VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"MACHINE LEARNING TO PREDICTING DELAYED ONSET OF TRAUMA FOLLOWING AN ISCHEMIC STROKE"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

ADITYA M. KAKDE (1CR16CS008)

AMUDEESHWARAN S. (1CR16CS016)

Under the Guidance of,

Dr. Sugato Chakrabarty

Professor

Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"MACHINE LEARNING FOR PREDICTING DELAYED ONSET OF TRAUMA POST AN ISCHEMIC STROKE"** carried out by **Mr. ADITYA M. KAKDE,** USN : **1CR16CS008, Mr. AMUDEESHWARAN S.,** USN **: 1CR16CS016**, bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Dr. Sugato Chakrabarty	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

External Viva

Name of the examiners	Signature with date
1.	_____
2.	_____

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "" **MACHINE LEARNING FOR PREDICTING DELAYED ONSET OF TRAUMA POST AN ISCHEMIC STROKE** " has been successfully completed under the guidance of Dr. Sugato Chakrabarty, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date: 30th May 2020

Team members:

ADITYA M. KAKDE (1CR16CS008) _____

AMUDEESHWARAN S. (1CR16CS016) _____

ABSTRACT

Strokes are currently the third leading cause of fatality globally. However only a small percentage of patients die immediately after the trauma. Some of the leading causes that eventually lead to death may be initial ischemic infarction, recurrent ischemic stroke, recurrent haemorrhagic stroke, pneumonia, coronary artery disease, pulmonary embolism, and other vascular or nonvascular causes. Most studies that apply machine learning to stroke focus on predicting the risk of having a stroke or the likelihood of survival given attributes of a patient, but not so much on likely outcomes of patients that do survive the initial stroke attack. Therefore, the goal of our project is to apply principles of machine learning over large existing data sets to effectively predict the most probable life threatening risks that may follow the first incident.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Prem Kumar Ramesh,** HOD, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide

Dr. Sugato Chakrabarty, Professor**,** Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Most Machine Learning models are built to predict the likelihood of a Stroke. However, in this project we aim to implement Machine Learning models to predict the causes of fatality post an episode of stroke, given the various medical and physical parameters of the patient.

To implement the model, we apply both Unsupervised and Supervised machine learning methodologies on the patients data available. Initially the dataset is analysed and a set of crucial features / attributes are produced, that can make accurate predictions. Crucial features that determine the outcome can be identified through the use of Principle Component Analysis.

Unsupervised Learning algorithms like K- Means Clustering is applied to patient profiles to classify them into several clusters that indicate various causes of fatality.

1.1 Relevance of the Project

Strokes being the leading cause of fatality globally, being able to diagnose the conditions earlier would prove to be life saving or least reduce it's effects on the patient. Using traditional Machine Learning models, it is possible to predict the occurrence of a Stroke given certain medical parameters of a subject. However, the cause of the fatality post the stroke is not known.

In this project, we aim at implementing Machine Learning models that would aid in pointing out the cause of fatality post an ischemic stroke. This would in turn help the diagnosis process by providing a more accurate cause and enable better treatment to reduce the effects of the stroke thus improving the patient's life expectancy.

1.2 Problem Statement

The goal of the project is to apply machine learning techniques and principles over large existing datasets to effectively predict the most probable life threatening risks that may follow the first stroke attack.

Most studies apply machine learning to predict the risk of having a stroke but not so much on the likely outcomes of the patients that do survive the initial stroke attack.

We apply both Supervised and Unsupervised machine learning methodologies to patient profile data.

Crucial features in determining the outcome can be identified using Principle Component Analysis (PCA)

Unsupervised learning principles such as K-Means Clustering can be applied to groups of individuals into canonical patient profiles.

Appending data on cause of death, we can then gain insight on the most likely cause of death for a new patient fitting any one of those profiles.

1.3 Scope of the Project

Patient profile data of over six years is retrieved from the International Stroke Trial Database. Analysis was started using 19,000 data points, but after refinement process to remove those data points with incomplete values, and those that did not succumb to the mentioned causes, as the goal is to predict the outcome of fatality is one were to die. After data pre-processing, over 4000 data points were considered and for each patient 14 crucial features were taken into consideration. Possible outcome of death DEAD1, DEAD2,… DEAD8 correspond to initial stroke, recurring ischemic, recurring haemorrhagic, pneumonia, heart disease, pulmonary embolism, other vascular, and non-vascular causes. To avoid comparing binary and continuous features, we set Age>65 and BP>150 to 1 and -1 otherwise.

1.4 Chapter Wise Summary

Most machine learning models aim at predicting the possibility of a stroke given certain medical features and parameters. This project however, aims to implement machine learning models that would predict the cause of the fatality post a stroke as affected a patient, using certain crucial identified features. The aim of this project would be to come up with significantly accurate methods that can predict the cause of fatality of a person affected by an ischemic stroke, such that on early detection suitable diagnosis can ensue.

To implement the models, both supervised and unsupervised machine learning models were used. Initially during the data pre-processing phase, the large dataset was reduced to a significantly smaller dataset by omitting all those data points that were incomplete or for those that the patient survived even after the stroke. The data set was also put through Principal Component Analysis to find the crucial features that affect the model's prediction directly.

Patient profile data of over six years is retrieved from the International Stroke Trial Database. Analysis was started using 19,000 data points, but after refinement process to remove those data points with incomplete values, and those that did not succumb to the mentioned causes, as the goal is to predict the outcome of fatality is one were to die. After data pre-processing, over 4000 data points were considered and for each patient 14 crucial features were taken into consideration. Possible outcome of death DEAD1, DEAD2,… DEAD8 correspond to initial stroke, recurring ischemic, recurring haemorrhagic, pneumonia, heart disease, pulmonary embolism, other vascular, and non-vascular causes. To avoid comparing binary and continuous features, we set Age>65 and BP>150 to 1 and -1 otherwise.

CHAPTER 2

LITERATURE SURVEY

The Research paper that was referred is:

Machine Learning for Predicting Delayed Onset Trauma Following Ischemic Stroke.

2.1 Overview

In this paper, we apply both unsupervised and supervised machine learning methodologies to patient profile data. First we will demonstrate :

(i) that features from differential diagnoses and medical interviews can be used in building classifiers that discriminate between likely outcomes of fatality.

(ii) Crucial features in determining outcome can be identified through Principle Components Analysis (PCA).

(iii) Unsupervised learning principles such as K-Means Clustering can be applied to group individuals into canonical patient “profiles”.

Appending data on cause of death, we can then gain insight on the most likely cause of death for a new patient fitting one of these profiles.

