**BLOOD DONATION ANALYSIS**

*A Thesis Submitted*

*In the partial fulfillment of the requirements for*

*The award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**



**By**

**Tarique Hussain (17281A05B2)**

PROJECT GUIDE

**Mr. N. Gowtham Kumar**

**Assistant Professor**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE**

(Approved by AICTE, New Delhi and Affiliated to JNTU, Hyderabad, T.S., Accredited by NAAC with ‘B++’ Grade) Singapur, Huzurabad, Karimnagar, Telangana, India-505468.

(2020-2021)

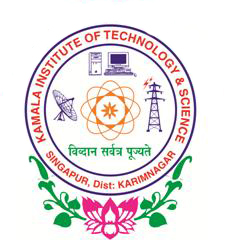
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE**

(Approved by AICTE, New Delhi and Affiliated to JNTU, Hyderabad, T.S., Accredited by NAAC with ‘B++’ Grade)

Singapur, Huzurabad, Karimnagar, Telangana, India – 505468.

**2020– 2021**

****

**CERTIFICATE**

This is to certify that **Md. Tarique Hussain (17281A05B2)** of the IV B. Tech (CSE) 2nd -semester has satisfactorily completed the dissertation work for Industry oriented major project entitled **“BLOOD DONATION ANALYSIS”** towards the partial fulfillment of B. Tech degree in this academic year i.e., 2020-2021

**Project Guide Project Coordinator**

**Head of the Department External Examiner**

**INDEX**

|  |  |
| --- | --- |
| **ACKNOWLEDGEMENT** | i |
| **ABSTRACT** | ii |
| **LIST OF FIGURES** | Iv |
| [LIST OF TABLES v](#_TOC_250006) | V |
| [CHAPTER 1: INTRODUCTION](#_TOC_250004) | 1 |
| **CHAPTER 2: SYSTEM ANALYSIS** | 2 |
| 2.1 Existing System | 2 |
| 2.2 Proposed System | 2 |
| 2.3 Functionalities of each Modules | 3 |
| 2.4 Software and Hardware Requirements | 7 |
| **CHAPTER 3: SYSTEM DESIGN**  3.1 Introduction | 8 |
| 3.2 Block diagram  3.3 Data Dictionary  3.4 UML diagram  3.4.1 Use case diagram  3.4.2 Sequence diagram  3.4.3 Collaboration diagram  3.4.4 Activity diagram | 8  9  15  16  17  18 |
| **CHAPTER 4: IMPLEMENTATION** | 19 |
| 4.1. Logistic Regression  4.2 Support Vector Machine  4.3 Random Forest  4.4 Decision Tree | 19  22  24  27 |
| **CHAPTER 5: TESTING** | 31 |
| **CHAPTER 6: RESULTS** | 32 |
| **CHAPTER 7: CONCLUSION** | 36 |
| **CHAPTER 8: FUTURE SCOPE AND ENHANCEMENT** | 37 |
| **CHAPTER 9: BIBLIOGRAPHY**  **CHAPTER 10: APPENDIX** | 38  40 |
|  |  |
|  |  |

**ACKNOWLEDGEMENT**

The success of any course depends mostly on the teachers who teach us. Only good teaching can interpret the syllabus and produce desirable changes and competent citizens. This one was a team effort and many people whose names do not appear on the cover deserve credit. First, we thank God almighty for his manifold mercies in carrying out our project successfully.

I heart fully thank our principal, **Dr.K. SHANKER,** for providing all the resources in completing our project.

I sincerely thank our Head Of the Department of Computer Science and Engineering and Professor, **Dr. Lt. RAVINDRA BABU KALLAM**, for encouraging us in doing this project.

I thank our project coordinator Mrs**. Dr .D. SUNITHA,** Assistant Professor in CSE department for providing continuous support in completion of this project.

I thank our Internal Guide **Mr. N.Gowtham Kumar**, Assistant Professor in CSE

Department for providing continuous support in completion of this project.

I also thank other teaching and non-teaching staff of Kamala Institute of Technology and Science, for supporting us in every stage of major project, entitled **BLOOD DONATION ANALYSIS** I thank our Parents and Friends for their moral support throughout the project work that helped to strengthen our will.

## **Tarique Hussain (17281A05B2)**

# ****ABSTRACT****

With the proliferation of big data, the need for intelligent and automated systems has risen. This need is probably felt the most in the field of health care, especially in the area of blood transfusion, since they require supplies at the earliest. Currently, transfusion services are heavily manual in nature, which is not ideal. The rising demand for blood and the decline in donation rates has put a lot of strain on the blood donation supply chain. Hence, creating intelligent systems that can make decisions and improve communication across the supply chain is of great importance. In this project, we are going to give a general summary of the various machine learning techniques which have been applied to this domain and compare their accuracy and scores to find the best model.

# ****LIST OF FIGURES****

|  |  |
| --- | --- |
| Figure: 2.4.1 Dataset from source | 7 |
| Figure: 3.1 Block Diagram  Figure: 3.3.1 Use case diagram  Figure: 3.3.2 Sequence diagram  Figure: 3.3.3 Collaboration diagram  Figure: 3.3.4 Activity diagram | 9  15  16  17  18 |
| Figure: 4.1 Logistic Regression Model | 19 |
| Figure: 4.2 Support Vector machine model  Figure: 4.3 Random Forest model  Figure: 4.4 Decision Tree | 22  24  27 |
| Figure: 6.1 Gathering data | 32 |
| Figure: 6.2 Data description | 32 |
| Figure: 6.3 Box Plot | 33 |
| Figure: 6.4 Correlation between variables | 34 |
| Figure: 6.5 Result | 35 |
|  |  |

CHAPTER-1

**INTRODUCTION**

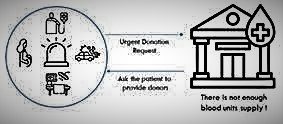
The first volunteer blood donation service was opened in 1921 since then the outreach for donors has been constant. The importance of blood donation is clear given that, there is no known substitute for human blood and its products and blood transfusions are a necessity for many patients ranging from someone who is fighting cancer, a sickle cell disease, a premature infant to someone that has suffered an accident. Blood donation stumbles upon several challenges. Its main challenge is that only 37% of this country’s population is eligible for donation and only 10% of this group are donating annually. The second biggest challenge is their shelf life, red blood cells need to be utilized within 35 to 42 days after donation to ensure the patient's safety. One of the ways to minimize the degree of impact these challenges present is to find a way to accurately predict blood donations. Data Science can utilize predictive analytics to process a given amount of information and analyze it to predict future blood donations. By predicting future blood donations, the healthcare system can better prepare and manage their resources. Blood donation organizations for example can organize staff schedules according to predicted donations as well as to present a stronger campaign for the lower donation months. On the other side of the spectrum the hospitals can better manage their resources by knowing when are they going to become available. Blood donation prediction models can benefit blood donation organizations as well as the patients in need.

**CHAPTER-2**

**SYSTEM ANALYSIS**

**2.1 Existing System:**

The donation of blood is very important because most often people needing blood do not receive it on time causing fatality. Such people include accidents, patients suffering from malaria or organ transplants. Extreme health conditions such as Leukemia and bone marrow cancer, where affected individuals experience sudden high blood loss and need an urgent blood supply and not providing it can lead to loss of life.

****

The problem with most existing systems that it does not cover the blood banks’ need by 100% of voluntary donations. These systems rely on patients who need blood to provide donors if the blood bank does not have enough blood units which can take a lot of time and effort.

**Disadvantages:**

* More Time is needed.
* More effort.
* High Cost

**2.2 Proposed System:**

The main goal of this proposal is to improve the blood bank system by reaching out to people more likely to donate through study some factors that affect their behavior by taking advantage of the ML algorithms then contact them. The proposed system will be done by developing a classifier using an ML algorithm that classifies people into more likely to donate blood and less likely to donate based on some factors.

In this project we are using four algorithms such as Logistic Regression, Support Vector Machines(SVM),Random Forest ,Decision Tree to get more accurate results. We Train each model and after testing we get the accuracy of them .The final step is to compare the scores and choose the model which gives best accuracy.



**The outcome of the proposed system** will improve the efficiency of the blood bank system by finding blood donors more accurately, and reduce costs by contacting the potential donors rather than contacting someone how is not willing to donate.

**Advantages:**

* Less Time is needed.
* Less or No Effort is needed.
* Cost is less.

