1	1	1	...	1	1	0

5 rows × 39 columns

Figure 3.3 Outlook of the created Data set

	id	OtherGestures	Smile	Laugh	Scowl	otherEyebrowMovement	Frown	Raise	OtherEyeMovements	Close-R	...	forwardHead
0	trial_lie_001.mp4	1	0	0	0	1	0	0	1	0	...	0
1	trial_lie_002.mp4	1	0	0	0	0	1	0	1	0	...	0
2	trial_lie_003.mp4	1	0	0	0	0	1	0	0	1	...	0
3	trial_lie_004.mp4	1	0	0	0	1	0	0	1	0	...	0
4	trial_lie_005.mp4	1	0	0	0	0	1	0	1	0	...	0

5 rows × 41 columns

Figure 3.4 Outlook of All_Gestures_Decptive_and_truthful Dataset

Data Exploration:

The data present do not change over time and the values are presented after the analysis of the images and are processed with respect to binary values. The processed values are either present or not this way the value variation are limited “YES” or “NO”(**Figure 3.5**).The data present has no form of noise which requires any specific attention or algorithms.

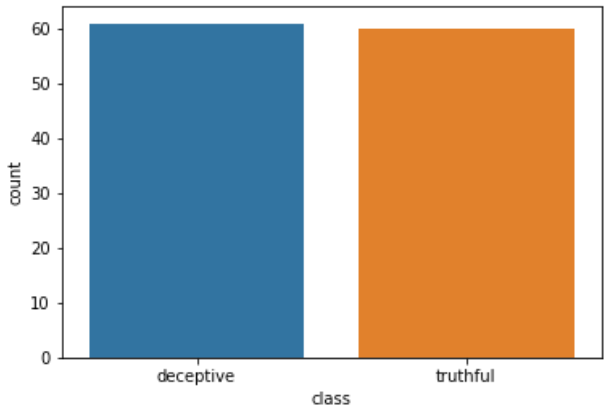


Figure 3.5 Distribution of class Labels in Facial Expression Dataset

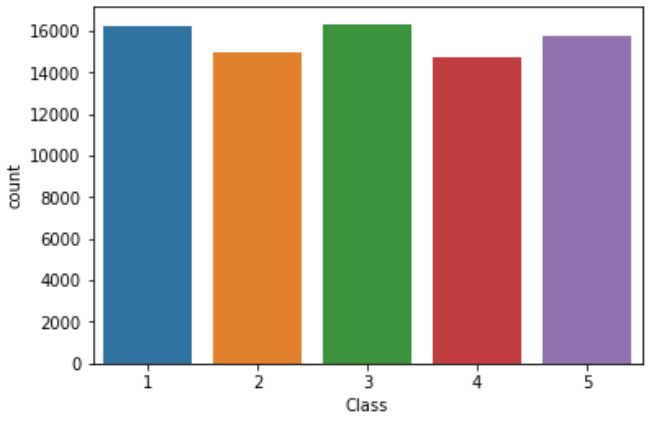


Figure 3.6 Distribution of class Labels in Postures Dataset

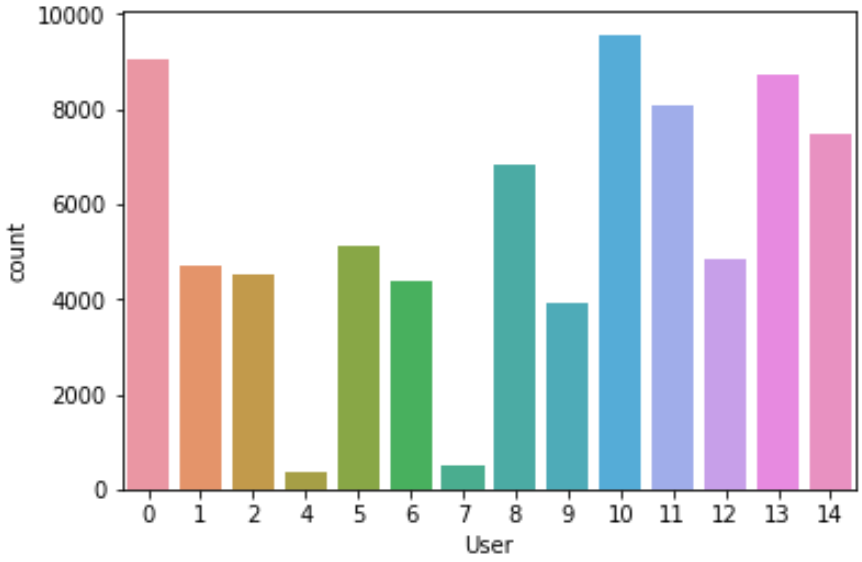


Figure 3.7 Distribution of Users in Postures Dataset

DATA PROCESSING:

The data was first processed by recognising all the "?" and NaN value are replaced with "0" after which the unwanted columns such as Id ,class are dropped. The class labels were transformed from deceptive to 1 and truthful to 0. This is done because of the idea we are following through, is to find deceptive as opposed to whether the person is truthful. This might sound like the same thing but when thinking in line of True positives and False positives, it matters a lot.

The fact that the true positive means that the accused for lying is actually deceitful but on the other hand if there is a false positive this means that the accused is actually truthful and is being wrongfully accused, Which hold more weightage than the getting the wrongful.

"A thousand sinners can be let go rather than wrongfully punishing a single innocent"
- Unknown

Thus, Deceptive is being given higher weightage in binary value than truth.

Results

Performance

Among the presented algorithms , Random forest worked the best, which is followed by random forest very closely .Where as Logistic regression does relatively well and finishing last would be MLP for Motion capture Hand posture.

In All_Gestures_Deceptive and Truthful both the models gave a similar value.

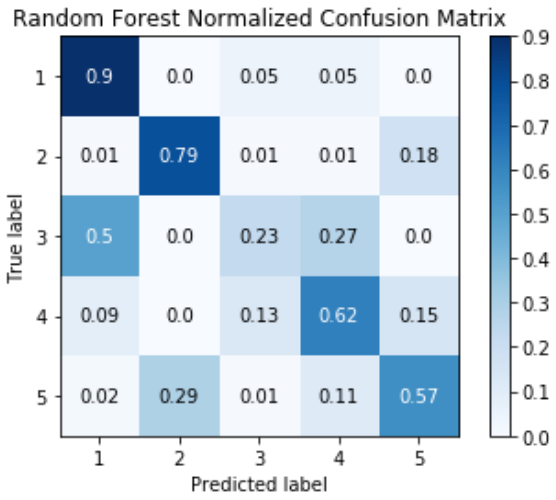
	MLP	Random Forest	Decision tree	Logistic Regression
Motion Capture Hand Postures	0.5324284525257	0.987515205839	0.9648505025929	0.7994109738139445

Table 3.2 Results for Hand posture module

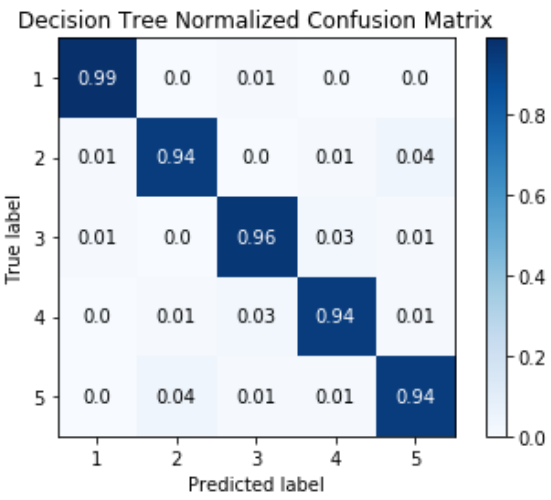
	Label propagation	Label spreading
Self-Generated	0.8	0.68