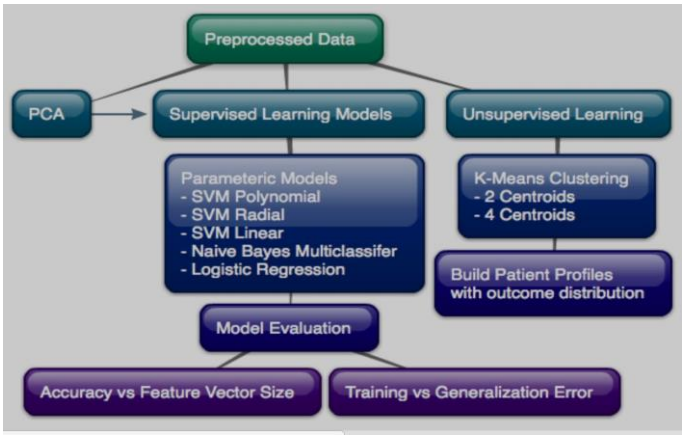


Fig 2.1 Outline of tools used in study

2.2 Machine Learning–Based Model for Prediction of Outcomes in Acute Stroke

2.2.1 Abstract

The prediction of long-term outcomes in ischemic stroke patients may be useful in treatment decisions, as well as in managing prognostic expectations. Several prognostic scoring systems have been developed for this purpose.

1. In light of recent advances in machine learning, application of the technique in the medical field has yielded promising results.

2. The complex and unpredictable nature of human physiology has, in many circumstances, proven to be better described by the machine learning algorithms. Unlike the traditional predictive models that use selected variables for calculation, machine learning techniques can easily incorporate a large number of variables, as all calculations are performed using a computer.3 These characteristics make machine learning techniques suitable for the medical field. In stroke, machine learning techniques are increasingly used in various areas including outcome prediction after endovascular treatment.4,5 With consideration of its expected impact on ischemic stroke management, we developed models using machine learning techniques to predict long-term stroke outcomes. We then compared the predictability to the Acute Stroke Registry and Analysis of Lausanne (ASTRAL) score, which is a wellknown prognostic model

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

For the implementation and analysis of the problem statement, we used several datasets which were presented in the .CSV format (Comma Separated Values). The use of some integrated development environments was also used. Environments like Jupyter Notebook supported by Anaconda.

These Integrated Development Environments and algorithms can be run across various operating systems that support Anaconda framework. However for the implementation we chose to use the Microsoft Windows 10 operating system.

The software and hardware requirements are as follows

3.1 Hardware Specifications

Processor	:	Core 2 duo or above
Hard Disk	:	5GB
RAM	:	2GB

3.2 Software Specifications

Operating System	:	Windows7/8/10 , Linux, MacOS
Dataset	:	The International Stroke Trial Database. Csv file

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

To visualize and implement the models, several large datasets were used for the study. All the datasets were presented as .CSV format (Comma Separated Values).

4.1 Interpretation of the Dataset

4.1.1 General Stroke dataset:

The General Stroke dataset collected from the Stroke Trials dataset is a general dataset presenting various features which help in predicting whether or not a stroke would occur in a given patient. The dataset presents values for various observed features collected from previous patients and testing procedures.

The features presented in the dataset include age, sex, profession type, presence of high blood pressure, presence of high blood glucose levels, stress levels, presence of heart disease, type of residential setting, body mass index, smoking status of the subject and lastly whether or not a stroke was observed in the subject.

4.1.2 International Stroke Trials Dataset

This dataset includes individual patient data from the International Stroke Trial (IST), one of the largest randomised trials ever conducted in acute stroke, available for public use, to facilitate the planning of future trials and to permit additional secondary analyses.

The IST dataset includes data on 19 435 patients with acute stroke, with 99% complete follow-up. Over 26.4% patients were aged over 80 years at study entry. Background stroke care was limited and none of the patients received thrombolytic therapy.

The dataset includes the following baseline data: age, gender, time from onset to randomisation, presence or absence of atrial fibrillation (AF), aspirin administration within 3 days prior to randomisation, systolic blood pressure at randomisation, level of consciousness and neurological deficit. The deficits were classified as one of the Oxfordshire Community Stroke Project (OCSP) categories: total anterior circulation syndrome (TACS), partial anterior circulation syndrome (PACS), posterior circulation syndrome (POCS) and lacunar syndrome (LACS). We extracted events within 14 days on: the occurrence of recurrent stroke, pulmonary embolism, and death (date and cause of death). At 6 months we extracted: degree of recovery, place of residence and current use of antiplatelet or anticoagulant drugs and death (date and cause of death). The cause of death was classified as: due to initial stroke, recurrent ischaemic stroke, recurrent haemorrhagic stroke, pneumonia, coronary artery disease, pulmonary embolism, other vascular cause or a nonvascular cause. Patients were assigned to one of 6 categories according to the place of residence at 6 months following stroke: own home, relatives home, residential care, nursing home, other hospital departments or unknown. The variables extracted are listed with a brief description of each in Tables 1, 2 and 3. Nineteen thousand four hundred and thirty five patients from 467 hospitals in 36 countries were randomised within 48 hours of symptoms onset, of whom 13020 had a CT before randomisation, 5569 were first scanned after randomisation and 846 were not scanned at all. Five thousand one hundred thirty two (26.4%) were aged over 80 years at study entry. Given that 5569 patients were first scanned after randomisation, and 846 were not scanned at all, the 'final diagnosis' is somewhat imprecise. However, since the analysis was by intention to treat, all participants were retained in the analysis, irrespective of the final diagnosis. The numbers of patients with each final diagnosis are given in Table 4. Whilst the 'final diagnosis variable' is of some interest, it may be influenced by events occurring after randomisation, so for any future analyses, the least biased assessment of the patient characteristics is that recorded at baseline, before randomisation.

4.2 System Architecture and Design

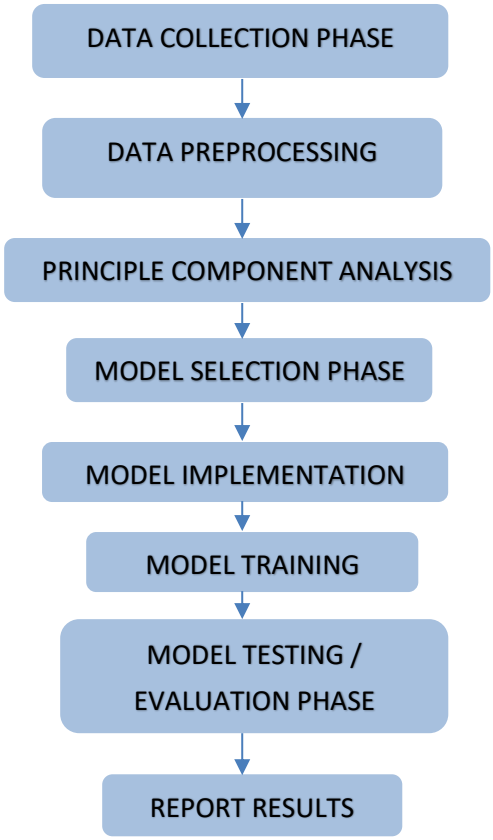


Fig 4.1 Workflow Followed

The above diagram illustrates the workflow followed throughout the project, for all Machine Learning models implemented.