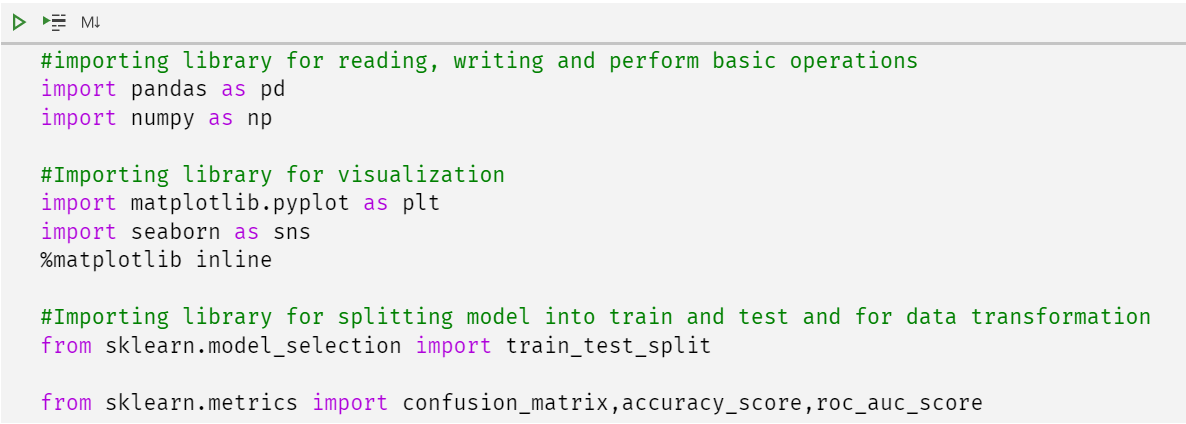
**2.3 Functionalities of each Module:**

The project contains the following main modules:

* Gathering of Data
* Data Exploration
* Feature Engineering
* Feature Transformation
* Implementation
* Comparing the ROC scores
* Results and Visualization

1. **Gathering of Data**

Firstly, we will define our libraries like NumPy, pandas, matplotlib and sklearn in the code. NumPy used for matrix multiplications, Pandas is used for data manipulation, Matplotlib for visualization of data and sklearn is for Machine learning Algorithms.







Next step is to import the blood prediction dataset in the code by using Pandas module. The blood prediction dataset contains features like Months since Last Donation, Number of Donations, Total Volume Donated, Months since First Donation and so on. After importing the data, we should check for the extreme values and missing values in all the features of the blood prediction dataset.

1. **Data Exploration**

**Data exploration** is the first step of data analysis used to explore and visualize data to uncover insights from the start or identify areas or patterns to dig into more. Using interactive dashboards and point-and-click data exploration, users can better understand the bigger picture and get to insights faster.

**Need of Data Exploration**

Data exploration can help businesses explore large amounts of data quickly to better understand next steps in terms of further analysis. This gives the business a more manageable starting point and a way to target areas of interest. In most cases, data exploration involves using data visualizations to examine the data at a high level. By taking this high-level approach, businesses can determine which data is most important and which may distort the analysis and therefore should be removed. Data exploration can also be helpful in decreasing time spent on less valuable analysis by selecting the right path forward from the start.

In our Project we are supplied with a dataset from the website drivendata.org. (https://www.drivendata.org/competitions/2/warm-up-predict-blood-donations/).We initially split the data into training and test sets using random selection. This allowed us to test the viability of the model on a known result set. Although we can determine internal model fit by using various measures, the best way to see how different models handle new information is to test them against data that we know the answers, which isn’t possible if we completely train our model on all known data. We went with a 80 to 20 split. Both sets contain four independent variables. The training set contains a dependent variable, which we are tasked with trying to predict in the test set. The independent variables are:

• **Months since Last Donation**: this is the number of months since this donor's most recent donation.

• **Number of Donations**: this is the total number of donations that the donor has made.

• **Total Volume Donated**: this is the total amount of blood that the donor has donated in cubic centimeters.

**• Months since First Donation**: this is the number of months since the donor's first donation.

The dependent variable in the training set is:

• **Made Donation**: this is the classifier on whether a given donor gives blood or not.

1. **Feature Engineering**

Feature engineering involves leveraging data mining techniques to extract features from raw data along with the use of domain knowledge. Feature engineering is useful to improve the performance of machine learning algorithms and is often considered as applied machine learning.

Features are also referred to as ‘variables’ or ‘attributes’ as they affect the output of a process.

Feature engineering involves several processes. Feature selection, construction, transformation, and extraction are some key aspects of feature engineering

**4. Feature Transformation**

Feature transformation is simply a function that transforms features from one representation to another. But why would we transform our features?

Well there are many reasons, such as:

Data types are not suitable to be fed into a machine learning algorithm, e.g. text, categories

Feature values may cause problems during the learning process, e.g. data represented in different scales

We want to reduce the number of features to plot and visualize data, speed up training or improve the accuracy of a specific model.

**5**. **Implementation**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. The project is implemented by accessing simultaneously from more than one system and more than one window in one system.

1. **Comparing the ROC scores**

By analyzing and comparing ROC scores we choose the best model.

**6. Results and Visualization**

To visualize the model of the dataset we need to map them onto the graph by using matplotlib module which is present in the python.

**2.4 REQUIREMENT SPECIFICATIONS**

**Software Requirements**

The software used for the development of the project is:

* Programming Language: Python 3.7
* Tools: Anaconda
* Operating system: Windows

**Hardware Requirements**

The hardware used for the development of the project is:

* RAM: 2GB or above
* Processor: i3 or above
* Processor speed: 2.4GHz
* Hard disk: 500GB

**CHAPTER-3**

**SYSTEM DESIGN**

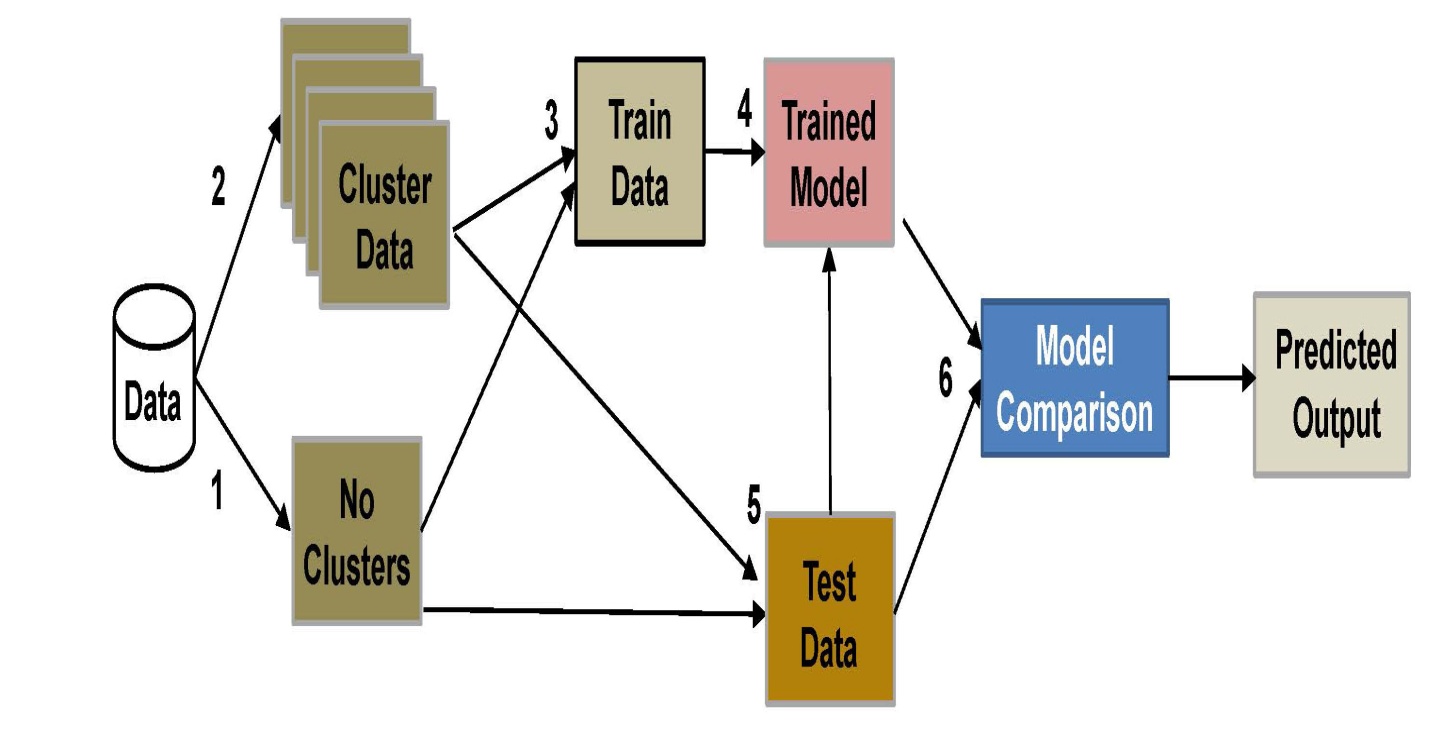
**3.1 INTRODUCTION**

The dataset was randomly partitioned into training set and testing set using a 80/20 train/test partition. Models are trained [4] using various algorithms using the entire training set, as well as trained on each cluster generated within the training set. Each model was trained once using what is referred to as a validation-set approach where there is one training set and one test set.

Once models are trained, the [5] test (i.e. holdout) data is fed into each trained model to measure [6] model performance. These measures allow us to gauge the generalizability of the remaining subset of data not used in the study, and provides us a feel to the degree of how overfit any models with respect to the training data.

