Using Machine Learning for Lie Detection: Classification of Human Visual Morphology

Supervisor: Dr. Mahbub Alam Majumdar

Co-Supervisor: Mr. Moin Mostakim

Authors:

Samin Azhan – 14101005

Anik Zaman – 17241023

Monjur Rakib Bhuiyan - 17241022

Department of Computer Science and Engineering,

BRAC University

Submitted on: 25th March, 2018

Declaration

This is to certify that the research work titled "Using Machine Learning for Lie Detection: Classification of Human Visual Morphology" is submitted by Samin Azhan, Anik Zaman and Monjur Rakib Bhuiyan to the Department of Computer Science & Engineering, BRAC University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering. We hereby declare that this thesis is based on results found from our own work. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree. We carried out research under the supervision of Dr. Mahbub Alam Majumdar and co-supervision of Mr. Moin Mostakim

Signature of Supervisor:

____

Dr. Mahbub Alam Majumdar

Signature of Co-Supervisor:

____

Mr. Moin Mostakim

Signature of Authors:

____

Samin Azhan

____

Anik Zaman

____

Monjur Rakib Bhuiyan

ACKNOWLEDGEMENTS

Thanks to Almighty Allah, the creator, and the proprietor of this universe for providing us direction, guidance and confidence to finish this research within time.

We are particularly thankful to Dr. Mahbub Alam Majumdar, our thesis supervisor and Mr. Moin Mostakim, our co-supervisor, for their support and guidance in completing our project. We are also grateful to the BRAC University Faculty Staffs of the Computer Science and Engineering, who have been a constant mentor for us in our entire undergraduate study period at BRAC University, especially for building our basics in education and improving our insights.

Lastly, we want to express our genuine appreciation to our parents and siblings for their support and care. We are thankful to all of our friends who helped us directly or indirectly by their support and encouragement to finish our thesis in time.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Though there exists different methods of detecting lies, machine learning can be used to create a reliable and more efficient system to detect lies. This thesis proposes a method of using nonverbal human behaviors to detect lies using machine learning. This lie detection system is based on micro-expressions of human beings which uses Facial Landmark Detection System and Azure Machine Learning. Movements of individual facial muscles are recorded while a person answers some certain questions. By using the two algorithms Two-Class Support Vector Machine and Linear Regression, we attempted to create a machine that can detect lies. We reached an accuracy of approximately 76.2%.

Chapter 01

INTRODUCTION

1.1 INTRODUCTION

Over the years, people have built many machines and software to detect lies. These traditional lie detectors rely upon calculating changes in person's blood pressure, heartbeat/pulse, respiration and skin conductivity. These methods determine abnormal behavior within a person and thus comes to the conclusion if that person is lying or not. However, this procedure is quite weak as it only measures comfort level rather than lie detection. Moreover, it can easily be cheated by people who have professional training in hiding their discomfort. This is why we have introduced artificial intelligence in this affair for better lie detection. By using machine learning, we can determine the facial movements and micro-expressions of a person when he/she is telling the truth and when he/she is telling a lie. Firstly, the idea is to ask the person some common questions of which the answers will be known to us, this is for the AI to gain some data of that person's visual morphology as everyone has different facial expressions. Later when we actually want to know if the person is lying or not, the machine will analyze the person's micro-expressions and facial movements such as eyebrow movement, lip positioning and so on, and compare the newly obtained information with the previously stored data, it will come to a conclusion if that person is actually lying or not. This method is far more superior and efficient than traditional polygraph and it is sure to produce more accurate results providing a proper implementation.

1.2 MOTIVATION

In the early 20th century, the polygraph was developed to detect lies or deception by measuring blood pressure, pulse, skin conductivity, respiration etc. and by finding correlations from these measurements. But the problem with polygraph is, it is highly invasive, very slow and can be fooled. It is inefficient to use in covert operations like espionage missions or intelligence activities because the subject who is being interrogated is aware of the situation that his pulse, blood pressure, level of respiration is being measured. On the other hand, micro-expressions are much more difficult to control compared to pulse, pressure and respiration. Therefore, we wanted to develop a more efficient system for lie detection that belongs in the 21st century technology which is mostly based on facial micro-expressions and artificial intelligence.

1.3 CONTRIBUTION SUMMARY

The contribution summary is given below:

- Developing a system purely based on reading cues from micro expressions.
- Creating a modern alternative to polygraph
- Developing and implementing an algorithm to read facial expressions to detect lies/deception
- Use that data train and create a machine that can detect lies

1.4 THESIS ORIENTATION

The rest of the thesis is organized as following chapters:

- Chapter 2 reviews previous work in lie detection technology such as Face reading technology using FACS and EEG; and background information of Two-Class Support Vector Machine Algorithm, Linear Regression Algorithm, polygraph, dlib library for implementation of the face and expression detection, and Azure Machine Learning Studio.
- Chapter 3 introduces our proposed model along with the summary of the working process.
- Chapter 4 includes experimental setup and result analysis.
- Chapter 5 concludes this thesis and includes future research.

Chapter 02

LITERATURE REVIEW

2.1 E.E.G. & MACHINE LEARNING FOR LIE DETECTION

Jennifer Marsman, a Principal Developer Evangelist in Microsoft's Developer and Platform Evangelism group, along with her husband Chris Caldwell, tried to develop a lie detector using raw EEG data and machine learning [11]. EEG stands for Electroencephalography which is an electrophysiological monitoring method to record electrical activity of the brain. She used a device called the Emotive Epoc+ which allowed her to get the raw EEG data when a person is lying or telling the truth. Her husband Chris was her test subject and she asked him a series of questions and recorded his EEG data through the Emotive headset. Later she used the Microsoft Azure Machine Learning Studio and the Two-Class Decision Jungle Algorithm to create a model that can detect possible lies. She achieved an accuracy of 71.2% with her approach. However, she only tested the method on her husband and she claims that her research is far from complete.

2.2 FACE READING TECHNOLOGY FOR LIE DETECTION BY UNIVERSITY OF BRADFORD

Hassan Ugail, Moi Hoon Yap and Bashar Rajoub tried to create a non-invasive technology for Lie/Guilt Detection. The system was expected to an alternative to polygraph being able to be used in covert situations. They tried detecting action units using FACS and used a thermal camera to trace change of emotion beyond the skin. The Facial Action Coding System (FACS) [1] is a worldwide known system for measuring facial expressions based on the original works of Carl-Herman Hjortsjö. Every AU (Action Unit) represents a different muscle movement in the face. The detailed information of this system is in the FACS manual [1]. It was first distributed in 1978 by Ekman and Friesen and has since experienced amendment. Here are few examples of AUs we used in our thesis project:

AU2 - Outer Brow Raiser, AU4 - Brow Lowerer, AU5 - Upper Eyelid Raise etc.

The researchers believed that particular behaviors can represent particular mental status. As humans have a rigid structure of skull, there are a limited number of facial expressions they can perform. These expressions have been turned into Action Units by Paul Ekman. This project used those action units using Facial Action Coding System to identify facial patterns and compare them. They have emphasized on the action units that can have greater change while a person is lying. Then using these series of different expressions the project keeps the necessary data in database to merge with the data from thermal analysis. As the skin temperature can outperform polygraph in measuring behavioral aspects, large psychological stress can be determined with greater accuracy. The system has been able to achieve 70% accuracy rate.