1. **Data Collection Phase :** The first stage of the workflow is the data collection phase. In the data collection phase the datasets were extracted and collected from various verified sources and collated together for the analysis phase.

2. **Data Pre-processing Phase:** The datasets collected and collated in the previous stage are then analyses for null values and other errors. The pre-processing stage makes sure to overcome all these errors and clean the dataset in manner that is suitable for applying machine learning techniques.

3. **Principle Component Analysis:** In this phase the cleaned dataset is used. This stage focuses on finding a subset of features from the list of all the

features present in the dataset. This subset of features are chosen as they contribute the most towards the working of the machine learning models. It is also necessary to short list these features from the list of all the features as most machine learning models find it hard to handle to a lot of features, especially if those features do not contribute towards the accurate working of the model. Pushing too many irrelevant features into the machine learning algorithm will diminish its accuracy thus giving us poorer results.

4. **Model Selection Phase:** This phase follows after the phase of finding the subset of features that contribute most to the efficient working of a machine learning model. In this phase we analyse the problem statement and the goals we are looking to achieve and make relevant choices, as to which machine learning models would be best to implement inorder to get the results we desire. This stage involves careful thinking and analysis before choosing the relevant models.

5. **Model Implementation:** In this phase the models selected in the previous phase are put into implementation. The models are coded using python and Jupyter Notebook backed by Anaconda framework.

6. **Model Training:** In this phase the implemented model is put into training. The dataset is divided into a testing and training set and the training set is used to train the machine learning model.

7. **Model Testing / Evaluation Phase :** The testing dataset is used to test the accuracy of prediction made by the trained model in the previous phase. This accuracy helps us assess the model's efficiency over unseen or new data points.

8. **Report Results:** In this phase, the results are collated and reported.

CHAPTER 5

IMPLEMENTATION

The implementation of the various models for the analysis of the problem statement included various steps. The first stage of the implementation was preceded by a phase of literature analysis and background data analysis.

5.1 Data

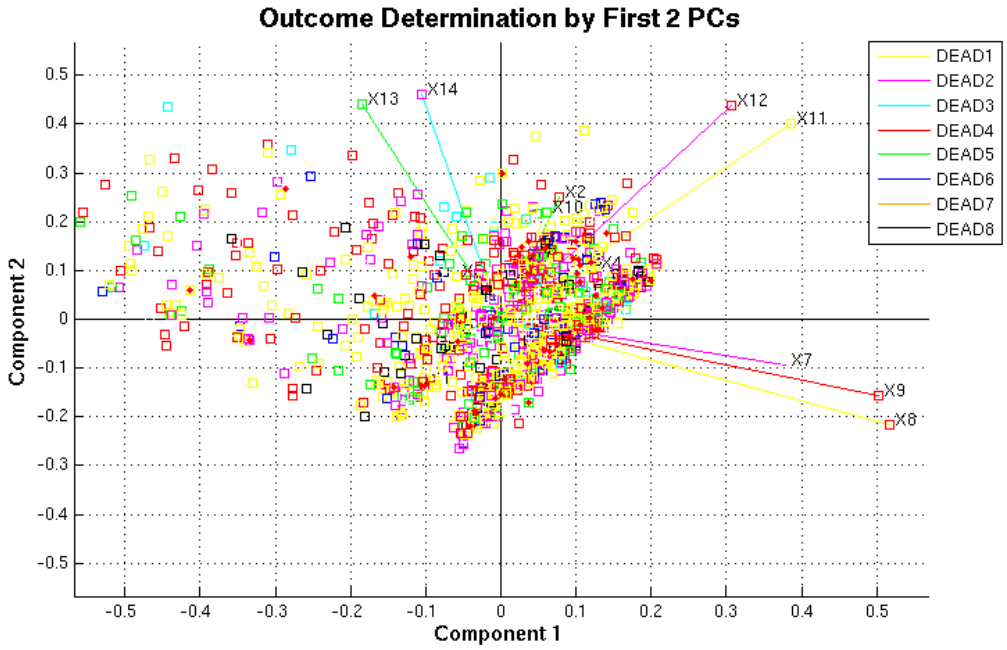
Patient profile data was obtained from a 6-year trial retrieved from the International Stroke Trial Database. We started with over 19,000 data points, but performed a refinement process to remove patients with incomplete patient profile, and those that remain alive, since our goal is predicting outcome of death if one were to die. In the end we generated preprocessed data set of ~4000 patients. For each patient, there are 14 features that we are focusing on: sex, age, atrial fibrillation, visible infarct under CT, aspirin, systolic blood pressure, facial, arm, leg deficit, dysphasia, hemianopia, visuospatial disorder, brainstem signs, and other deficits. Possible outcome of death DEAD1, DEAD2, … DEAD8 correspond to initial stroke, recurring ischemic, recurring hemorrhagic, pneumonia, heart disease, pulmonary embolism, other vascular, and non-vascular causes. To avoid comparing binary and continuous features, we set Age>65 and BP>150 to 1 and -1 otherwise.

Feature	Metric	Description
SEX	1, 0	Gender of patient (Male = 1, Female = 0)
AGE	1, -1	Age in years (>=65 = 1, <65 = -1)
RATRIAL	1, 0	1 = Presence of atrial fibrillation
RVISINF	1, 0	1 = Infarct visible on CT imaging
RASP3	1, 0	1 = Aspirin taken within 3 days of randomization
RSBP	1, -1	1 = Systolic blood pressure >=150
RDEF1	1,0,-1	Face Deficit (Yes, No, Can't Access)
RDEF2	1,0,-1	Arm/hand Deficit (Yes, No, Can't Access)
RDEF3	1,0,-1	Leg/foot Deficit (Yes, No, Can't Access)
RDEF4	1,0,-1	Dysphasia (Yes, No, Can't Access)
RDEF5	1,0,-1	Hemianopia (Yes, No, Can't Access)
RDEF6	1,0,-1	Visuospatial Disorder (Yes, No, Can't Access)
RDEF7	1,0,-1	Brainstem/Cerebellar signs (Yes, No, Can't Access)
RDEF8	1,0,-1	Other Deficits (Yes, No, Can't Access)

Fig 5.1 Feature Description and Quantification

5.2 Feature Selection

Principal Components Analysis (PCA) maps data of original feature dimension *n* to smaller dimension *k*. These new principal components or PCs are linear combinations of original features that carry maximal variance when data is projected onto it. Original data set is represented by only 14 features, which happen to be predictive of stroke risk according to literature. Therefore most algorithms were done on full feature dimension. Feature selection techniques such as PCA, however, can give intuition on the most important factors in determining patient outcome.