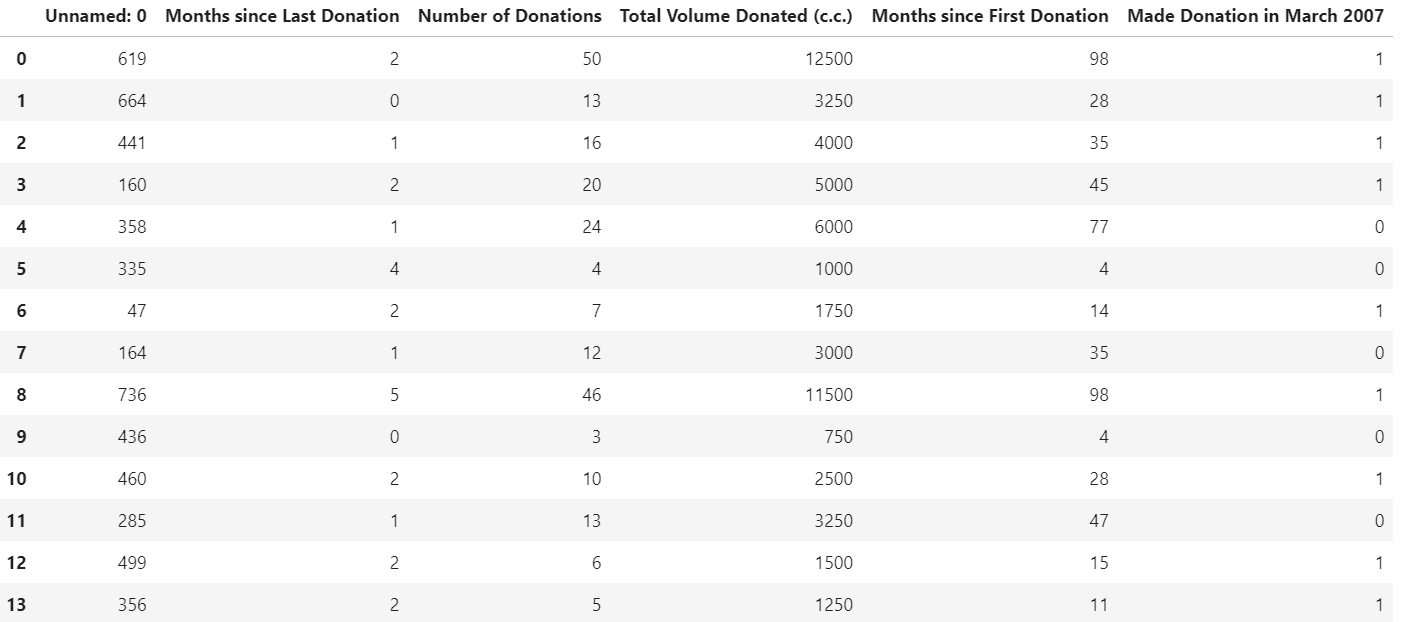
**3.2 BLOCK DIAGRAM**

The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.



3.1.1 Block Diagram

**3.3 DATA DICTIONARY**

****

• **Months since Last Donation**: this is the number of months since this donor's most recent donation.

• **Number of Donations**: this is the total number of donations that the donor has made.

• **Total Volume Donated**: this is the total amount of blood that the donor has donated in cubic centimeters.

**• Months since First Donation**: this is the number of months since the donor's first donation.

• **Made Donation**: this is the classifier on whether a given donor gives blood or not.

• **1**: Donated blood

• **0**: Not donated blood

**3.3 DATA FLOW DIAGRAMS**:

Data flow diagram (DFD) is a graphical representation of “flow” of data through an information system, modeling its process concepts. Often, they are a preliminary step used to create an overview of the system which can later be elaborated. DFD’s can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It doesn’t show information about timing of processes, or information about whether processes will operate in sequence or parallel. A DFD is also called as “bubble chart”.

**DFD Symbols:**

In the DFD, there are four symbols:

* A square defines a source or destination of system data.
* An arrow indicates dataflow. It is the pipeline through which the information flows.
* A circle or a bubble represents transforms dataflow into outgoing dataflow.
* An open rectangle is a store, data at reset or at temporary repository of data.

**Dataflow:** Data move in a specific direction from an origin to a destination.

**Process:** People, procedures or devices that use or produce (Transform) data. The physical component is not identified.

**Sources**: External sources or destination of data, which may be programs, organizations or other entity.

**Data store:** Here data is stored or referenced by a process in the system’s #

|  |  |
| --- | --- |
|  |  |

In our project, we had built the data flow diagrams at the very beginning of business process modeling in order to model the functions that our project has to carry out and the interaction between those functions together with focusing on data exchanges between processes.

**3.2.1 Context level DFD:**

A Context level Data flow diagram created using select structured systems analysis and design method (SSADM). This level shows the overall context of the system and its operating environment and shows the whole system as just one process. It does not usually show data stores, unless they are “owned” by external systems, e.g. are accessed by but not maintained by this system, however, these are often shown as external entities. The Context level DFD is shown in fig.3.2.1



Figure 4.2.1 Context Level DFD for Customer churn prediction

The Context Level Data Flow Diagram shows the data flow from the application to the database and to the system.

**3.2.2 Top level DFD:**

A data flow diagram is that which can be used to indicate the clear progress of a business venture. In the process of coming up with a data flow diagram, the level one provides an overview of the major functional areas of the undertaking. After presenting the values for most important fields of discussion, it gives room for level two to be drawn.

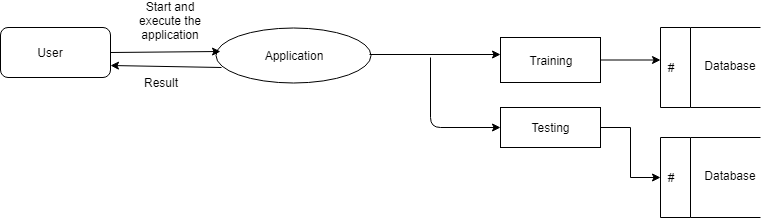


Figure 3.2.2 Top Level DFD

After starting and executing the application, training and testing the dataset can be done as shown in the above figure.

**3.2.3 Detailed Level Diagram**

This level explains each process of the system in a detailed manner. In first detailed level DFD (Generation of individual fields): how data flows through individual process/fields in it are shown.

In second detailed level DFD (generation of detailed process of the individual fields): how data flows through the system to form a detailed description of the individual processes.

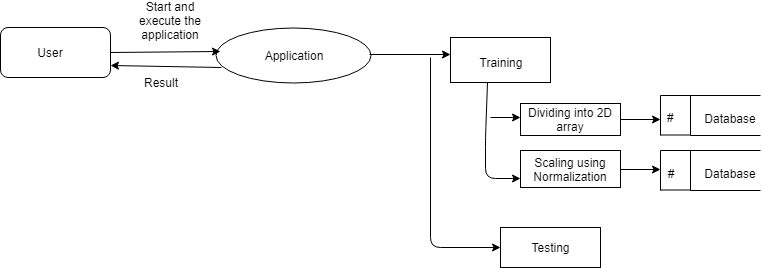
****

Figure 4.2.3.1 Detailed level DFD

After starting and executing the application, training the dataset is done by using dividing into 2D array and scaling using normalization algorithms, and then testing is done.

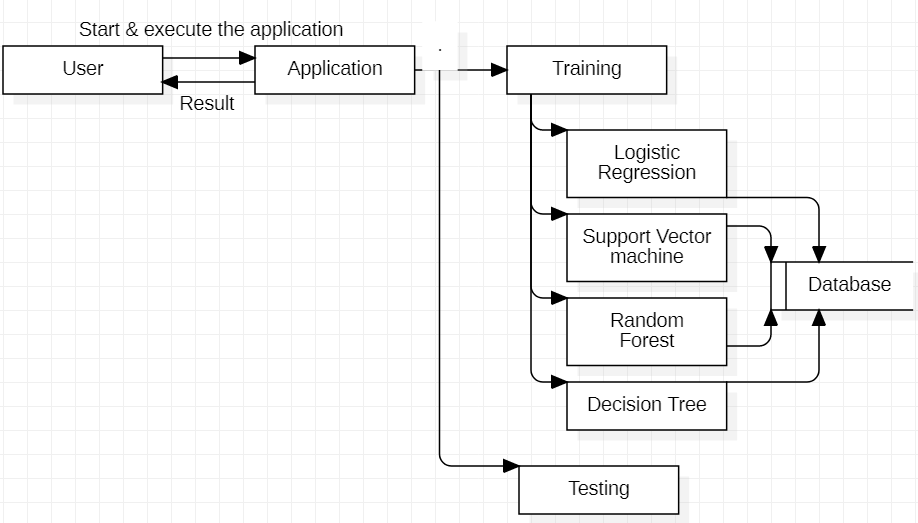


Figure 4.2.3.2 Detailed level DFD

After starting and executing the application, training the dataset is done by using linear regression and then testing is done.