2.3 TWO-CLASS SUPPORT VECTOR MACHINE

Support vector machines (SVMs) are a well-researched class of supervised learning methods. Supervised learning is a machine learning method that enables the machine to map an input to an output based on example input-output pairs and this particular implementation of SVM is suited to make predictions of two possible outcomes, based on either continuous or categorical variables. Support vector machines are among the earliest of machine learning algorithms, and SVM models have been used in many applications, from information retrieval to text and image classification. SVMs can be used for both classification and regression tasks.

Two-Class Support Vector Machine is an algorithm that performs well for experiments that contain more than a hundred features. This SVM model is a supervised learning model that requires labeled data. In the training process, the algorithm analyzes input data and recognizes patterns in a multi-dimensional feature space called the hyperplane. All input examples are represented as points in this space, and are mapped to output categories in such a way that categories are divided by as wide and clear a gap as possible.

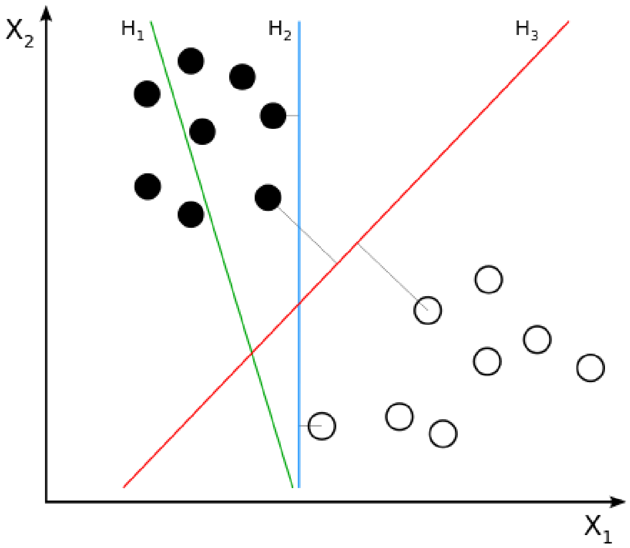


Fig 2.1: Support Vector Machine (Source: Wikipedia)

In the figure H_1, H_2 and H_3 are possible hyperplanes and the two different dots represent to different classes. For prediction, the SVM algorithm assigns new examples into one category or the other, mapping them into that same space.

2.4 LINEAR REGRESSION

Linear regression is a common statistical method, which has been adopted in machine learning and enhanced with many new methods for fitting the line and measuring error. In the most basic sense, regression refers to prediction of a numeric target. Linear regression is still a good choice when it is applied on a very simple model for a basic predictive task. Linear regression also tends to work well on high-dimensional, sparse data sets lacking complexity.

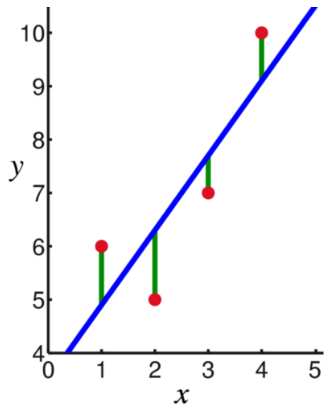


Fig 2.2: Linear Regression (Source: Wikipedia)

Linear Regression is made with an assumption that there's a linear relationship between X and Y.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

Here, y is the response, β values are called the model coefficients. These values are "learned" during the model fitting/training step, x_1 is the first feature and x_n the n^{th} feature. This model tries to predict the value of y depending on the model coefficients and features.

2.5 POLYGRAPH

Lie detection as a technique of investigating guilt has been around for years. Though the orthodox techniques are manual and depend mostly on the analyzation of the human who is acting as an investigator.

The most popular technique of lie detection is with the help of a polygraph. Polygraph measures and records a few physiological pointers, for example, blood pressure, heartbeat, breathe, and skin conductivity while the subject is asked and answers a series of questions [10]. From the information, we collected from article [2,3], we can say that the hypothesis/theory this strategy stands on is that suspicious or misleading answers will create physiological reactions that can be separated from those related to non-misleading answers. But the problem with polygraph is, it is a highly expensive approach and it is highly invasive.

2.6 'dlib' LIBRARY FOR DETECTING FACIAL LANDMARKS

'dlib' is a library of python to detect facial landmarks [5]. Facial landmarks are used to localize and represent most noticeable regions of the face, such as:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

According to the article [5], the pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face. The indexes of all 68 coordinates are illustrated on [Fig 2.3].

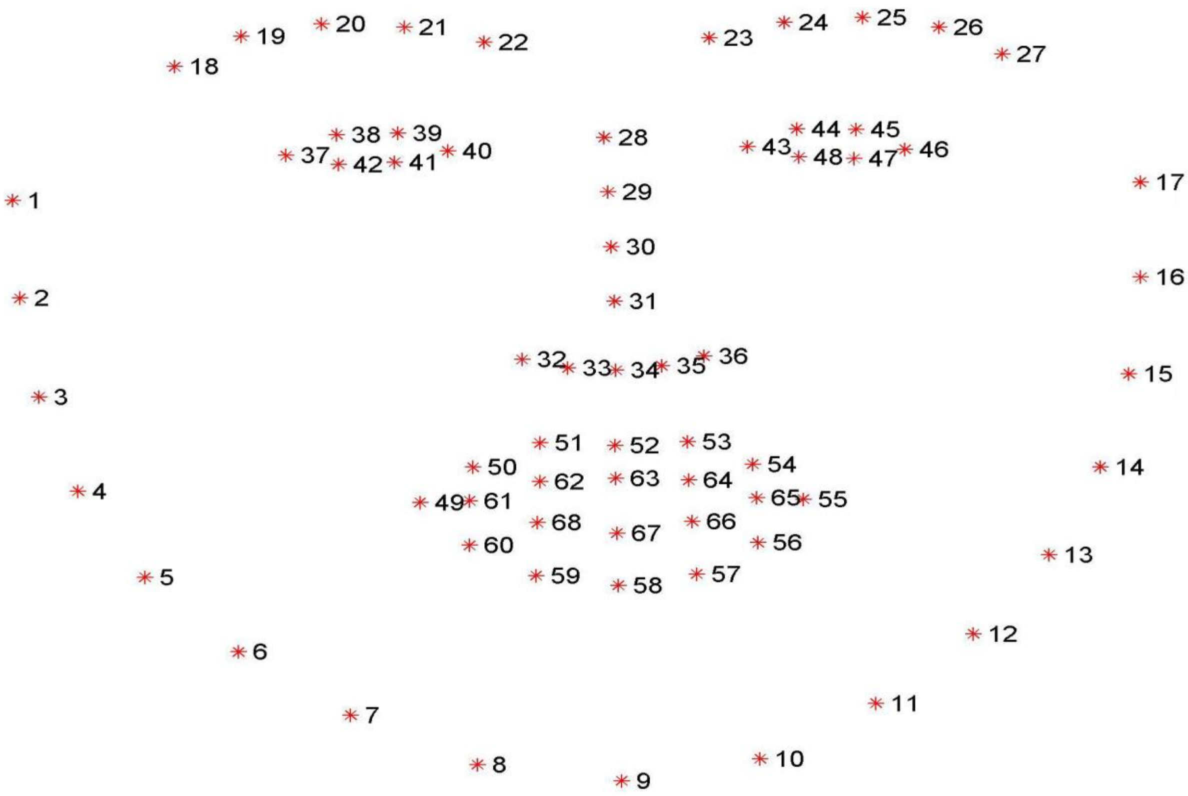


Fig 2.3: Diagram of the 68(x, y) facial landmark coordinates (Source: pyimagesearch.com)

All of these coordinates signify different landmarks of the face, such as coordinates 18-22 represents the right eyebrow. So, by detecting all these coordinates of a face, we can detect which parts of the face are moving in real time resulting in the detection of micro expressions. The end result from the micro expressions and action units is predicting the possibility of telling if a person is being truthful or deceptive.