Table 3.3 Results for All_gesture module

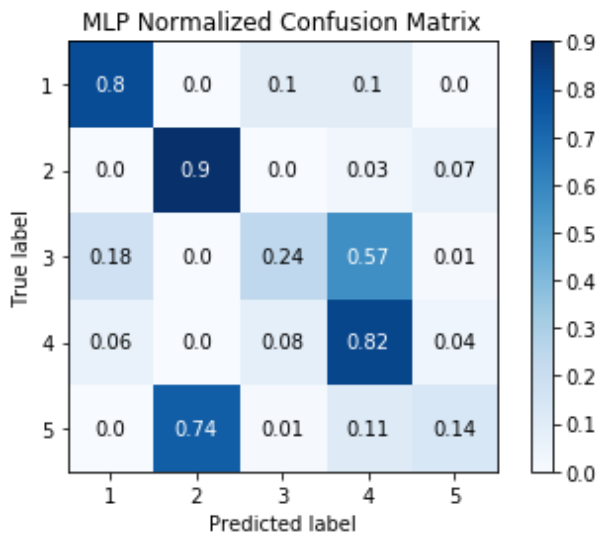
Random Forest:



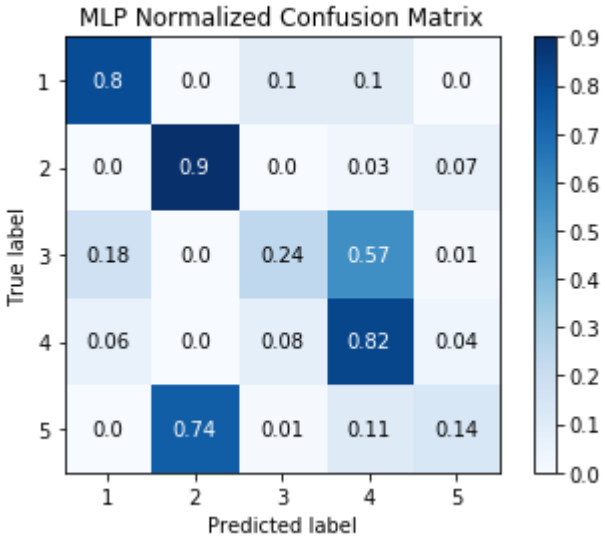
Decision Tree:



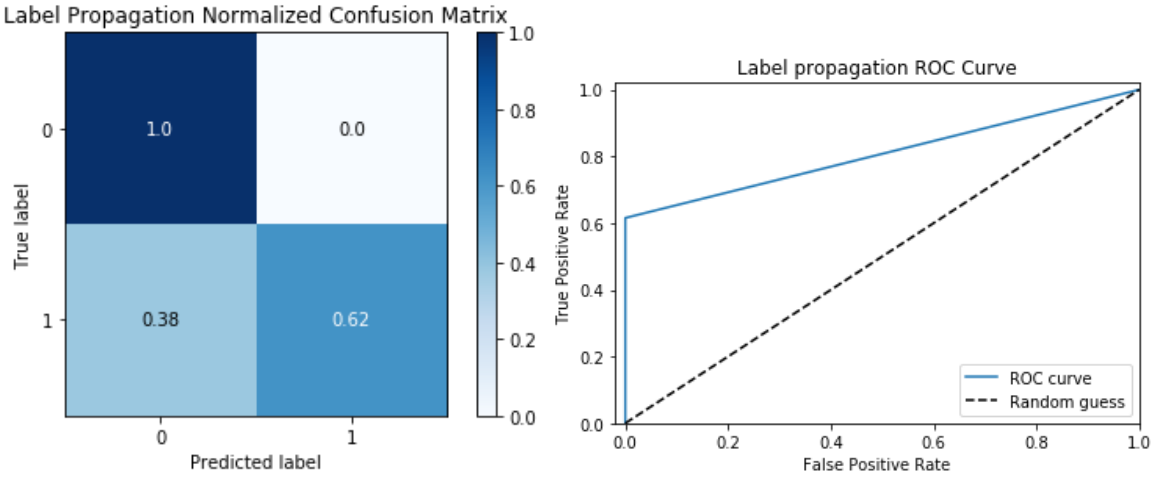
Logistic Regression:



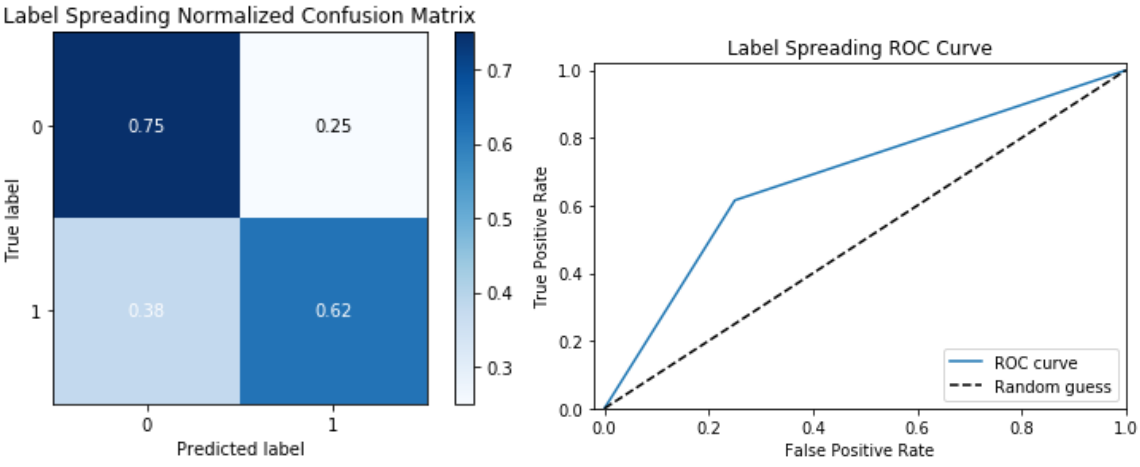
Multi-layer perceptron:



Label Propagation:



Label Spreading:



Chapter 4. Discussion and Conclusions

Decisions, Difficulties, and Discussion

Module 1:

Did not have enough data points to train the model. Had to create data points. This was easy because all the attributes in the dataset were binary. But the data set created had no labels and hence we decided to go with semi-supervised learning.

Module 2:

The dataset for this model was pretty clean and pre-processed. Dealing with this dataset posed no difficulties

Conclusion and Future Work

Hand gestures and facial expression are two major areas that have to be accounted for detecting lies. Most of the algorithms we experimented with, gave good results.

Future work is to combine hand gestures and facial expression data points and carry out the analysis and compare with the analysis done separately.

Chapter 5. Project Plan / Task Distribution

Task owner	Module 1	Module 2
Adithya Baskaran	1) Writing Script to create Facial Expression dataset 2) Implement one machine learning algorithm for facial expression dataset 3) Fine tuning hyperparameters Label propagation 4) Evaluating results	1) Implement one machine algorithm for postures Dataset 2) Fine tuning hyperparameters of Logistic Regression 3) Evaluating results

Priyanka Kumar	1) Data Exploration of Facial Expression dataset 2) Implement one machine learning algorithm for facial expression dataset 3) Evaluating results	1) Preprocess Postures Dataset 2) Implement two machine algorithms for postures Dataset 3) Fine tuning hyperparameters of Random Forest and MLP 4) Evaluating results 5) Comparing performances of four algorithms
Smitha Eshwarahalli Ramesh	1) Preprocess Facial Expression Dataset 2) Fine tuning hyperparameters Label Spreading 3) Creating a user Interface for the models 4) Comparing performances of two algorithms	1) Data Exploration of Postures dataset 2) Implement one machine algorithm for postures Dataset 3) Fine tuning hyperparameters of Decision Trees 4) Evaluating results