**3.3 UNIFIED MODELLING LANGUAGE DIAGRAMS**:

The Unified Modelling Language (UML) is a Standard language for specifying, visualizing, constructing and documenting the software system and its components. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

* **User Model View** 
  1. This view represents the system from the user’s perspective.
  2. The analysis representation describes a usage scenario from the end-user’s perspective.
* **Structural Model View** 
  1. In this model the data and functionality are arrived from inside the system.
  2. This model view models the static structures.
* **Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* **Implementation model View**

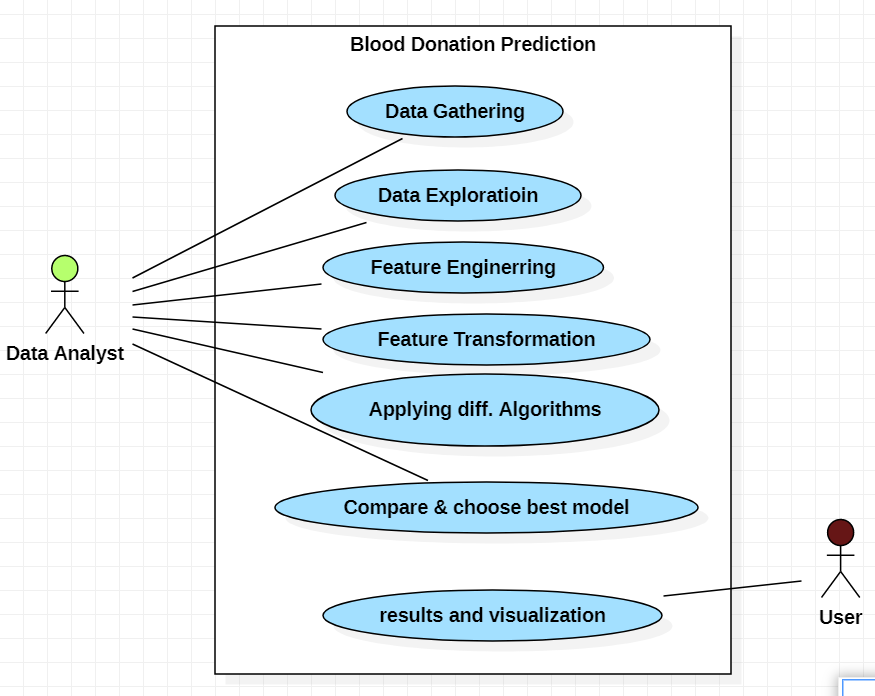
In this the structural and behavioral as parts of the system are represented as they are to be built.

* **Environmental Model View**

In this the structural and behavioral aspect of the environment in which the system is to be implemented are represented.

**3.3.1 Use Case Diagram:**

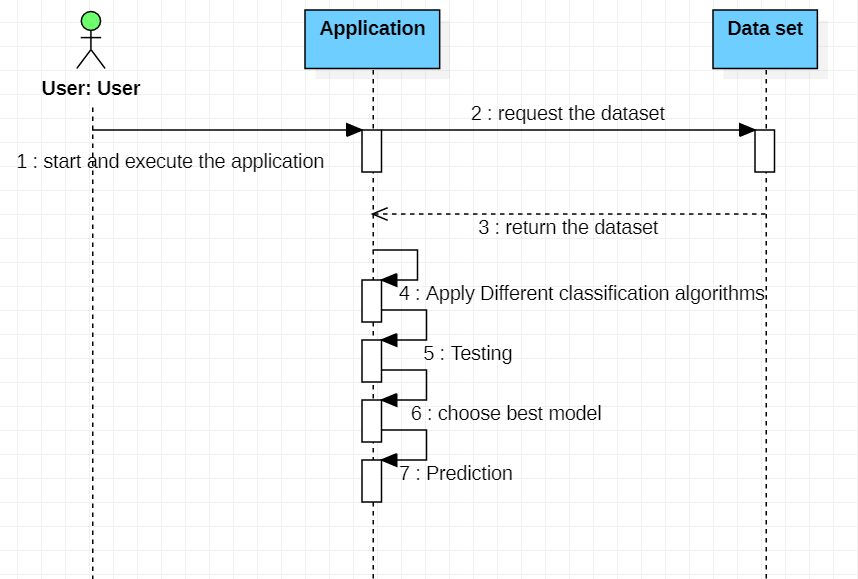
Use case diagrams are one of the five diagrams in the UML for modelling the dynamic aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modeling the dynamic aspects of systems).Use case diagrams are central to modelling the behavior of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations.



**Figure 3.3.1 USECASE DIAGRAM**

**3.3.2 Sequence Diagram:**

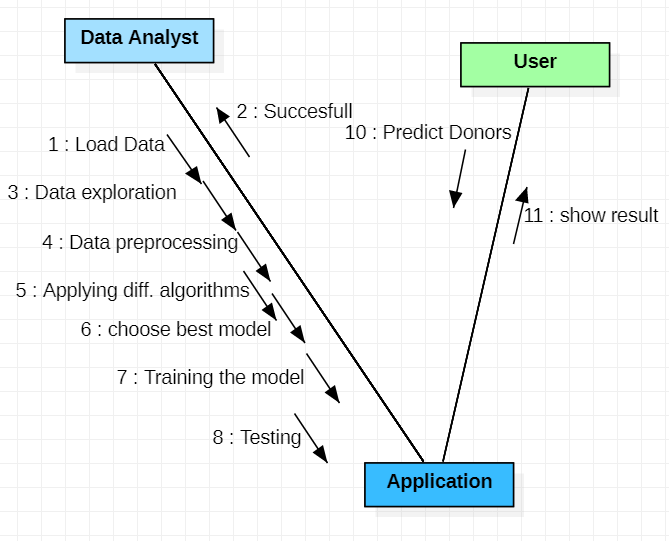
Sequence diagram is an interaction diagram which is focuses on the time ordering of messages. It shows a set of objects and messages exchanged between these objects. This diagram illustrates the dynamic view of a system.

Data 

**Figure 3.3.2 SEQUENCE DIAGRAM**

**3.3.3 Collaboration Diagram:**

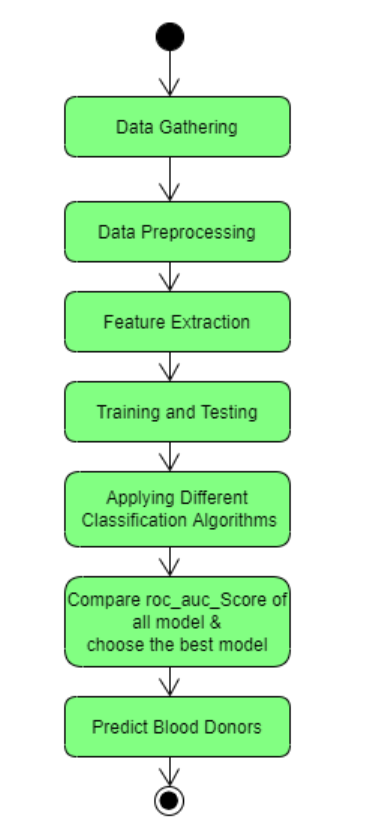
Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Collaboration diagram and sequence diagram are isomorphic.

****

**Figure 3.3.3 COLLABORATION DIAGRAM**

**3.3.4 Activity Diagram:**

An Activity diagram shows the flow from activity to activity within a system it emphasizes the flow of control among objects**.**

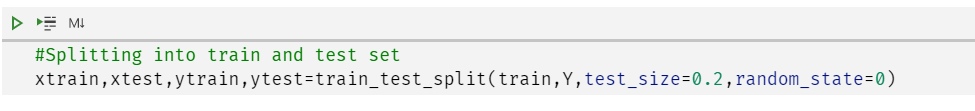


**Figure 3.3.3 ACTIVITY DIAGRAM**

**CHAPTER-4**

**IMPLEMENTATION**

Before making models we need to split the data into training and testing data sets

****

**4.1. Logistic Regression**

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome which is measured with two possible values for a variable. The variable in question has only two possible outcomes. I this case, there was a blood donation the value is 1 (TRUE, success, yes etc.) or 0 indication no donation (FALSE, failure, no etc.).

The goal of logistic regression is to find the best fitting model to describe the relationship between the variable in question (yes or no donation) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients as well as its standard errors and significance levels to predict a logic transformation of the probability.

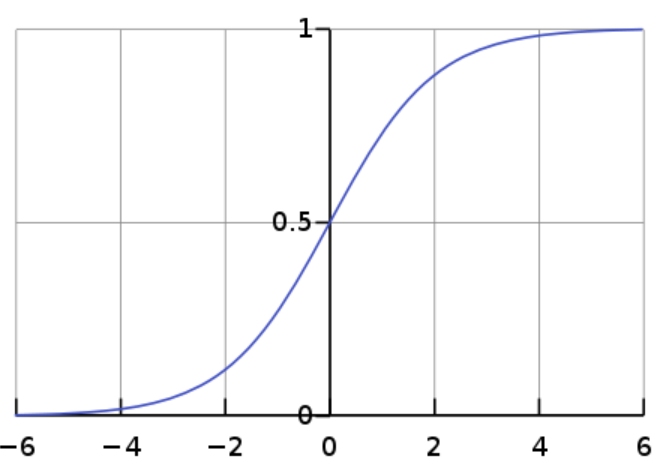
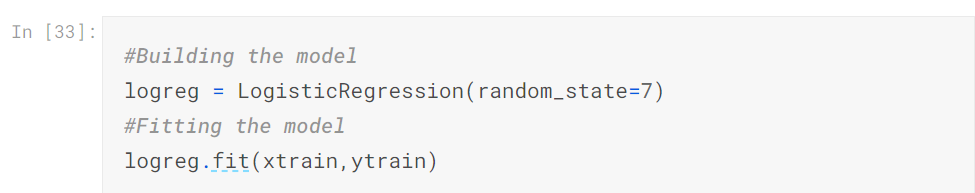


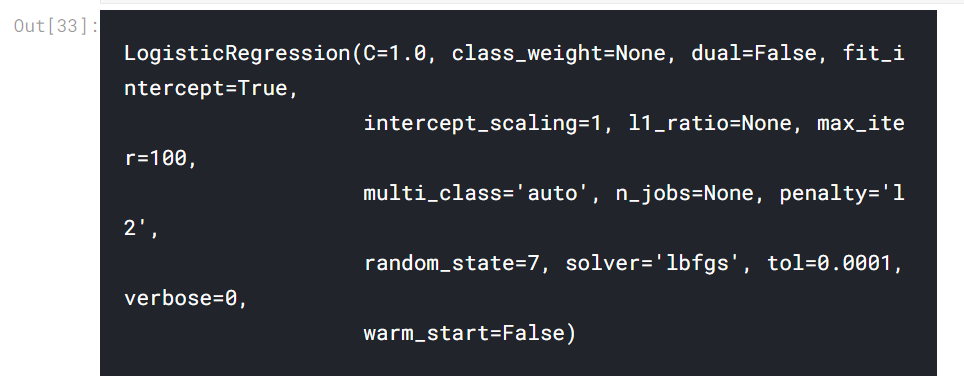
Fig.4.1 Logistic Regression

Rather than choosing parameters that minimize the sum of squared errors, estimation in logistic regression chooses parameters that maximize the likelihood of observing the sample values.