2.7 AZURE MACHINE LEARNING

Azure Machine Learning Studio is a collaborative tool that can used to build, test, and deploy predictive analytics solutions on given data. This tool is created by Microsoft and it has perfectly implemented 25 machine learning algorithms including Logistic regression, Two-class boosted decision tree, Support vector machine Algorithm and so on.

Azure Machine Learning Studio provides an interactive, visual workspace to easily build, test, and iterate on a predictive analysis model [Fig 2.4]. One can drag-and-drop datasets and analysis modules onto an interactive canvas, connecting them together to form an experiment, which can be executed in Machine Learning Studio. To iterate model design, one can edit the experiment, save a copy if desired, and run it again. It is also possible to convert training experiment to a predictive experiment, and then publish it as a web service so that the model can be accessed by others.

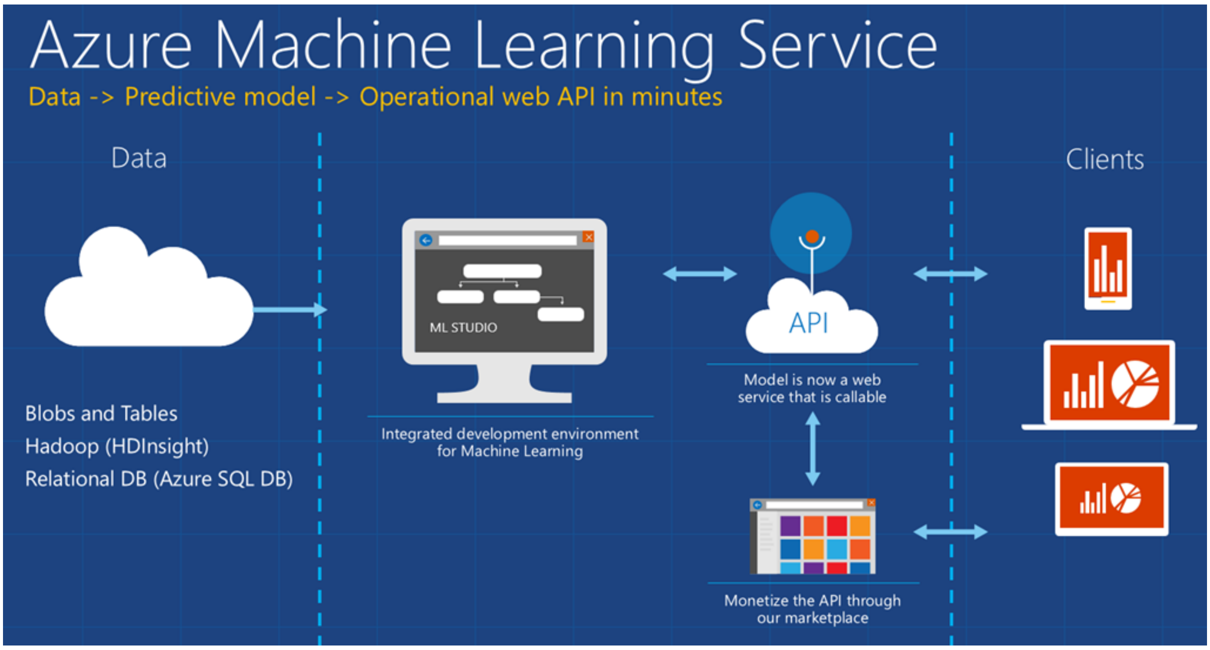


Fig 2.4: Azure Machine Learning (Source: msdn.microsoft.com)

We are using Azure Machine Learning Studios by importing the data we get from using dlib on our volunteers and applying the algorithms Two-Class Support Vector Machine and Linear Regression to create a model and train it so that it can detect if the person is telling lies or the truth.

2.8 ACCURACY AND LIMITATIONS

Research suggests that machines can detect lies better than a human being but with recognizable error rates. It is also to be added that techniques used to beat polygraph tests, so-called countermeasures, may be effective. Despite unreliability, results of machines in lie detection are accepted for analysis in court in countries such as Japan.

Paul Ekman has used the Facial Action Coding System (FACS) and when combined with voice and speech measures, it reaches detection accuracy rates of up to 90 percent. However, there is currently no evidence to support such a claim. His experiments utilize micro-expressions, which last less than one-fifth of a second, and may leak some emotions somebody may or may not want to cover up, for example, outrage or guilt [7]. However, it has been taken into consideration that signs of emotion don't necessarily represent signs of guilt. An innocent person may be a victimized and appear guilty. As to his experiments [6], lies about feelings right now have the greatest result from facial expressions and voice signs while lies about convictions and activities, for example, crimes utilize signals from gestures and words are included.

Moreover, our project is solely based on the visual changes on a person's face. We do not think it is possible to create something that can detect lies with an accuracy of 100% only through visual data as lies are more of a matter in the mind. However, we humans tend to express our emotions with the different expressions of our face and it is possible for us to analyze those expression to try and predict if that person is lying or telling the truth.

In addition to that, not much research has been done on visual lie detection up to this point and it is very difficult for us to obtain a large amount of data to train our machine properly.

Chapter 03

PROPOSED MODEL

3.1 INTRODUCTION TO PROPOSED MODEL

Our system use three basic steps. We start with finding a face in the streaming as soon as the questions are asked. The algorithm then searches for the group of pixels clustered in a pattern which resembles a human face. Then it moves on to feature extraction. The approach relies on finding and analyzing geometric features of eyes, lips, nose and the line of mouth and understand the relative changes between them. This data is used for recognizing different types of expressions in the face. The presence of some expression can draw the line between lie and truth. Our project is focused to detect the change in expressions on the face and through the algorithm, come to a conclusion if there's any sign of deception or guilt.