5.3 K- Means

K-Means clustering algorithm was implemented in Python using Anaconda with 2 and 4 centroids. K-means is an unsupervised learning algorithm, which clusters patient profiles into k

centroids by minimizing weighted norms between data point and centroid position. We represented each patient with ***pj*** representing a 14-dimensional vector containing profile information, and ***μj*** denotes the mean of points in cluster ***Gi.***

$$argmin_G \sum_{i=1}^{k} \sum_{p_j \in G} || p_j - \mu_j ||^2 \qquad (1)$$

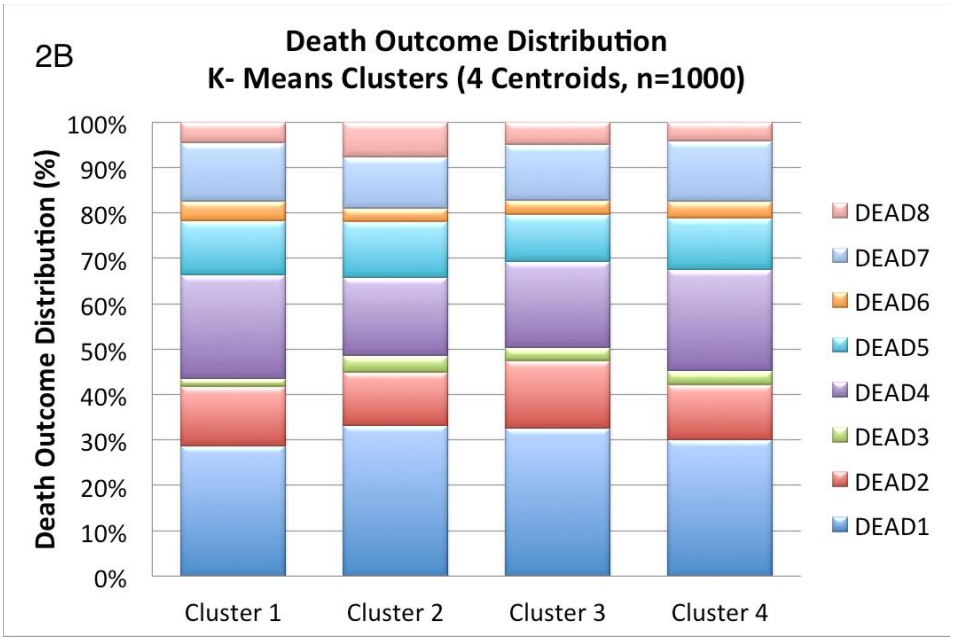


Fig 5.2 Death outcome distribution (4 centroids)

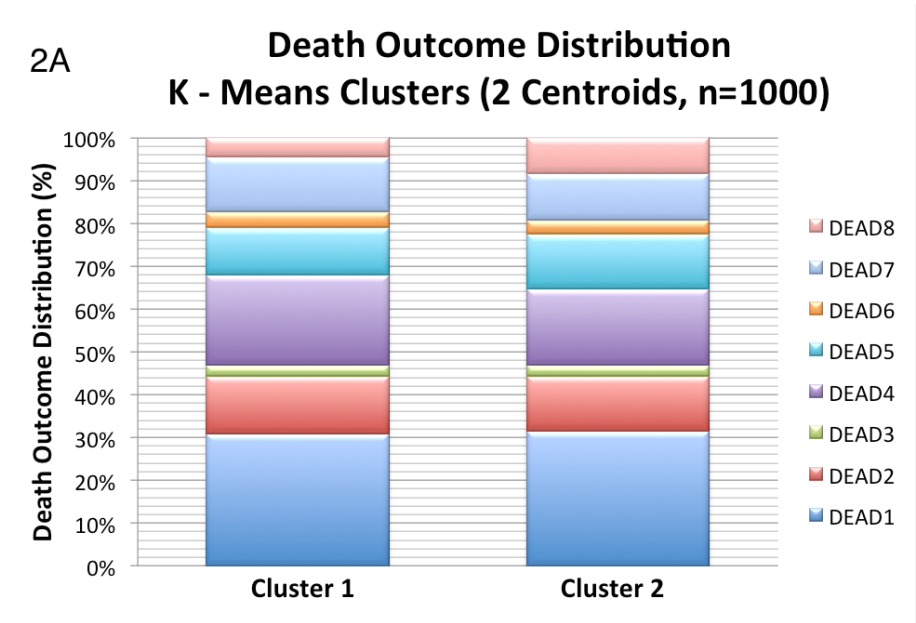


Fig 5.3 Death Outcome Distribution (2 centroids)

5.4 Naïve Bayes

Multiclass Naïve Bayes was implemented in python based on frequency of observed features values and corresponding outcomes. Laplace smoothing of smoothing parameter = 1.0 was applied. Assumption of independence and Gaussian distribution was made despite some features having high correlation (i.e. facial, arm, and leg deficit usually come together).

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem**.**

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad (2)$$

Let us take an example to get some better intuition. Consider the problem of playing golf. The dataset is represented as below.

We classify whether the day is suitable for playing golf, given the features of the day. The columns represent these features and the rows represent individual entries. If we take the first row of the dataset, we can observe that is not suitable for playing golf if the outlook is rainy, temperature is hot, humidity is high and it is not windy. We make two assumptions here, one as stated above we consider that these predictors are independent. That is, if the temperature is hot, it does not necessarily mean that the humidity is high. Another assumption made here is that all the predictors have an equal effect on the outcome. That is, the day being windy does not have more importance in deciding to play golf or not.

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

According to this example, Bayes theorem can be rewritten as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \qquad (3)$$

The variable **y** is the class variable(play golf), which represents if it is suitable to play golf or not given the conditions. Variable **X** represent the parameters/features.

X is given as,

$$X = (x1, x2, x3, \ldots\ldots, xn) \qquad (4)$$

Here x1,x2….xn represent the features, i.e they can be mapped to outlook, temperature, humidity and windy. By substituting for **X** and expanding using the chain rule we get,

$$P(y|x1,\dots,xn) = \frac{P(x1|y)\dots P(xn|y)P(y)}{P(x1)P(x2)..P(xn)} \qquad (5)$$

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remain static. Therefore, the denominator can be removed and a proportionality can be introduced.