**Python Code**:

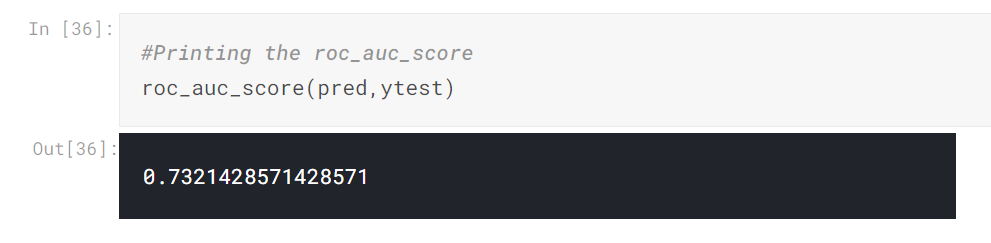


**Output:**

  
By removing the variables with the highest p-values from the previous model and running Logistic Regression model again on the dataset we obtain the following outcome.





****

Thus the model can predict 73 (= 67 + 6) observations correctly while 27 (= 21 + 6) observations incorrectly. This model gives us an accuracy of approximately 73%.

**4.2 Support Vector Machine**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

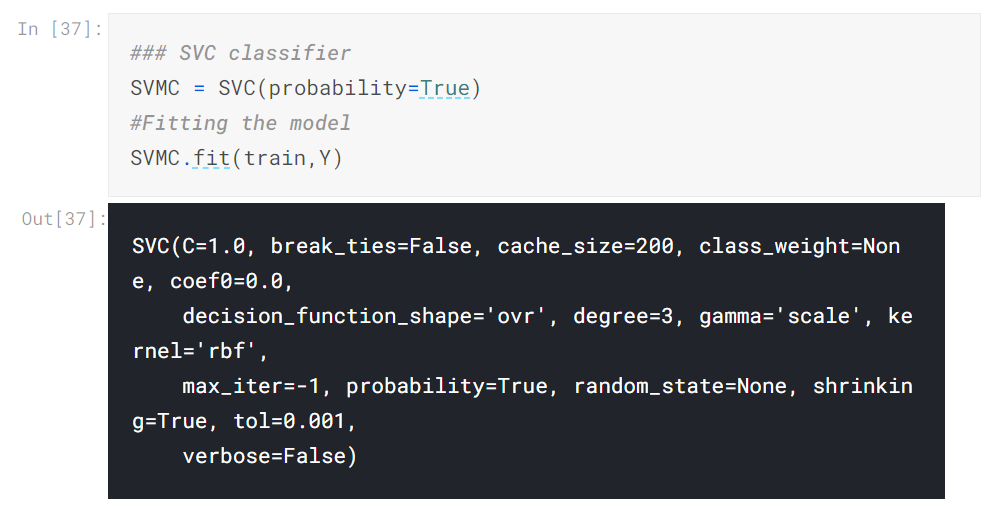
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

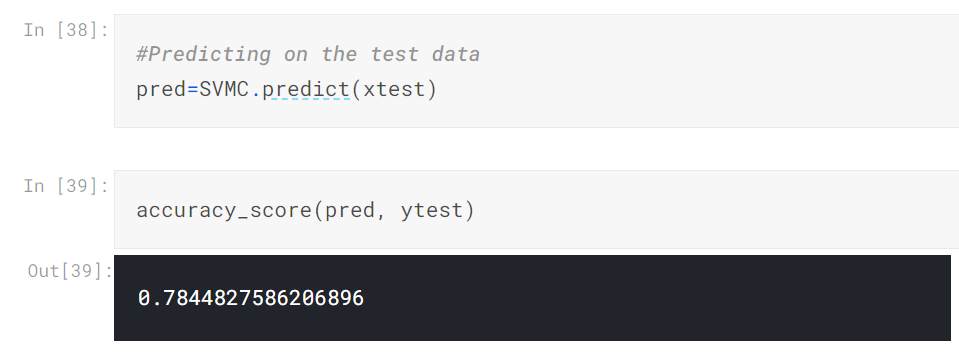


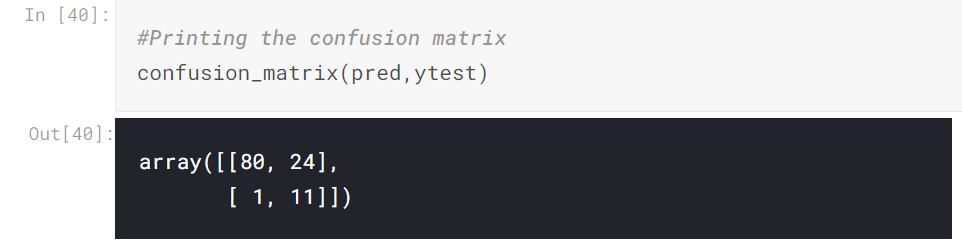
Fig,4.2 Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier defined by a separating hyperplane. Given labeled training data, 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 [6].

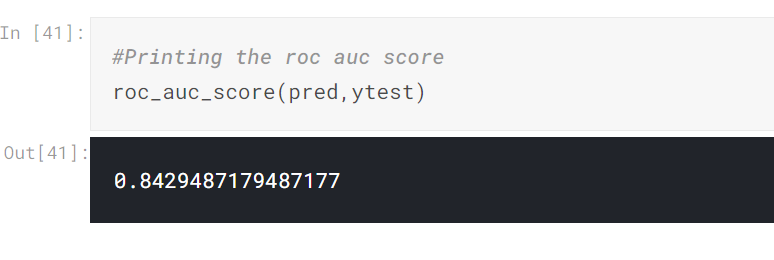
**Code:**







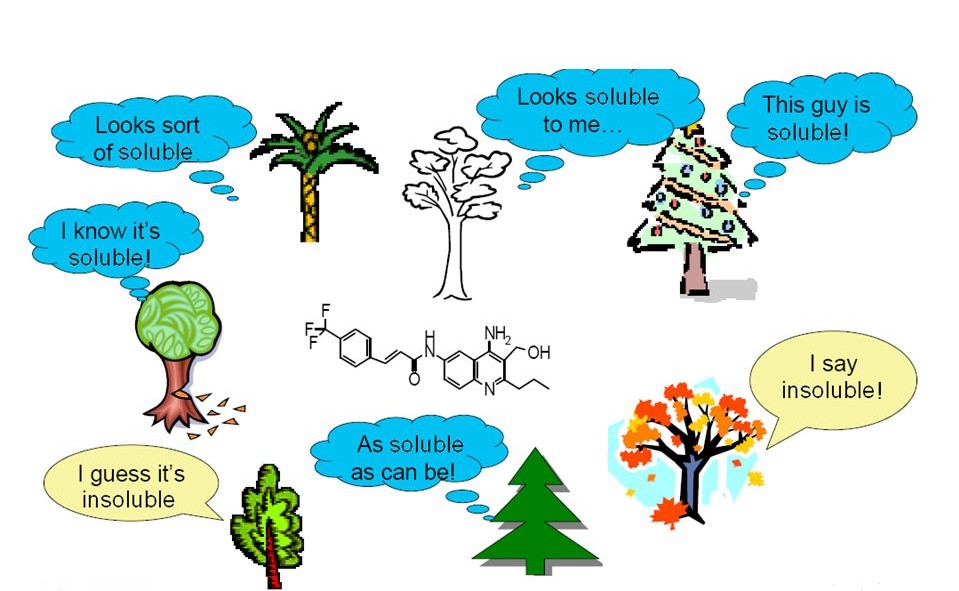
In this case, the model can predict 91 (= 80 + 11) observations correctly while 25(= 24 + 1) observations incorrectly.