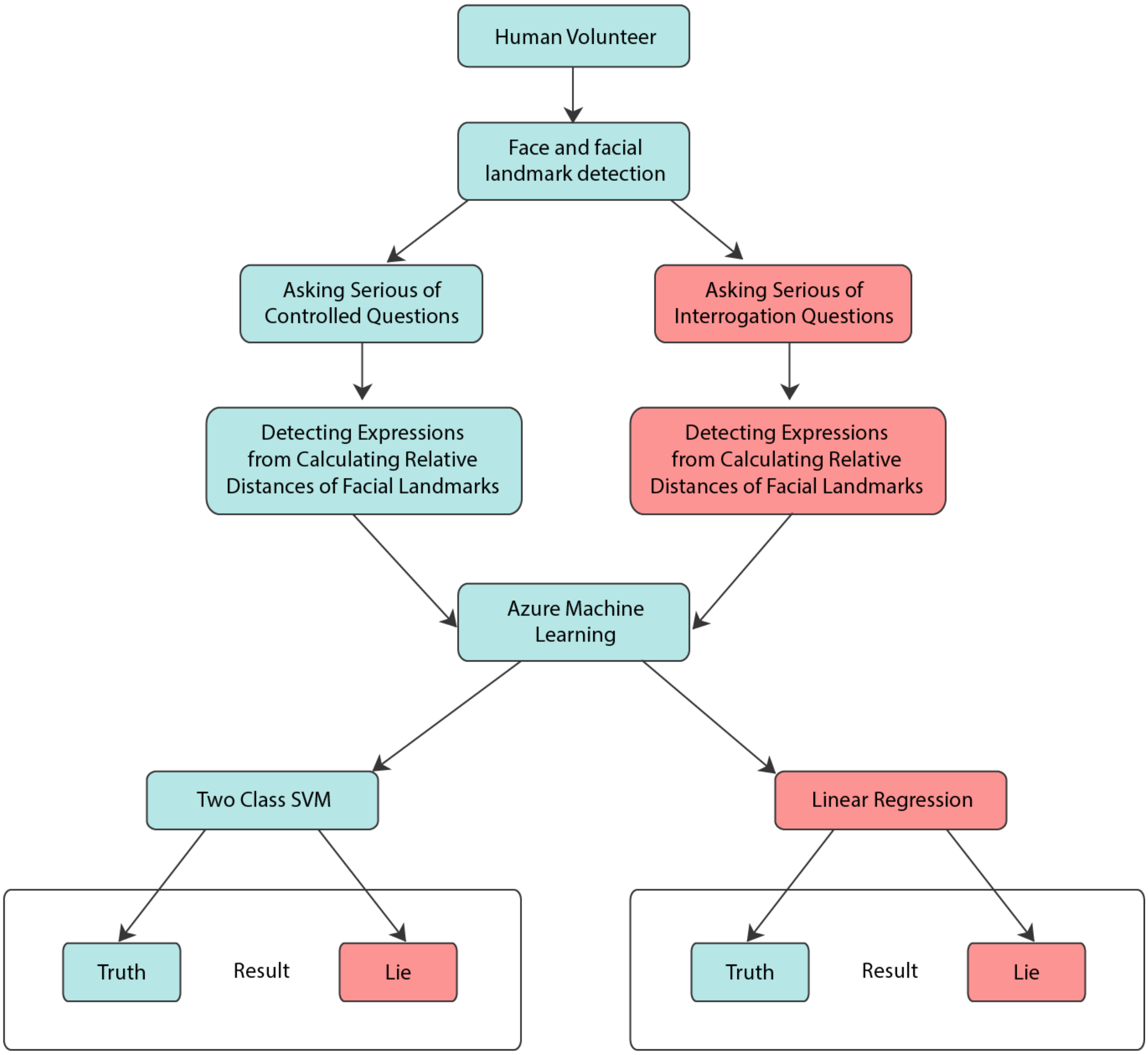


Fig 3.1: Workflow of our proposed system

Figure 3.1 is a demonstration of the workflow of our proposed system. Firstly, we detect the face and facial landmarks of the human volunteer for our experiment in real-time through our algorithm and our camera. Next, we ask the volunteer some controlled questions of which the answers are known to us and the volunteer. In this stage, the volunteer is required to tell the truth. The algorithm then starts to analyze the expressions by of the volunteer while answering the questions by calculating movements of the facial landmarks.

In the next part we ask some questions that the volunteer has to lie while replying. The algorithm analyzes and stores the data same as before, only this time it is learning how that person behaves while telling lies.

Next comes the complex part of our workflow. We start asking a series of interrogation questions and our algorithm does the same analysis when we were asking the controlled questions. In this stage, the volunteer can choose to lie or tell the truth. We take all these data and use it first to create and train our model in Azure Machine Learning. We will be using the algorithms Two-Class Support Machine and Linear Regression, and then evaluate the model to determine how accurate it is.

3.2 FACE AND FACIAL LANDMARKS DETECTION

To detect face and facial landmarks of a human subject we used 'dlib' library of python. It has written in the literature review of this library, the pre-trained facial landmark detector inside the 'dlib' library can estimate the location of 68 (x, y)-coordinates which map to facial structures on the face of the subject. In Fig 3.2 of page (16), we can see a demonstration of the facial landmarks detection. The violet color curves consisting of all the 68 coordinates represents key facial attributes of the subject. All of the facial landmarks (eyebrow, eyes, nose, mouth and jawline) are drawn in violet curves and the bounding box of the face is drawn in red. The facial landmarks curve and the bounding box moves if the subject showcase any expressions in real time. Thus, from the movements of the curves, we can analyze which action unit should be labeled for which expression.

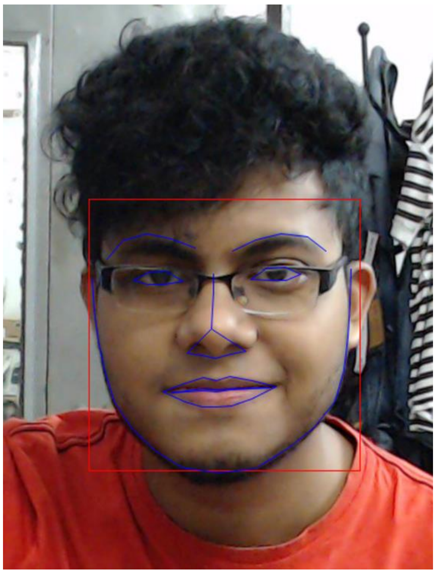


Fig 3.2: Facial landmarks are detected in an image or video

3.3 STORING DATA FOR AZURE ML

Azure Machine Learning Studio has a few ways to read data from provided datasets and one of those formats is a .csv file. This type of file can also be opened and manipulated using another Microsoft software known as Microsoft Excel. After getting the 68 points using dlib, we must break each point into x,y coordinates and store them separately. This gives us 136 values for each detection. We also need a “is_Truth” column so we can label the gathered data as lie or truth (0 being lie and 1 being true). In the end, we have a total number of 137 columns of which 136 are features and the last one is the label.

3.4 MAKING SENSE OF BEHAVIOUR

As we've mentioned earlier our three steps questionnaire helps us to make sense of the behavior of the volunteer. The first two steps are to detect the general approach of countering a lie or truth. The third approach compares the data with the previous steps and generates the final result. The first two steps take ten question each so that we can get a larger amount of diverse data to compare in order to be more accurate. Not to be mentioned that, the more questions asked at the first two steps, more the possibility of detection lies in the third step increases. We are importing all of the 137 columns and letting the machine learn how a human might behave when telling the truth and how he/she behave while telling a lie.

Chapter 04

EXPERIMENTAL SETUP AND RESULT ANALYSIS

4.1 INTRODUCTION TO EXPERIMENTAL SETUP

The system we have been working on is mainly dependent on the camera in the first phase. We have used basic 15 megapixel USB camera prioritizing formulating the algorithm first. The system is made using the Python programming language. For detecting face we have integrated dlib library with it. The 68 points of dlib library provided to detect different points of face helps to move forward and interpret a practical shape from it.