References

[1]https://en.wikipedia.org/wiki/Support_vector_machine

[2]https://en.wikipedia.org/wiki/Random_forest

[3]https://en.wikipedia.org/wiki/Decision_tree

[4]https://en.wikipedia.org/wiki/Logistic_regression

[5]https://en.wikipedia.org/wiki/Label_Propagation_Algorithm

[6]https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelSpreading.html

[7]http://web.eecs.umich.edu/~mihalcea/downloads.html#RealLifeDeception

[8]https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures

Appendix:

```
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ ls
All_Gestures_Deceptive_and_Truthful.csv  create_data.py  facial_exp_deceit_detector.py  hand_gestures_deceit_detector.py  Postures.csv  README.md
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ python create_data.py -o training_data.csv -n 100000
training_data.csv successfully created
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ ls
All_Gestures_Deceptive_and_Truthful.csv  create_data.py  facial_exp_deceit_detector.py  hand_gestures_deceit_detector.py  Postures.csv  README.md  training_data.csv
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$
```

Figure 2.1

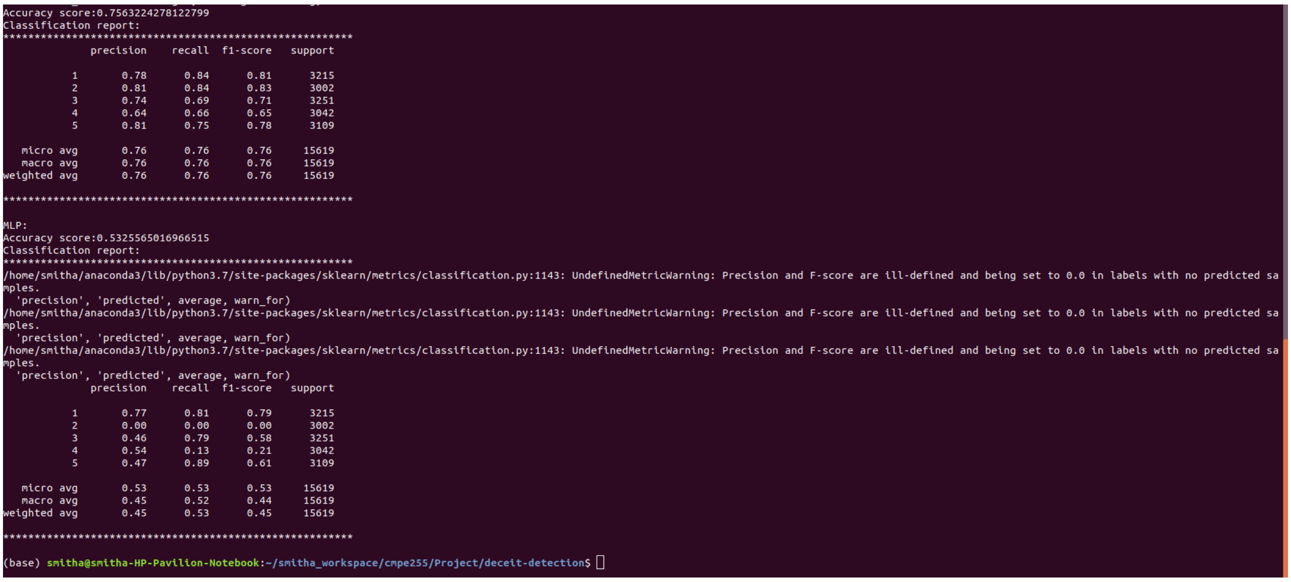
```
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ ls
All_Gestures_Deceptive_and_Truthful.csv  create_data.py  facial_exp_deceit_detector.py  hand_gestures_deceit_detector.py  hg_prediction.dat  Postures.csv  README.md  training_data.csv
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ python facial_exp_deceit_detector.py -i training_data.csv -t All_Gestures_Deceptive_and_Truthful.csv -o fe_pr
ediction.dat
LABEL PROPAGATION:
Accuracy score:0.8
Classification report:
**************************************************
              precision    recall  f1-score   support

           0       0.71      1.00      0.83        12
           1       1.00      0.62      0.76        13

   micro avg       0.80      0.80      0.80        25
   macro avg       0.85      0.81      0.79        25
weighted avg       0.86      0.80      0.79        25

**************************************************
LABEL SPREADING:
Accuracy score:0.72
Classification report:
**************************************************
              precision    recall  f1-score   support

           0       0.69      0.75      0.72        12
           1       0.75      0.69      0.72        13

   micro avg       0.72      0.72      0.72        25
   macro avg       0.72      0.72      0.72        25
weighted avg       0.72      0.72      0.72        25

**************************************************
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ ls
All_Gestures_Deceptive_and_Truthful.csv  create_data.py  facial_exp_deceit_detector.py  fe_prediction.dat  hand_gestures_deceit_detector.py  hg_prediction.dat  Postures.csv  README.md  training_data.csv
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$
```