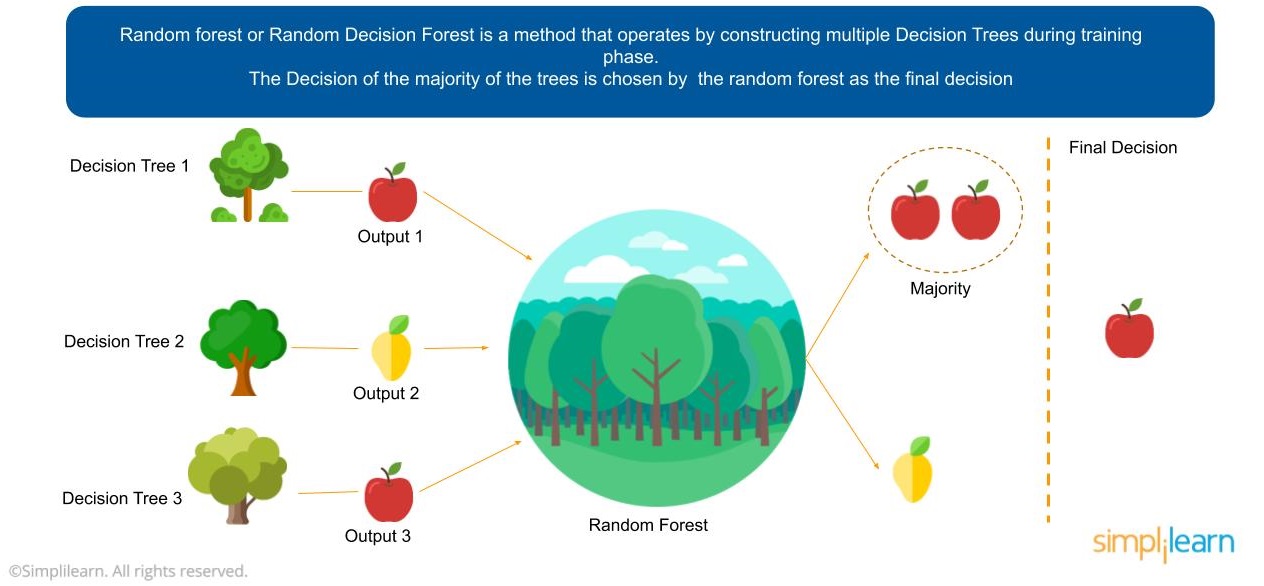
This model gives us an accuracy of approximately **84%** which is much higher than previous models.

1. **Random Forest**

Random forest algorithm is a supervised algorithm. As you can guess from its name this algorithm creates a forest with number of trees. It operates by constructing multiple decision trees. The final decision is made based on the majority of the trees and is chosen by the random forest.

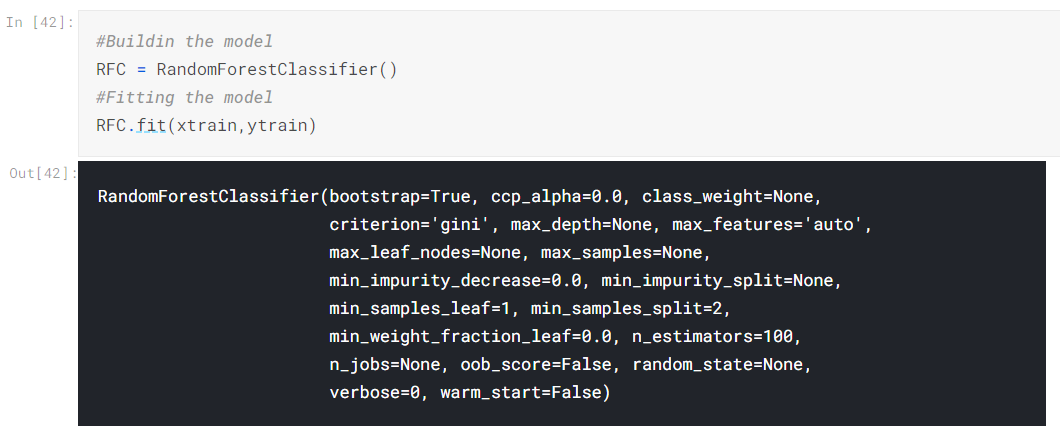


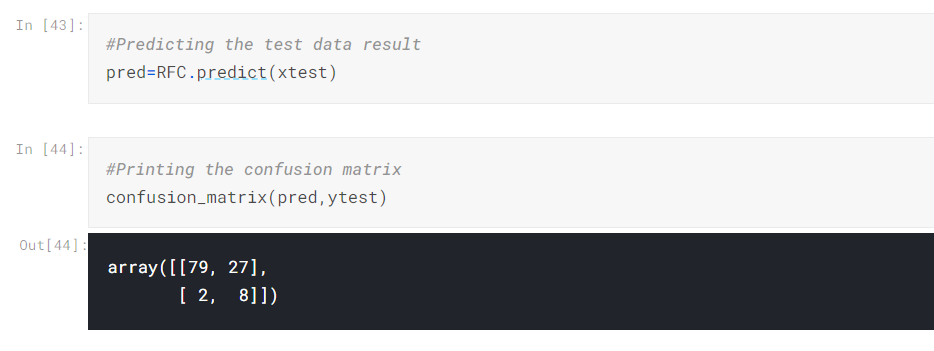
The method of combining trees is known as an ensemble method. Ensembling is nothing but a combination of weak learners (individual trees) to produce a strong learner



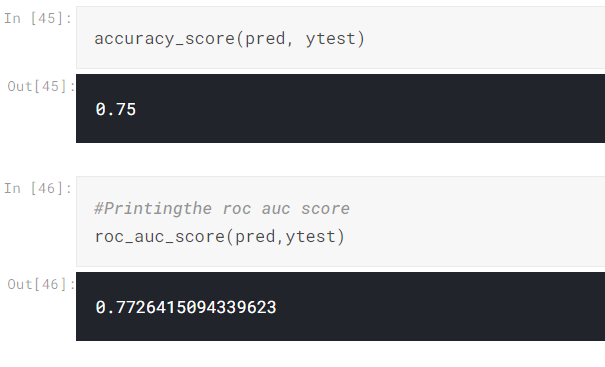
**Fig. 4.3 Random Forest**

Random Forest can be used to solve regression and classification problems. In regression problems, the dependent variable is continuous. In classification problems, the dependent variable is categorical.





This model can predict 87 (= 79 + 8) observations correctly while 29(= 27 + 2) observations incorrectly.



This model gives us an accuracy of approximately **77%.**

1. **Decision Tree**

Decision tree is very simple yet a powerful algorithm for classification and regression. As name suggest it has tree like structure. It is a non-parametric technique. A decision tree typically starts with a single node, which branches into possible outcomes. Each of those outcomes leads to additional nodes, which branch off into other possibilities. This gives it a tree like structure.

For example of a decision tree can be explained using below binary tree. Let’s suppose you want to predict whether a person is fit by their given information like age, eating habit, and physical activity, etc. The decision nodes here are questions like ‘What’s the age?’, ‘Does he exercise?’, and ‘Does he eat a lot of pizzas’? And the leaves, which are outcomes like either ‘fit’, or ‘unfit’. In this case this was a binary classification problem **(yes or no type problem).**

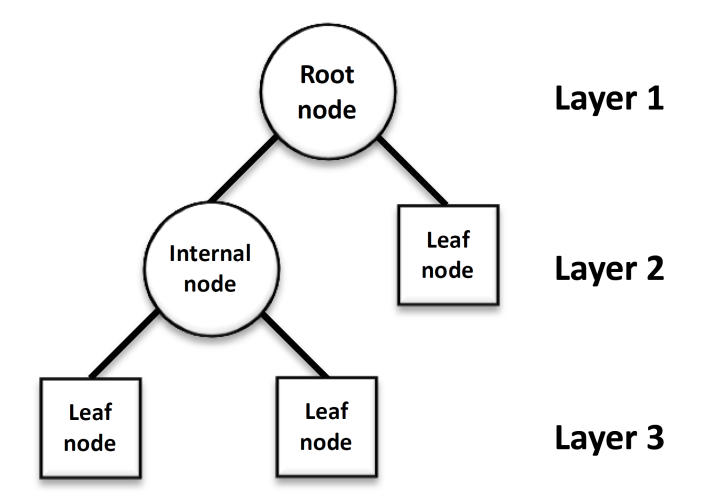
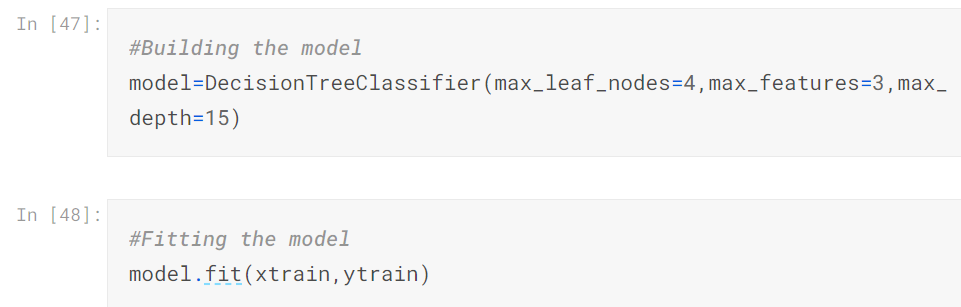
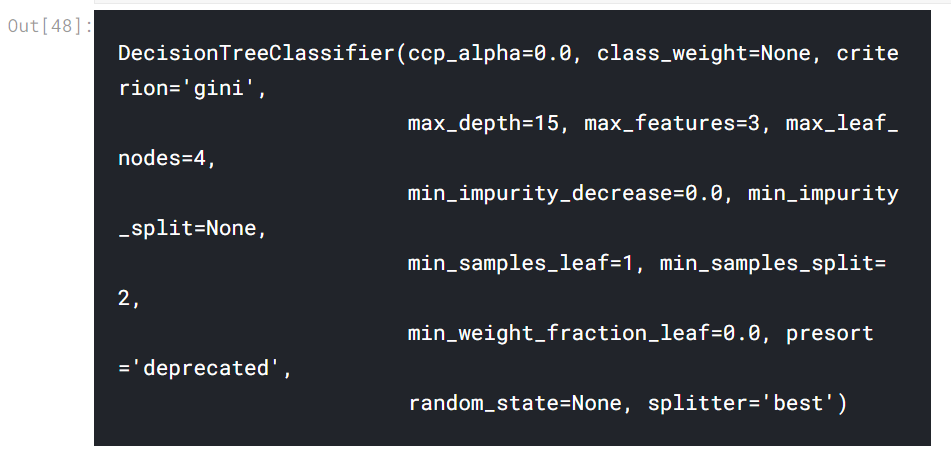
.