When the installation of useful softwares and libraries were done, we approached towards storing the data in an acceptable and readable format for Azure machine learning studio. As previously mentioned, we are gathering a total of 137 columns including 136 features and 1 label. The 136 features contain the x,y coordinates of the 68 facial landmark points and the column names are set as 1_x, 1_y, 2_x, 2_y, …, 68_x, 68_y. As it is easier for Python's machine learning libraries to read of integer values, we decided to keep these values as integers.

After storing the data on a .csv file, It is time to create a machine learning model using Two-Class SVM. For the next part the system requires three step interview session of a volunteer. The first two steps are known questions where the volunteer has to answer truth or lie depending on the demand of system. This is the learning step for the system, which is used in the third session of questionnaire. Now the answers of the volunteer will be of his wish as will be compared will the data received during the first two sessions. Then the algorithm decides the merit of answer based on the intensity of the volunteer's natural traits. The data may contain multiple occurrence of Action Units in which case the intensity of the Unit comes into play.

In short, the steps we have followed are:

1. Setting up necessary softwares
2. Detecting the face

3. Interviewing
4. Storing Data for Azure ML
5. Create training model using Two-class SVM & Linear Regression in Azure ML
6. Result & Accuracy

4.2 SETTING UP NECESSARY SOFTWARES

The system uses Python programming language. Python is a widely used high level programming language. We are currently using Python 2.7. This language supports automatic memory management and comes with many useful libraries that can be used at our system.

Besides that, we are using OpenCV to detect faces. OpenCV is a open source Python library widely used in detecting faces in real-time.

In addition to that, we are using a library called dlib. Dlib was initially created for C++ but now has an implementation of Python. However, installing dlib for Python was not a simple task. First, we need to install Visual Studio 2015, then a software called “Cmake”. After that we needed Anaconda to make sure all the packages needed to run dlib was properly installed. Then we had to build the dlib library and set environment variable dlib_DIR. After multiple attempts we finally got it running and was able to use it to gather the 68 facial landmarks.

4.3 DETECTING THE FACE

In dlib, facial landmarks are tracked in geometrical system. The x,y value of a point reflects the specific facial coordinates of the landmarks. Here training data teaches how to detect facial landmarks positions from pixels and turn them into combination of specific facial landmarks. The library estimates the face based on 68 landmark positions distributed on face with each landmark having an index number designated.

In Figure (4.3.1), there are two pictures of before and after detecting the 68 landmark positions. Each white dot contains (x,y) coordinates and with the help of dlib, we are able to calculate these values in real-time and on still images.

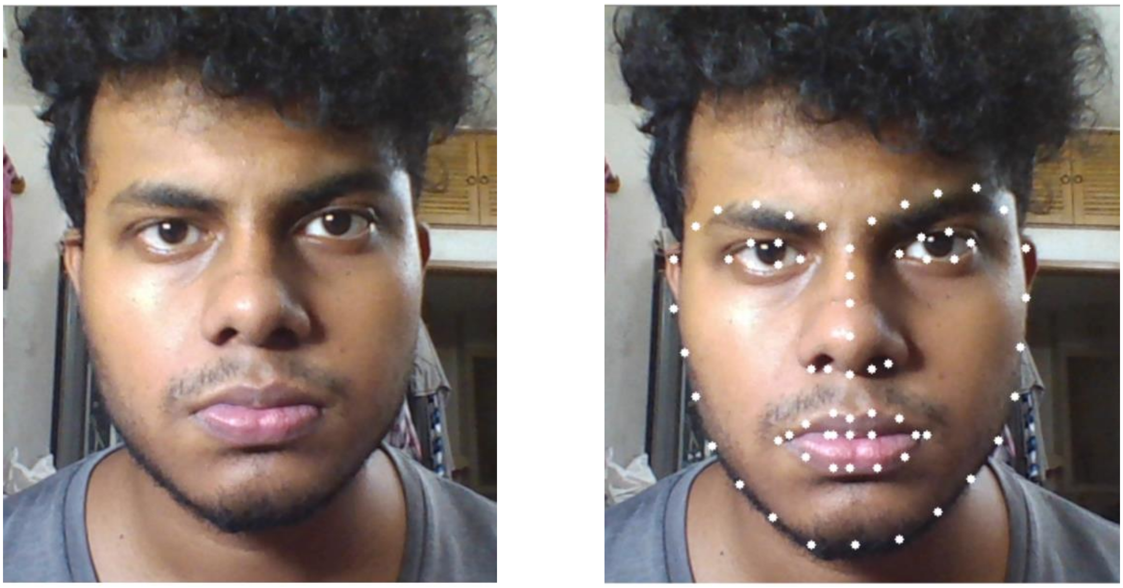


Fig 4.3.1: Facial Landmark 68 Markup Implementation

In the code, we kept an array named ‘shape’ whose length is 68. This array holds all the 68 landmark points within itself. Each index of the array holds two points which are the x and y coordinates of that particular index. After achieving the values of all these points, it was possible for us to work with them and store them to train and evaluate our model.

4.4 INTERVIEWING

As mentioned before, the interview session is divided in three segments. A volunteer has to take part in all three of them in order to complete the measurement. The skipping of the first two question segments will result in having to outcome in the last one. More the questions are in the first two sessions the accuracy of the last step increases.

The Interview steps are:

- The Truth Segment
- The Lie Segment
- Interrogation Segment

4.4.1 The Truth Segment:

Here we ask some predetermined questions to the volunteer. The answers are given to them beforehand and they have to tell the truth regardless of their will or instinct. The volunteers need to be flexible, honest to the cause and convincing while answering the questions.

A set of example questions are:

Questions	**Expected True Answers**
Did you volunteer for this session?	Yes
Is your name …?	Yes
Is the earth flat?	No
Is water a solid?	No
Do we need Oxygen to survive?	Yes
Is William Shakespeare a scientist?	No
Is Russia the largest country in the world?	Yes

Is Hillary Clinton the current president of USA?	No
Do dogs live in the sea?	No
Does the sun rise in the east?	Yes

Table 4.4.1.1: Truth Segment Questions and Answers

4.4.2 The Lie Segment:

Here we ask some predetermined questions to the volunteer. The answers are given to them beforehand and they have to tell the lies regardless of their will or instinct. The volunteers need to be flexible, honest to the cause and convincing while answering the questions.

A set of example questions are:

Questions	Expected Lie Answers
Did you volunteer for this session?	No
Is your name …?	No
Is the earth flat?	Yes
Is water a solid?	Yes
Do we need Oxygen to survive?	No
Is William Shakespeare a scientist?	Yes

Is Russia the largest country in the world?	No
Is Hillary Clinton the current president of USA?	Yes
Do dogs live in the sea?	Yes
Does the sun rise in the east?	No

Table 4.4.2.1: Lie Segment Questions and Answers

4.4.3 Interrogation Segment:

In this segment the volunteers are asked some predetermined questions but not given to them beforehand. The volunteer has to answer each question in a given period of time without thinking much. Not being able to answer will also have impact on the measurement of detecting the perspective or wellness of telling lies. Many of these questions are controversial and the answers may be considered a taboo in certain societies giving the opportunity to have a more genuine outcome of facial expression.

Some of the example questions of this segments are:

Questions
Do you support the legalization of gay marriage?
Did you ever cheat with your partner?
Do I look good?
(After explaining a complex Socioeconomic-Scientific theory) Do you understand?

Do you think your parents are happy together?