In our case, the class variable(**y**) has only two outcomes, yes or no. There could be cases where the classification could be multivariate. Therefore, we need to find the class **y** with maximum probability**.**

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \qquad (6)$$

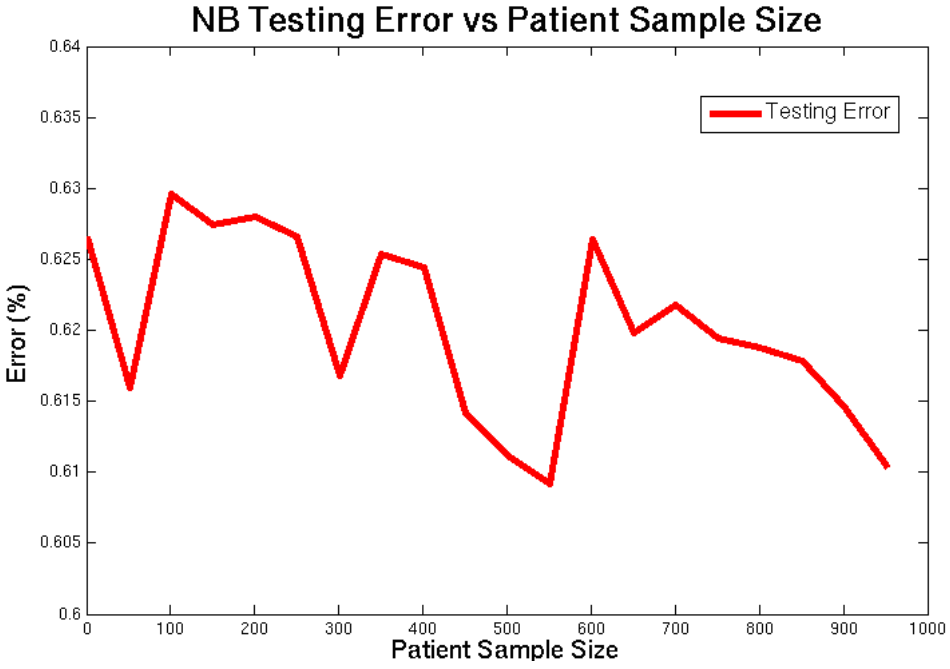


Fig 5.3 NB Testing Error vs Patient Sample Size

5.5 K – Nearest Neighbours (KNNs)

KNN classifies each new test patient based on the most popular labeling of k-nearest neighbors, as determined by the weighted norm of Euclidean distances. For our model, we have K = 3 since it is large enough to reduce noise on classification but avoids making boundaries between classes indistinguishable.

The KNN Algorithm

1. Load the data
2. Initialize K to your chosen number of neighbors

3. For each example in the data

3.1 Calculate the distance between the query example and the current example from the data.

3.2 Add the distance and the index of the example to an ordered collection

4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances

5. Pick the first K entries from the sorted collection

6. Get the labels of the selected K entries

7. If regression, return the mean of the K labels

8. If classification, return the mode of the K labels

Choosing the right value for K

To select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before.

Here are some things to keep in mind:

1. As we decrease the value of K to 1, our predictions become less stable. Just think for a minute, imagine K=1 and we have a query point surrounded by several reds and one green (I'm thinking about the top left corner of the colored plot above), but the green is the single nearest neighbor. Reasonably, we would think the query point is most likely red, but because K=1, KNN incorrectly predicts that the query point is green.

2. Inversely, as we increase the value of K, our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point). Eventually, we begin to witness an increasing number of errors. It is at this point we know we have pushed the value of K too far.

3. In cases where we are taking a majority vote (e.g. picking the mode in a classification problem) among labels, we usually make K an odd number to have a tiebreaker.

Advantages

1. The algorithm is simple and easy to implement.
2. There's no need to build a model, tune several parameters, or make additional assumptions.
3. The algorithm is versatile. It can be used for classification, regression, and search (as we will see in the next section).

Disadvantages

1. The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

5.6 Support Vector Machines

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

Cost Function and Gradient Updates

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \quad (7)$$

$$c(x, y, f(x)) = (1 - y * f(x))_+ \quad (8)$$

Hinge loss function (function on left can be represented as a function on the right)

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+ \tag{9}$$

Loss function for SVM

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \left(1 - y_i \langle x_i, w \rangle\right)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases} \tag{10}$$

Gradients

When there is no misclassification, i.e our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

$$w = w - \alpha \cdot (2\lambda w) \tag{11}$$

Gradient Update — No misclassification

When there is a misclassification, i.e our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w) \tag{12}$$