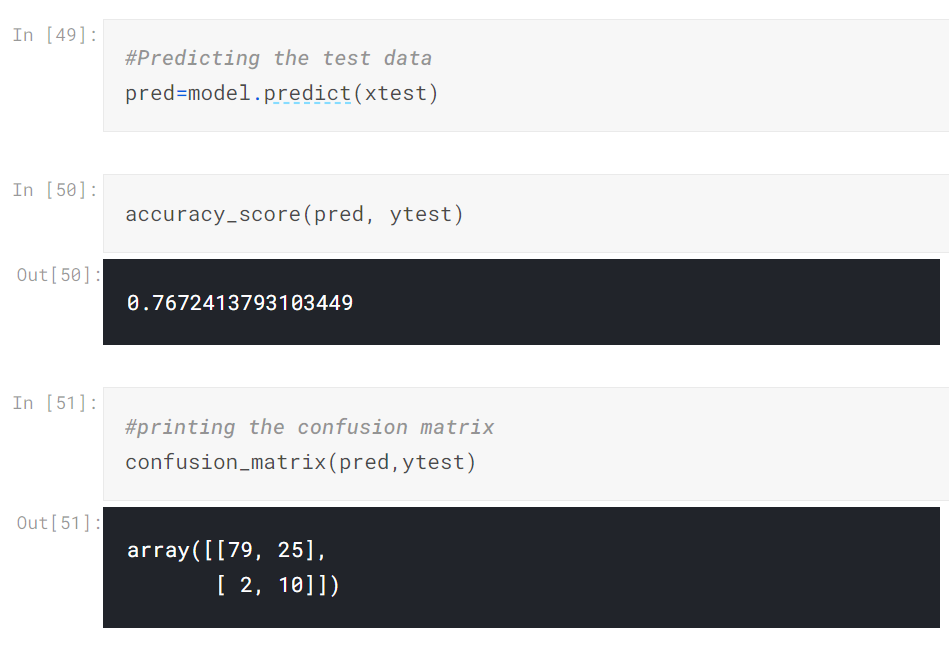
Fig. 4.4 Decision Tree

**Code**

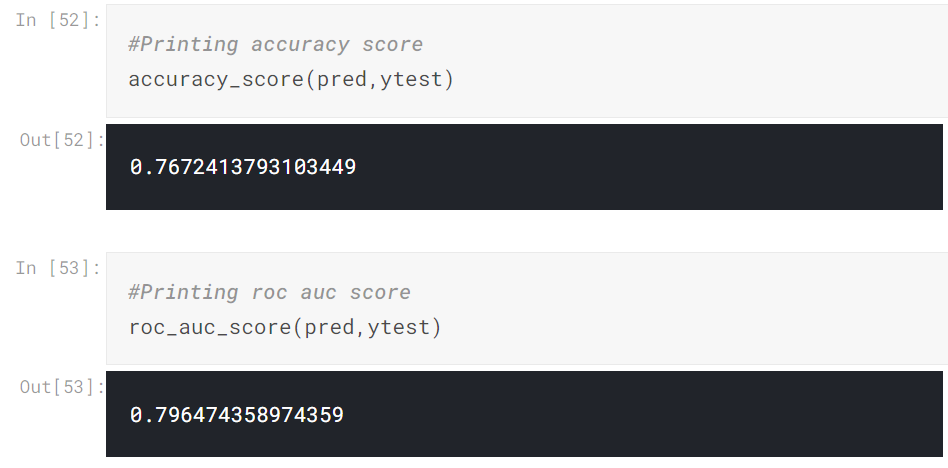
**For Building the model and to fit the model the following code is used**







In this case, the model can predict 89 (= 79 + 10) observations correctly while 27(= 25 + 2) observations incorrectly.



This model gives us an accuracy of approximately **79%.**

**CHAPTER-5**

**TESTING**

It is the process of testing the functionality and it is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as at undiscovered error. A successful test is one that uncovers an as at undiscovered error. Software testing is usually performed for one of two reasons:

* Defect Detection
* Reliability estimation

**5.1 BLACK BOX TESTING:**

The base of the black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system. Now a days, it is becoming to route the testing work to a third party as the developer of the system knows too much of the internal logic and coding of the system, which makes it unfit to test application by the developer. The following are different types of techniques involved in black box testing. They are:

* Decision Table Testing
* All pairs testing
* State transition tables testing
* Equivalence Partitioning

Software testing is used in association with Verification and Validation. Verification is the checking of or testing of items, including software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques as reviews, inspections, walk-through. Validation is the process of checking what has been specified is what the user actually wanted.

* Validation: Are we doing the right job?
* Verification: Are we doing the job right?

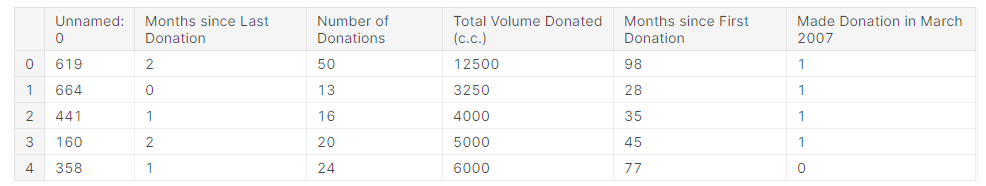
In order to achieve consistency in the Testing style, it is imperative to have and follow a set of testing principles. This enhances the efficiency of testing within SQA team members and thus contributes to increased productivity.

**CHAPTER-6**

**RESULT**

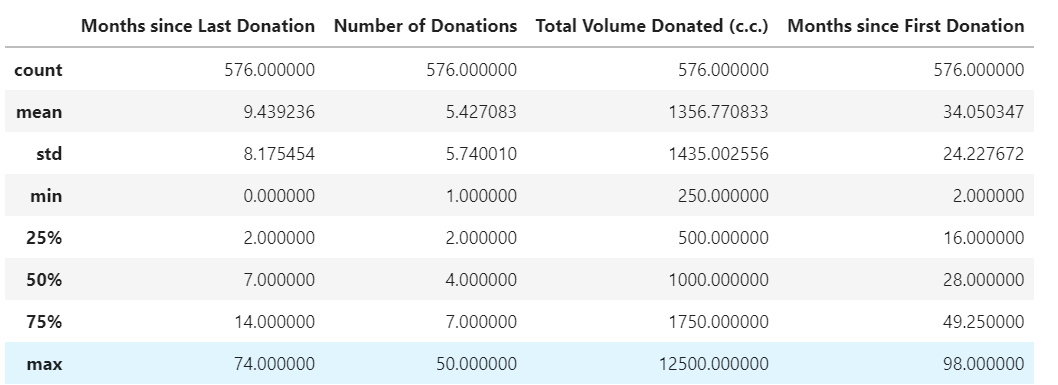
In machine learning project Result is viewed by module wise.

1. **Gathering of Data**

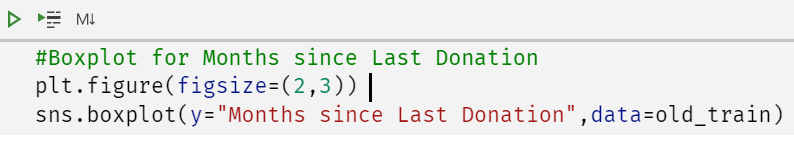
****

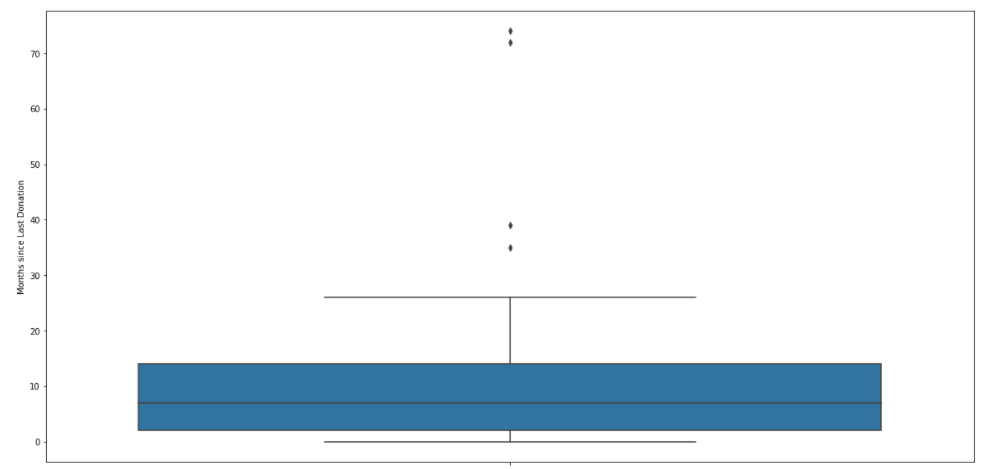
**Fig 6.1:** Shows data in blood donation dataset.