Table 4.4.3.1: Interrogation Segment Sample Questions

Obviously more radical and tricky questions are encouraged and encountered during the interrogation segment.

4.5 STORING DATA FOR AZURE MACHINE LEARNING

After the interview session, it is important to store the 68 facial landmark values in a suitable way for azure machine learning studio. There are a number of ways to import datasets to azure machine learning studio and one of the best ways is to create a .csv file. These file are accepted for azure and also can be opened using Microsoft Excel to check and validate the data for ourselves.

58_y	59_x	59_y	60_x	60_y	61_x	61_y	62_x	62_y	63_x	63_y	64_x	64_y	65_x	65_y	66_x	66_y	67_x	67_y	68_x	68_y	is_Truth
77	78	76	73	75	70	71	78	69	82	70	86	69	94	70	86	69	82	70	78	69	1
76	77	76	72	74	69	70	77	69	81	69	85	69	94	69	85	68	81	69	77	68	1
76	76	76	71	74	68	70	76	69	80	69	84	69	93	69	84	68	80	69	76	68	1
76	76	76	71	74	68	70	77	69	80	69	85	69	93	69	84	68	80	69	77	69	0
76	76	76	71	74	68	70	76	69	80	69	84	68	93	70	84	69	80	69	76	69	1
76	76	76	71	74	68	70	76	69	80	69	84	69	92	69	84	69	80	69	76	69	0
76	76	76	71	74	68	70	76	69	80	70	84	69	92	70	84	69	80	69	76	69	0
76	76	76	71	74	68	71	76	69	80	70	84	69	93	70	84	69	80	69	76	69	1
77	76	76	70	75	68	71	76	69	80	70	84	69	93	70	84	69	80	70	76	69	0
76	76	76	71	74	68	71	76	69	80	70	84	69	93	70	84	69	80	69	76	69	1
76	76	76	71	74	68	70	76	69	80	70	84	69	92	70	84	69	80	69	76	69	0
76	76	76	71	75	68	71	76	69	80	70	84	69	92	70	84	69	80	69	76	69	1
77	76	76	71	75	68	71	76	69	80	70	84	69	93	70	84	69	80	70	76	69	0
77	76	76	71	75	68	71	76	70	80	70	84	69	93	70	84	69	80	70	76	69	1
77	76	76	71	75	68	71	77	70	80	70	85	70	93	70	85	69	80	70	77	69	1
77	76	77	71	75	68	71	77	70	80	70	84	69	92	70	84	69	80	70	77	70	0
77	76	77	71	75	68	71	77	70	81	70	85	70	93	70	85	69	81	70	77	70	1
77	76	77	71	75	68	71	77	70	81	70	85	70	93	70	85	70	81	70	77	70	1
77	77	77	71	75	68	71	77	70	81	70	85	70	93	70	85	69	81	70	77	70	0
77	77	77	71	75	68	71	77	70	81	70	85	70	93	70	85	69	81	70	77	70	1
77	77	77	71	76	68	71	77	70	81	70	85	70	93	70	85	70	81	70	77	70	0
77	77	77	71	75	68	71	77	70	81	70	85	70	93	71	85	70	81	70	77	70	1
77	77	77	71	75	68	71	77	70	81	70	85	70	93	70	85	70	81	70	77	70	0
77	77	77	71	75	68	71	77	70	81	70	85	70	93	70	85	69	81	70	77	70	0
77	77	77	72	76	69	71	77	70	81	70	85	70	93	71	85	70	81	70	77	70	0
77	77	77	72	76	68	71	77	70	81	70	85	70	93	70	85	69	81	70	77	70	1
77	77	77	72	76	69	71	77	70	81	70	85	70	93	71	85	70	81	70	77	70	1

Fig 4.5.1: Dataset for Azure Machine Learning

As dlib gave us the 68 points, now we must break each point into x,y coordinates and store them separately. This gives us 136 values for each detection. We also need a "is_Truth" column so we can label the gathered data as lie or truth (0 being lie and 1 being true).

As a result, we have a total number of 137 columns of which 136 are features and the last one is the label. As shown in the picture, the first 136 columns are named as 1_x, 1_y, 2_x, 2_y…

68_x, 68_y and the last column is named as “is_Truth”. For our project we gathered data from 38 volunteers, mostly containing our family and friends, and our dataset consists of 5094 rows each containing the 137 columns.

4.6 CREATING TRAINING MODEL USING TWO-CLASS SVM AND LINEAR REGRESSION IN AZURE ML

After storing the data for Azure Machine Learning Studio, it is time for us to actually create our training model. In Azure Machine Learning Studio, first we must save our dataset by importing dataset from local files. After that, we need a “Select Column in Dataset” module to select the 137 columns. Then we are using a **Split Data** module which enables us to randomize all the gathered data and split it into two sets first containing 80% and the second containing the remaining 20%.

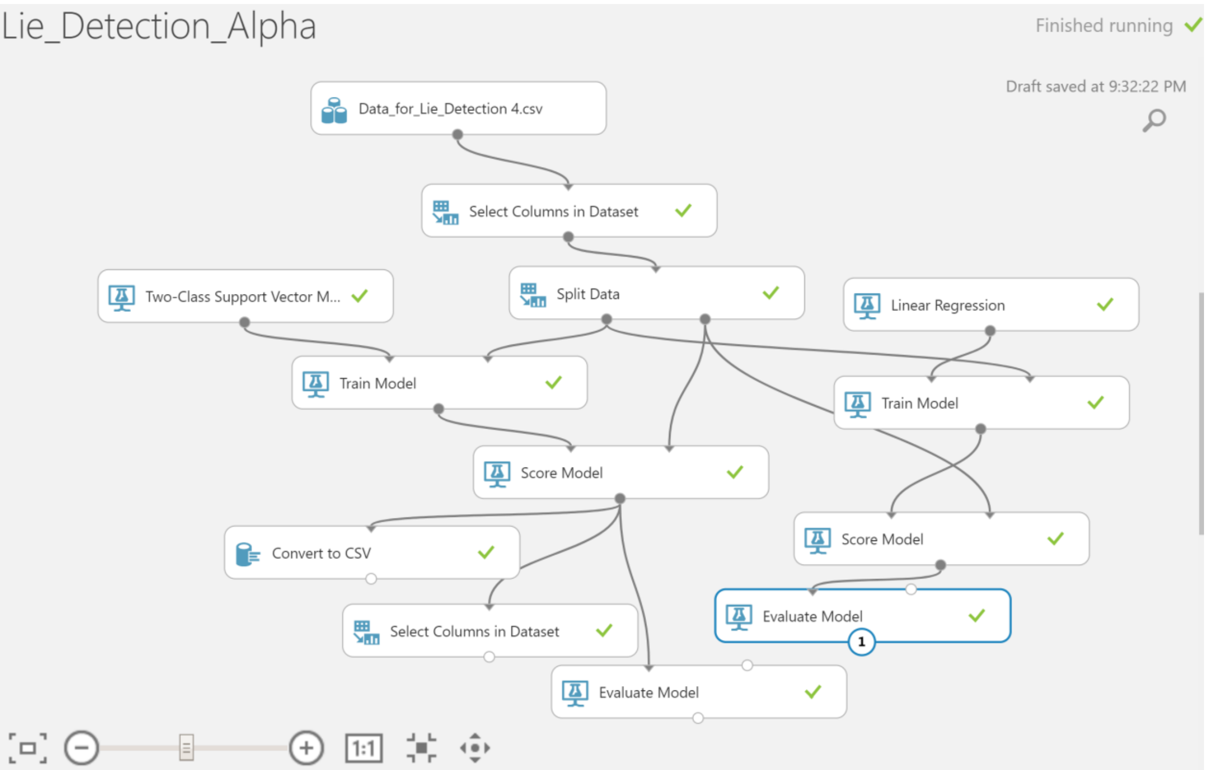


Fig 4.6.1: Our Model in Azure ML

Then we used the built-in perfect implementation of the SVM algorithm named **Two-Class SVM** module which works well in a case like ours that contains more than 100 features and has to make a decision between 2 labels.