Gradient Update — Misclassification

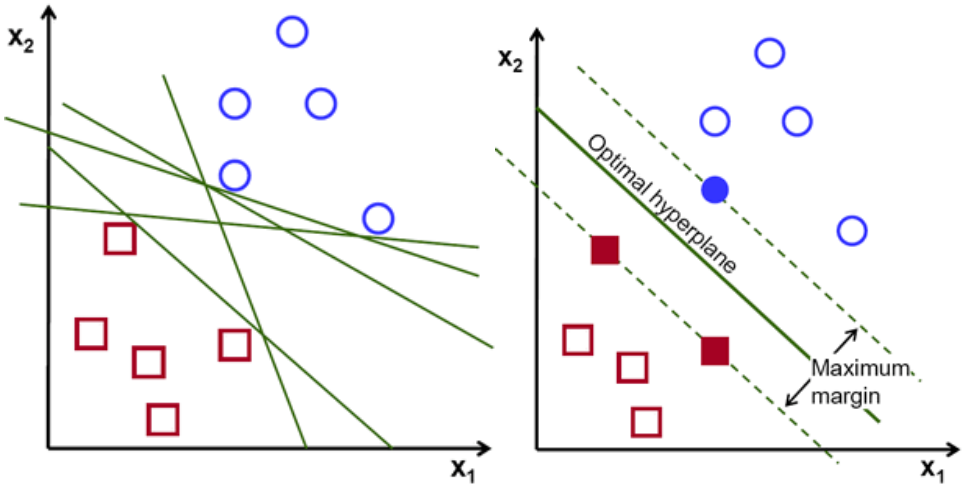


Fig 5.4 Support Vector Machines

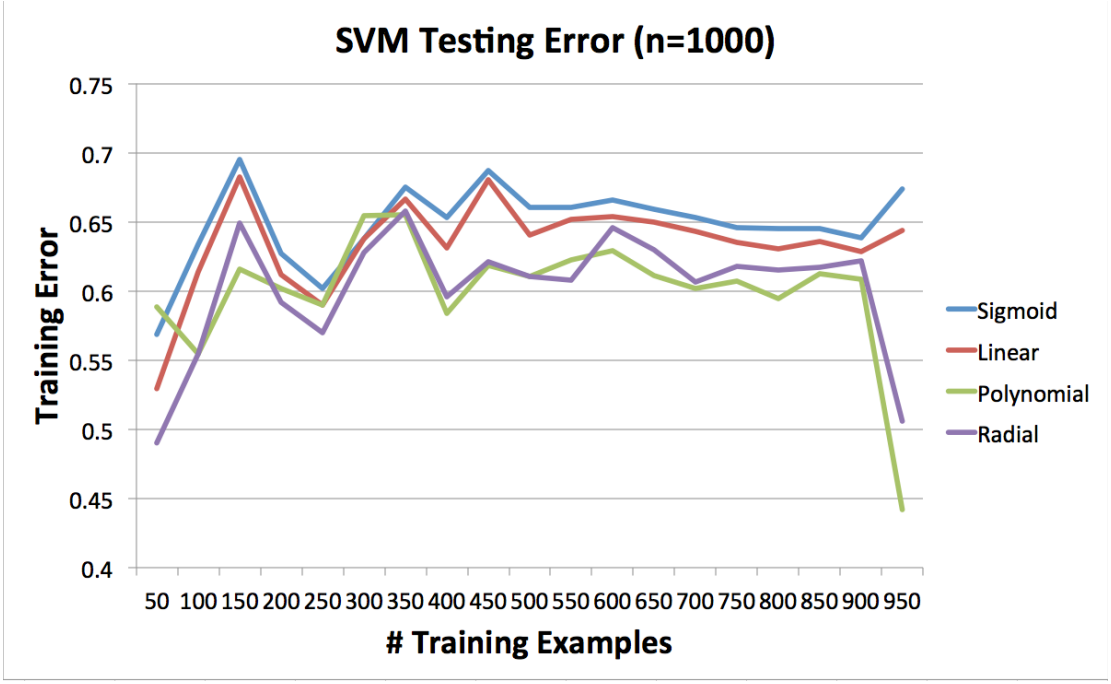


Fig 5.5 SVM Testing Error

5.7 Multinomial Logistic Regression (softmax)

The Multinomial Logistic Regression, is a supervised learning algorithm where output can take on arbitrary k outcome classes. It requires significantly more training time than Naïve Bayes since iterative algorithms are necessary in parameter estimation. Most of the computation was done with **mnrfit** function on MatLab. In order to build the multiclass model, we estimate $\theta 1, \theta 2, \ldots \theta k \in$ Rn+1 parameters, where θi vector stores coefficients of ith outcome for each n feature and intercept term. Probability of a patient being classified into certain outcome equals:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{\exp(\theta_j^T x^{(i)})}{\sum_{1 \le l \le k} \exp(\theta_l^T x^{(i)})}\right] \quad (13)$$

We use the cost function as defined by and determine corresponding theta parameters.

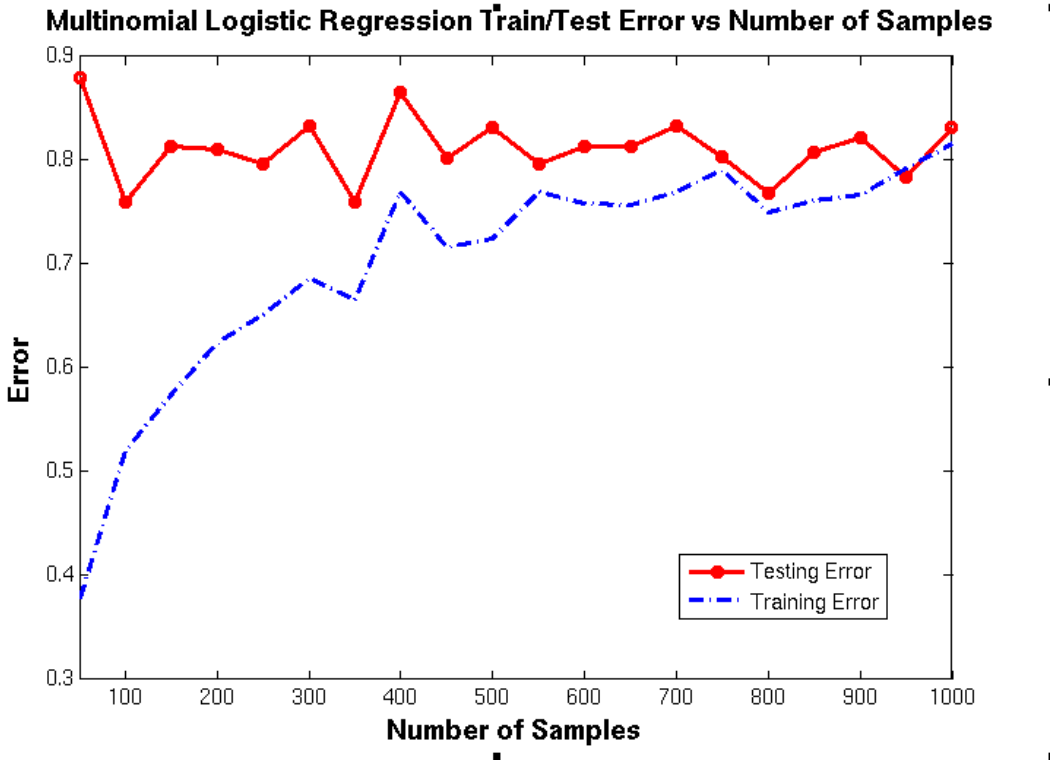


Fig 5.6 Multinomial Logistic Regression Train/Test Error

CHAPTER 6

RESULTS AND DISCUSSION

In this article, we demonstrated how Naïve Bayes and support vector machines (polynomial) leads to maximal outcome prediction accuracy of 40 %, and 56% respectively in classifying 8 different death outcomes following initial ischemic trauma, using 14 crucial features.

Comparing to the average predictive accuracy following randomization (12.5%), our best algorithm achieves up to a 4.5 fold prediction accuracy increase, which carries immense clinical utility in improving a patient's chance of survival and quality of life.

With the combination of unsupervised learning algorithms such as K-Means and supervised outcome data, we also built canonical "profiles" of the most common patients doctors are likely to encounter following initial stroke attack. Each one represents a corresponding distribution of death outcomes.

New patients can then be fitted into the most representative profile and plan of action will be taken to minimize chances of the most likely ensuing risks.

6.1 Feature Selection

Based on literature studies on predictive factors of ischemic stroke, it is well known that features such as age, sex, blood pressure, infarct size, and craniofacial deficits are very important in determining patient outcome. Therefore, our initial hypothesis was that running our learning algorithms on full feature dimension produces lowest generalization error, which is true. Although our data is not characterized by large feature dimension, principal component analysis could still reveal some important relationships between different features as well as the predictive value each individual feature has on outcome of fatality. As shown in Figure - 8, over 99% of all variance is accounted for just by the first two principle components. Features: age, sex, and blood pressure were quite indicative of death outcome.

Finally, there seems to be high correlation between arm/hand deficit and leg/foot deficit as well as hemianopia and visuospatial disorder.