Figure

Figure 2.2

```
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ ls
All_Gestures_Deceptive_and_Truthful.csv  create_data.py  facial_exp_deceit_detector.py  hand_gestures_deceit_detector.py  Postures.csv  README.md  training_data.csv
(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$ python hand_gestures_deceit_detector.py -i Postures.csv -o hg_prediction.dat
RANDOM FOREST:
Accuracy score:0.9861066649593444
Classification report:
**************************************************************
              precision    recall  f1-score   support

           1       1.00      0.99      0.99      3215
           2       0.99      0.97      0.98      3002
           3       0.98      0.99      0.98      3251
           4       0.98      0.99      0.98      3042
           5       0.99      0.99      0.99      3109

   micro avg       0.99      0.99      0.99     15619
   macro avg       0.99      0.99      0.99     15619
weighted avg       0.99      0.99      0.99     15619

**************************************************************

DECISION TREE:
Accuracy score:0.9612651258083104
Classification report:
**************************************************************
              precision    recall  f1-score   support

           1       0.98      0.99      0.99      3215
           2       0.95      0.95      0.95      3002
           3       0.97      0.96      0.96      3251
           4       0.96      0.96      0.96      3042
           5       0.95      0.94      0.95      3109

   micro avg       0.96      0.96      0.96     15619
   macro avg       0.96      0.96      0.96     15619
weighted avg       0.96      0.96      0.96     15619

**************************************************************

LOGISTIC REGRESSION:
```

Figure 2.3



```
Accuracy score:0.7563224278122799
Classification report:
*******************************************************
              precision    recall  f1-score   support

           1       0.78      0.84      0.81      3215
           2       0.81      0.84      0.83      3002
           3       0.74      0.69      0.71      3251
           4       0.64      0.66      0.65      3042
           5       0.81      0.75      0.78      3109

   micro avg       0.76      0.76      0.76     15619
   macro avg       0.76      0.76      0.76     15619
weighted avg       0.76      0.76      0.76     15619

*******************************************************

MLP:
Accuracy score:0.5325565016966515
Classification report:
*******************************************************
/home/smitha/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1143: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted sa
mples.
  'precision', 'predicted', average, warn_for)
/home/smitha/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1143: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted sa
mples.
  'precision', 'predicted', average, warn_for)
/home/smitha/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1143: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted sa
mples.
  'precision', 'predicted', average, warn_for)
              precision    recall  f1-score   support

           1       0.77      0.81      0.79      3215
           2       0.00      0.00      0.00      3002
           3       0.46      0.79      0.58      3251
           4       0.54      0.13      0.21      3042
           5       0.47      0.89      0.61      3109

   micro avg       0.53      0.53      0.53     15619
   macro avg       0.45      0.52      0.44     15619
weighted avg       0.45      0.53      0.45     15619

*******************************************************

(base) smitha@smitha-HP-Pavilion-Notebook:~/smitha_workspace/cmpe255/Project/deceit-detection$
```

Figure 2.4