In the stored data, we are labeling lies as is_Truth = 0 and labeling truth as is_Truth = 1. As a result, the Two-Class SVM algorithm separates these two classes by the features provided in the first 136 columns and draws the best possible hyperplane to differentiate them [Fig 4.6.2].

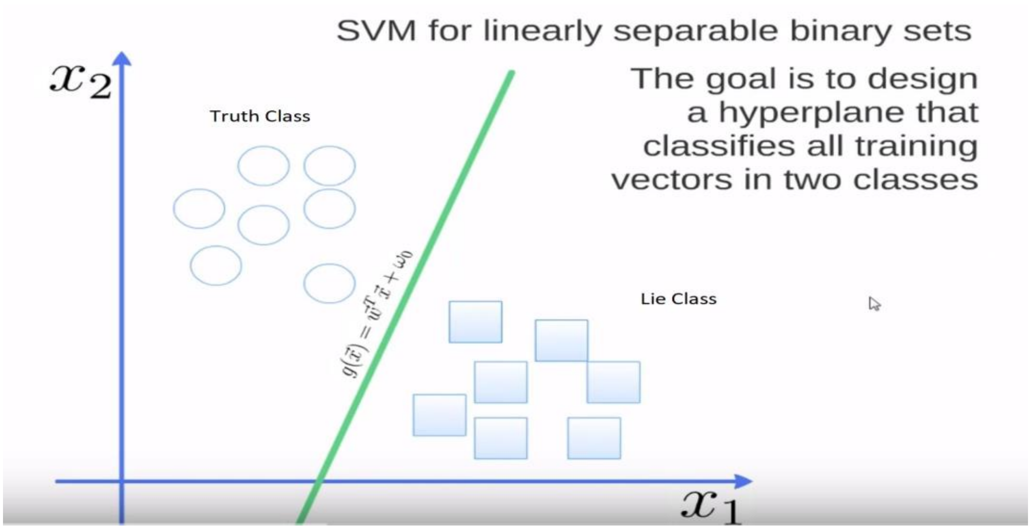


Fig 4.6.2: Two-Class SVM trying to detect lies

Similarly, we will use **Linear Regression** module in order to compare the two algorithms to see which performs better than the other. The Linear Regression algorithm is a fast training model under supervised learning and with the given 136 features, it will make a prediction of the is_Truth column.

After using the two algorithms, we need a **Train** module for each of them which has two inputs, one for the algorithm and the other for the data to be trained with. Previously, we have split all our data into 80% and 20%. Now we will be using that 80% on the second input of the Train module to train our machine.

When training is complete, we need to test the model by adding a **Score** module for each algorithm. This module also has two inputs. One from the trained model and one for testing some given data. We will be using the remaining 20% data to test our model. Azure Machine

Studio also gives us great visualization to help us understand better. If we right click on the score module and click visualize we can see the following figure.

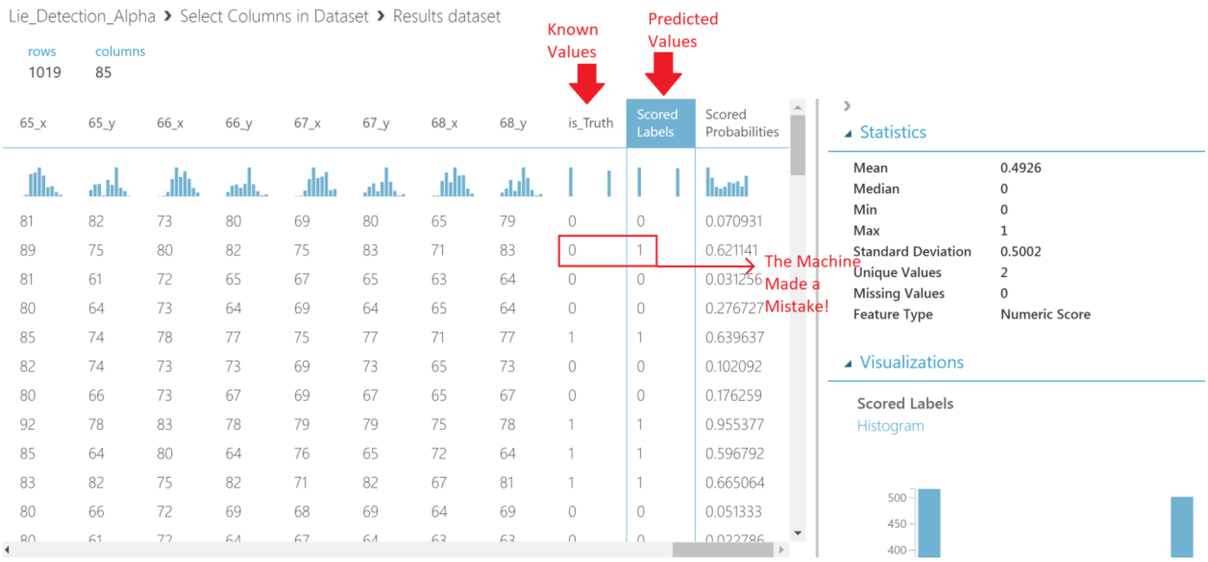
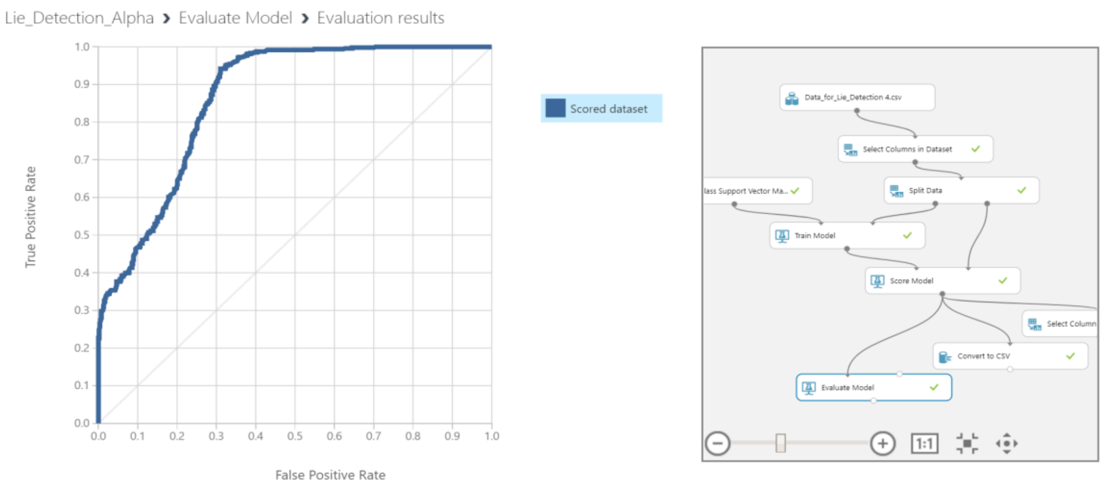