6.2 Unsupervised Learning

In this project, we used K-Means clustering algorithm as both an exploratory tool to determine underlying structure within data points as well as a way to generate canonical "patient profiles" for important clinical applications.

Resultant clustering representation confirms the presumed idea that gender is one of the most predictive measures for eventual outcome of death. This is evident in both the case with 2 centroids and 4 centroids. From Figure - 2A, we gain insight on the characteristics of the common stroke

victims. They are profile 1: Male, age 65 or older, high-blood pressure, with facial, arm, leg deficit, and signs of dysphasia. And profile 2: Female, age 65 or older, high blood pressure, with arm/hand deficit and dysphasia. There were 844 patients corresponding to profile 1, and 156 corresponding to profile 2. After appending supervised data for these patients, calculating distribution of death outcome, and conducting a 2-sample *t*-Test to compare the differences in mean patients falling into each outcome

category, we determined *p*-values for differences in outcome distribution. From Table - 3, it is evident how profile 1 patients have a statistically significant higher chance of eventually dying of pneumonia/immune system failure or other vascular causes: p = 3.90E-04 and p = .0423 respectively. Similarly profile 2 patients face much higher risks of coronary heart disease and non-vascular causes of death than former candidates: p = .0552 and p = 3.09E-05 respectively. Figure – 2B illustrates K-Means algorithm applied in generating four profiles of distinct

outcome distribution.

6.3 Supervised Learning

All results from learning algorithms were performed on full feature set, which led to minimization of training and generalization error. We first implemented a Multiclass Naïve Bayes classifier as our baseline supervised, parametric model. As seen in Figure - 3, the model performed fairly well, achieving approximately 40% testing accuracy (60% error), considering there were 8 outcomes to choose from. Comparing this to percentage accuracy of random decision 12.5%, we achieved a greater

than 3-fold prediction accuracy increase. As the number of classification outcomes increase, generalization error generally rises as well. Furthermore, in the context of predicting likely outcomes following initial ischemic attack, even minor increases in prediction accuracy carries high clinical utility. There are future steps to take in reducing error reducing error percentages. It was also discovered that most death outcomes corresponded to initial stroke, pneumonia, and non-vascular causes (DEAD1, DEAD4, and DEAD8 respectively) and our Naïve Bayes model almost exclusively predicted those three outcomes, thus resulting in a fairly high generalization error.

The next parametric supervised learning algorithm we explored was the multiclass logistic regression (SoftMax).

As seen in Figure – 5, the algorithm's generalization error was quite high at approximately 78% for all sample sizes.

Training error starts relatively low at 37% and asymptotically increases to match generalization error as sample size increases towards n=1000. This larger error value was largely due to failure upon converging on true theta parameters during training for large sample sizes with **mnrfit** MatLab software. Overall, multinomial logistic regression can only serve as a reference point, and has less capability in outcome prediction.

Taking a different approach with the non-parametric KNN classifier (with K=3), we achieved a 34% testing accuracy (67% generalization error), which is slightly worse than Naïve Bayes. Figure - 6 depicts the significantly smaller training error.

Finally, to gain more insight into the data, we used support vector machines with multiple kernel options (Figure - 4). Our SVM model using polynomial kernels provided the best accuracy of 56% when using at least 1000 training examples. This leads to the optimal 4.5 fold increase in prediction accuracy. Radial kernels performed quite well as well with 49% accuracy. Finally, linear and sigmoid kernels performed with only 37% and 36% accuracy respectively. Given that linear kernels had relatively poor performance, we confirm the fact that our data is not linearly separable as indicated previously by PCA results. For the polynomial and radial

SVMs, the generalization error decreases with increasing training examples, specifically past the threshold of *n*=1000. Our SVM algorithm was limited because of the extremely high feature vector dimensions, resulting in over-fitting and inaccurate generalization to other data. Though this issue could have been mediated by extensive parameter adjustment or feature reduction, we chose not to apply these techniques because of the complexity of our data, the significance of all of our utilized features, and already having reduced our error significantly.