Fig 4.6.3: Model's Scored Labels

Here, we see that is_Truth column is provided for the testing 20% data and next to it, a new column named "Scored Labels" has been created. This is the machine making its best predictions. As shown in the figure, the machine made a mistake on the 2nd row, with the given data the machine predicted it was the truth while we see it was actually a lie by looking at the is_Truth column. Next to the Scored Labels column, we see a column named "Scored Probabilities". The values of this column are determined the probability on which the machine relied on to come up with its decision. When the machine made a mistake we see the probability was 0.62. If we imagine the hyper plane created by the Two-Class SVM algorithm was on the 0.5 mark, then the machine predicted that the given data for this iteration was slightly on the truth class. So it decided that the person was telling the truth. Similarly, in Linear Regression, after analyzing the given data and features, it found a match of the label nearest to truthful behavior. And as a result, it predicted it to be true but actually it was a lie. As we see, the machine was not far off in both the cases. For small fractional values the machine decided wrong sometimes. But then again, we cannot consider a machine to be absolutely accurate all the time.

4.7 RESULT AND ACCURACY

To see the results and judge the accuracy of our model, we need to add the **Evaluate Model** module. This module requires a score model input on which it can evaluate. If we right click on evaluate mode and click visualize, we can see the accuracy of our model. First let's check the evaluation of our Linear Regression model. The relative absolute error of this model was quite high and as a result our Linear Regression model scored an accuracy of 61.8%. In total 1019 iterations, it got 630 right and 389 wrong. We believe the reason behind this low accuracy is that we are working with a large number of features and an algorithm like Linear Regression is not so suitable for a case like ours. If it was possible for us to compress our data and work with less features then we were bound to get better results in this algorithm. On the other hand, the Two-Class SVM performed much better. If we click visualize on the Evaluate module of the Two-Class SVM, what we get is shown on [Fig 4.7.1].



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
370	111	0.762	0.737	0.5	0.863

False Positive	True Negative	Recall	F1 Score
132	406	0.769	0.753

Positive Label	Negative Label
1	0

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	131	3	0.132	0.654	0.426	0.978	0.272	0.605	0.994	0.001
(0.800,0.900]	51	26	0.207	0.678	0.526	0.863	0.378	0.630	0.946	0.018
(0.700,0.800]	61	40	0.306	0.699	0.613	0.779	0.505	0.663	0.872	0.051
(0.600,0.700]	56	36	0.396	0.718	0.676	0.740	0.622	0.704	0.805	0.088
(0.500,0.600]	71	27	0.493	0.762	0.753	0.737	0.769	0.785	0.755	0.123
(0.400,0.500]	46	22	0.559	0.785	0.792	0.730	0.865	0.855	0.714	0.157

Fig 4.7.1: Evaluating Model

From the figure, we see a graph of which the X-Axis is True Positive Rate and the Y-Axis is False Positive Rate. For a model to be accurate, it must have a high True Positive Rate and a low False Positive Rate. For our model, the True Positive indicates that it was able to detect the truth correctly and False Positive indicates that it predicted the answer was true but actually it was false. Similarly, True Negative means it was correct in prediction a lie and False Negative means it thought it was a lie but actually it was the truth. As a result, the total of True Positives and True Negatives are the ones that the machine got right and the total of False Positives and False Negatives are the ones it got wrong. We see from a total of 1019 test cases, it got 776 right and 243 wrong. As a result, our model is approximately 76.2% accurate.

Chapter 05

CONCLUSION AND FUTURE RESEARCH

5.1 CONCLUSION

Through our research we have tried to accomplish the proposed system to detect lies with the help of Machine Learning. Our idea of analyzing Human Visual Morphology and using Facial Landmark Detection System is used as a means of detecting micro expressions precisely. Then by the mean of three step questionnaire system and training and evaluating our model, we have been approximately 76.2% accurate in detecting lies. However, there are some challenges in our proposed system, such as: faces can be covered by makeup powder for which extracting the correct expression might get even harder, wearing dark sunglasses sometimes becomes an obstruction for detecting facial landmarks properly. Furthermore, our research led us to the fact that lie detection should not concealed in visual morphology alone. To detect lies more accurately, we need to understand brain activity by extracting and analyzing the raw EEG data and also examine voice and speech measures. However, these processes are quite expensive, still not properly implemented and are still in need of future research. While visual data combined with the brain signal and speech measures data is most likely to produce the best lie detection system, we believe our project was a huge leap forward. We hereby conclude our research with the hope of seeing the day on which the perfect lie detector is created with the help of our proposed model.

5.2 FUTURE PLANS

In the future we want the research of this project to go one step ahead in order to learn from its previous results and come up with more accurate decision making skills. In addition, we want to increase the accuracy level by replacing our hardware system with high resolution DSLR cameras and take multi angled shots for capturing the emotions more precisely. More importantly, given the right technologies, we would like to capture and analyze raw EEG data from the brain, collect and examine voice and speech data and also include a thermal sensor determine the change of temperature in one's body while answering certain questions. We also want to test this model on a minimum number of 100,000 people and create a huge dataset. We believe if all these future researches could be done properly, one day we will have a lie detector that can detect lies 99% of the time or maybe even a 100%.

REFERENCES:

[1] Ekman, P., Hager, J. C., & Friesen, W. V. (2002). Facial action coding system: the manual. Salt Lake City: Research Nexus.

[2] Moore, M. H. (2002). The polygraph and lie detection. Washington, D.C.: National Academy Press.

[3] Meijer, E. H., & Verschuere, B. (2014). The Polygraph. Detecting Deception, 59-80. doi: 10.1002/9781118510001.ch3

[4] Choi, O. (2015). Using fMRI for Lie Detection: Ready for Court? Psychiatric Expert Testimony: Emerging Applications, 84-101. doi:10.1093/med/9780199346592.003.0006

[5] Facial landmarks with dlib, OpenCV, and Python. (2017, April 30). Retrieved December 09, 2017, from https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[6] Tian, Y., Kanade, T., & Cohn, J. F. (n.d.). Facial Expression Analysis. Handbook of Face Recognition, 247-275. doi:10.1007/0-387-27257-7_12

[7] Adelson, Rachel (July 2004). "Detecting Deception". Monitor on Psychology. American Psychological Association. 37 (7): 70. Retrieved 26 April 2012.

[8] Yap, M. (2014). Facial Behavioral Analysis: A Case Study in Deception Detection. British Journal of Applied Science & Technology, 4(10), 1485-1496. doi:10.9734/bjast/2014/6369

[9] Cascia, M. L., Valenti, L., & Sclaroff, S. (n.d.). Fully automatic, real-time detection of facial gestures from generic video. IEEE 6th Workshop on Multimedia Signal Processing, 2004. doi:10.1109/mmsp.2004.1436520

[10] Trovillo, P. V. (1939). A history of lie detection. Chicago.

[11] Marsman J., Caldwell C. (2015) Using EEG & Azure Machine Learning to Build a Lie Detector. Retrieved March 23, 2018.