Model Testing Error N Iterations

Naïve Bayes 0.60 1000

SoftMax 0.80 1000

KNN 0.66 1000

SVM (Sigmoid) 0.68 1000

SVM (Linear) 0.63 1000

SVM (Polynomial) 0.44 1000

SVM (Radial) 0.51 1000

6.4 Relationship between Initial Blood Pressure and type of stroke

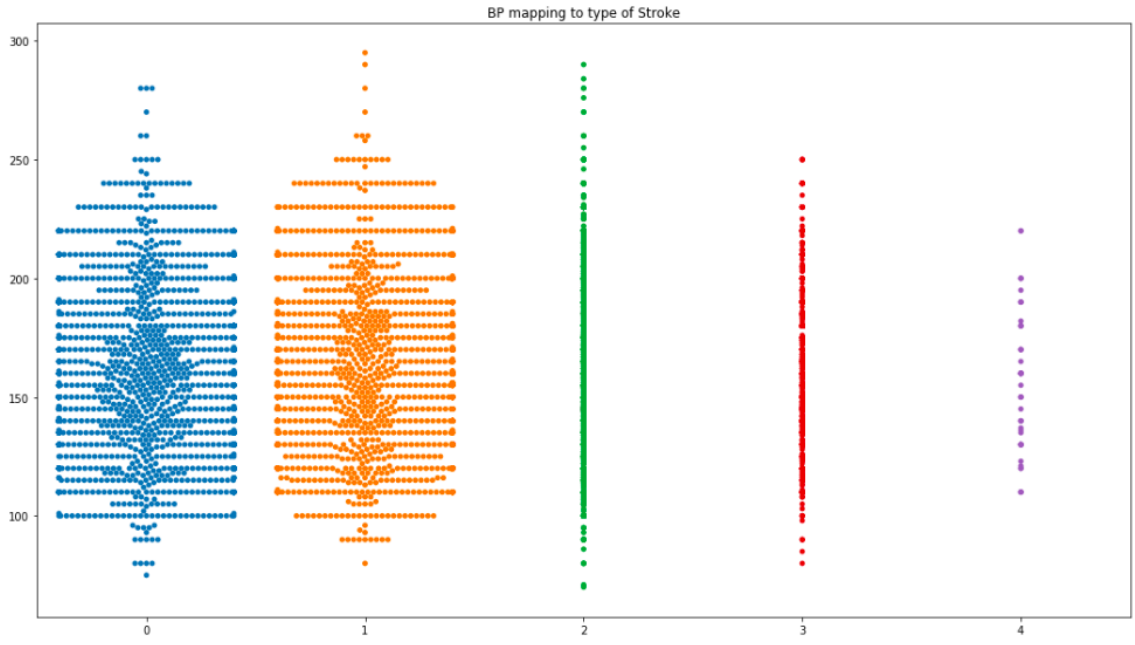


Fig 6.1 Relationship between Initial BP and type of Stroke

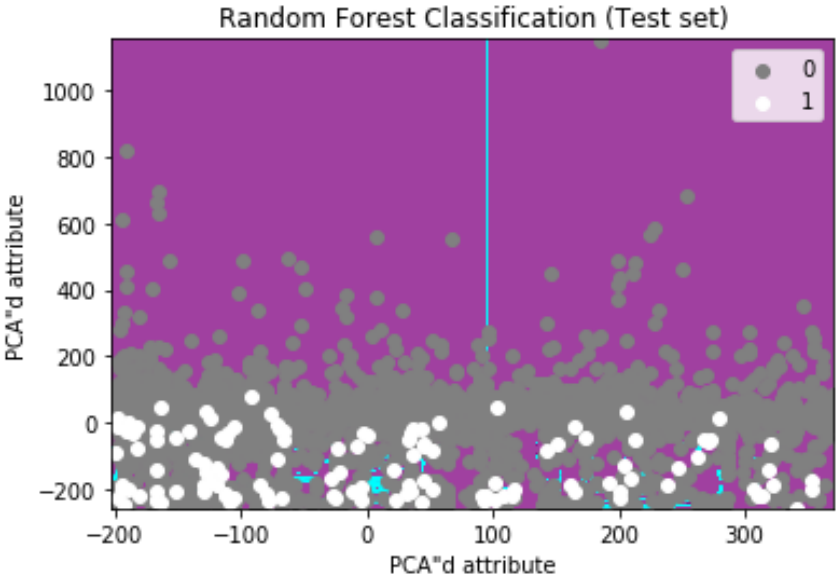
- Most prevalent types of strokes was found to be TACS and PACS.
- Subjects having Diastolic BP of ~150 have more chances of suffering from TACS, however PACS cannot be completely dismissed.

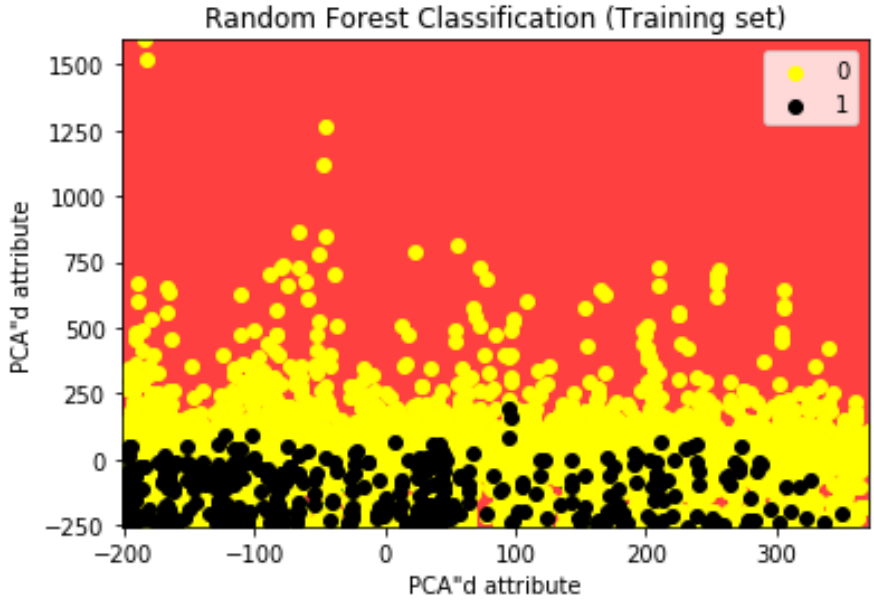
6.5 Decision Tree Classifier

```
Precision:  0.2857142857142857
Accuracy:  0.9829493087557604
Classification report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      8538
           1       0.29      0.03      0.05       142

    accuracy                           0.98      8680
   macro avg       0.63      0.51      0.52      8680
weighted avg       0.97      0.98      0.98      8680


Confusion matrix:
[[8528   10]
 [ 138    4]]
```

- Decision Tree Classifier made for the stroke dataset returned the following accuracy.
- We are also working on improving the model by applying XGBoost methods to the existing decision tree

6.6 Random Forest Model





- Random Forest Model was built and tests results are as follows.
- The Goal of building the random forest model was to improve the achieved accuracy of the Decision Tree made in the previous attempts of the study.
- Random forests predict the occurrence of the fatality than the type of fatality itself.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

In this article, we demonstrated how Naïve Bayes and support vector machines (polynomial) leads to maximal outcome prediction accuracy of 40 %, and 56% respectively in classifying 8 different death outcomes following initial ischemic trauma, using 14 crucial features.

Comparing to the average predictive accuracy following randomization (12.5%), our best algorithm achieves up to a 4.5 fold prediction accuracy increase, which carries immense clinical utility in improving a patient's chance of survival and quality of life. With the combination of unsupervised learning algorithms such as K-Means and supervised outcome data, we also built canonical "profiles" of the most common patients doctors are likely to encounter following initial stroke attack. Each one represents a corresponding distribution of death outcomes.

New patients can then be fitted into the most representative profile and plan of action will be taken to minimize chances of the most likely ensuing risks.

FUTURE SCOPE

Future work involves discovering predictive value of individual features and their relationship/correlation strength with each other. We also plan on extending our classification model to patients that do not die immediately after initial ischemic infarction. Furthermore, our models can take into account the likelihood of outcomes at different timespans after initial attack (14 days, 6 months, 1 year, etc…) such that physicians can gain intuition on optimal treatment plans based on particular stage of patient recovery. Finally, many more related studies may be done using similar learning tools but starting with different sorts of initial trauma (i.e. hemorrhagic stroke, thrombotic stroke, transient ischemic attack, …) as well as discovering the role of pre and post-conditioning factors on survival rates.

REFERENCES

[1]. Prediction of Delayed onset of trauma in ischemic stroke- Anthony Ma, Gus Liu – department of computer science, Stanford University.

[2]. "Understanding Stroke Risk." Understanding Stroke Risk. N.p., n.d. Web. Sandercock, Peter Ag, Maciej Niewada, and Anna Członkowska. "The International Stroke Trial Database." Trials 12.1

[3]. "Understanding Stroke Risk." *Understanding Stroke Risk*. N.p., n.d. Web. 24 Nov. 2014.

[4]. "Types of Stroke." Johns Hopkins, n.d. Web. 24 Nov. 2014.

[5]. Ishikawa, H., N. Tajiri, J. Vasconcellos, Y. Kaneko, O. Mimura, M. Dezawa, and C. V. Borlongan. "Ischemic Stroke Brain Sends Indirect Cell Death Signals to the Heart." *Stroke* 44.11 (2013): 3175-182. Web.