1. **Data Exploration**

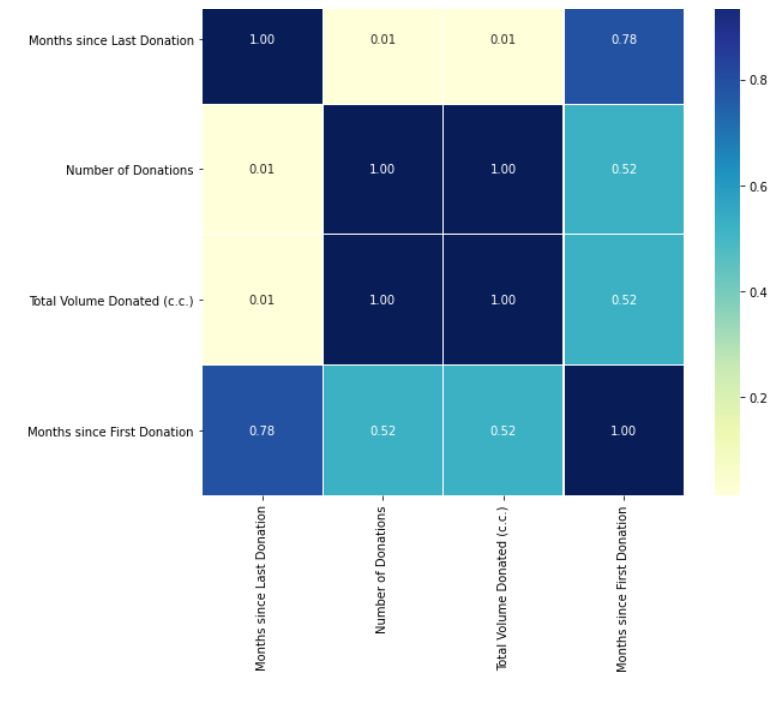
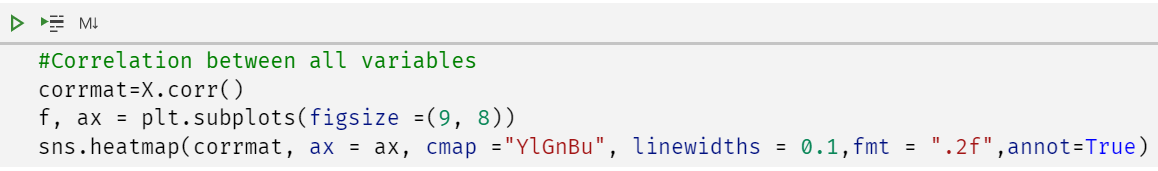
****

**Fig 6.2:** Data description





**Fig 6.3:** Box Plot



**Fig 6.4:** Correlation between variables

**3. Result and Visualization**

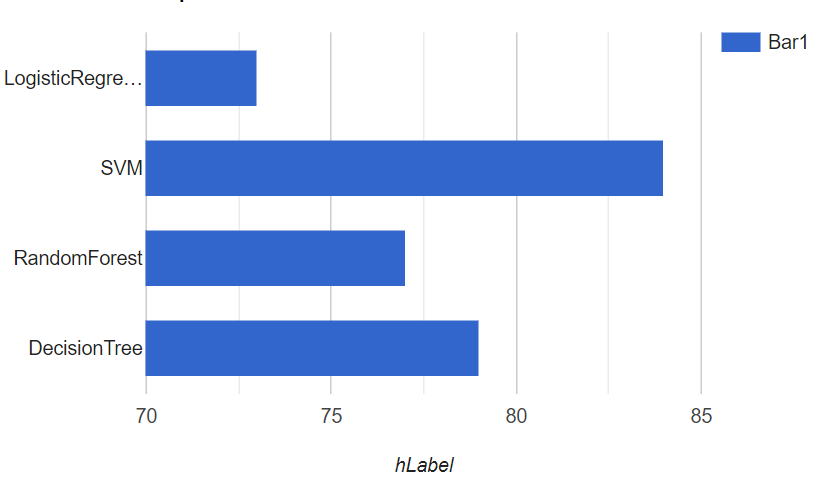
****

Fig. 6.5 Result

**CHAPTER-7**

**CONCLUSION**

We have concluded that machine learning techniques like SVM and decision trees etc., are vital for the development of a balanced and dependable mechanism for communicating with and identifying viable blood donors. The existing system has physical redundancies which would be solved by integrating machine learning. Initiatives like these would lead to a decrease in response time and avoidable costs would be cut down. The possibility of finding an ideal donor would increase resulting in lower rates of transfusion related complexities.

**CHAPTER-8**

**FUTURE SCOPE &ENHANCEMENT**

To model our blood donation pattern, we can improve our current model by including more variables into our model like donor demographics, attitude, subjective norm, self-efficacy, intention, moral norm, self-identity, anticipated regret, donation anxiety and behavior.

In order to improve the number of people donating blood, we can focus more in blood donation campaigns focused towards people who recently donated blood against those people who have donated blood in a long period of time. The reason is that by focusing more on people who have recently donated blood would increases the odds of these individuals to donate their blood once again.

**CHAPTER-9**

**BIBLIOGRAPHY**

[1] A. Mireille Baart, Y. V. (2013, Sep 30). *Prediction of hemoglobin levels in whole blood donors:*

*how to model donation history*. Retrieved Nov 17, 2017, from Wiley Online Library: The third article, “Predicting blood donation intentions and behavior among Australian blood donors: testing an extended theory of planned behavior model” by Barbara M. Masser, Katherine M. White, Melissa K. Hyde, Deborah J. Terry, and Natalie G. Robinson (September 1, 2008).

[2] GivingBlood. (n.d.). *Why Give Blood*. Retrieved Nov 16, 2017, from Community Blood Center

Real Donors. Real Life Heroes: <http://givingblood.org/donate-blood/why-give-blood.aspx>

[3] Masser BM, W. K. (2008, Nov 19). *Wiley Online Library*. Retrieved Nov 17, 2017, from

Predicting blood donation intentions and behavior among Australian blood donors: testing an extended theory of planned bahavior model.: <http://onlinelibrary.wiley.com/doi/10.1111/j.1537-2995.2008.01981.x/abstract>

[4] Program, A. S. (n.d.). *Together we can make a difference*. Retrieved Nov 17, 2017, from ASBP

Armed Services Blood Program: <http://www.militaryblood.dod.mil/Donors/about_blood.aspx>

[5] Wullianallur Raghupathi, V. R. (2014, Feb 7). *Big data analytics in healthcare: promise and*

*potential*. Retrieved Nov 17, 2017, from PMC US National Library of Medecine National Institute of Health: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4341817/>

[6] Chapter 2 : SVM (Support Vector Machine) — Theory,

https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

**CHAPTER-10**

**Appendix**

**Basic Concepts:**

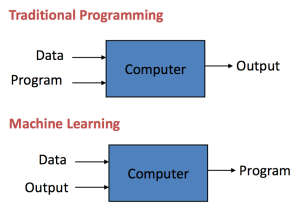
Machine Learning is getting computers to program themselves. If programming is automation, then machine learning is automating the process of automation.

Writing software is the bottleneck, we don’t have enough good developers. Let the data do the work instead of people. Machine learning is the way to make programming scalable.

* **Traditional Programming**: Data and program is run on the computer to produce the output.
* **Machine Learning**: Data and output is run on the computer to create a program. This program can be used in traditional programming.

Machine learning is like farming or gardening. Seeds is the algorithms, nutrients are the data, the Gardner is you and plants is the programs.

**Traditional Programming vs Machine Learning:**



Every machine learning algorithm has three components:

* **Representation**: how to represent knowledge. Examples include decision trees, sets of rules, instances, graphical models, neural networks, support vector machines, model ensembles and others.
* **Evaluation**: the way to evaluate candidate programs (hypotheses). Examples include accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence and others.
* **Optimization**: the way candidate programs are generated known as the search process. For example, combinatorial optimization, convex optimization, constrained optimization.

### **Types of Learning:**

There are four types of machine learning:

* **Supervised learning**: (also called inductive learning) Training data includes desired outputs.  This is spam this is not, learning is supervised.
* **Unsupervised learning**: Training data does not include desired outputs. Example is clustering. It is hard to tell what is good learning and what is not.
* **Semi-supervised learning**: Training data includes a few desired outputs.
* **Reinforcement learning**: Rewards from a sequence of actions. AI types like it, it is the most ambitious type of learning.

### **Applications of Machine Learning:**

Sample applications of machine learning:

* **Web search**: ranking page based on what you are most likely to click on.
* **Computational** biology: rational design drugs in the computer based on past experiments.
* **Finance**: decide who to send what credit card offers to. Evaluation of risk on credit offers. How to decide where to invest money.
* **E-commerce**:  Predicting customer churn. Whether or not a transaction is fraudulent.
* **Space exploration**: space probes and radio astronomy.
* **Robotics**: how to handle uncertainty in new environments. Autonomous. Self-driving car.
* **Information extraction**: Ask questions over databases across the web.
* **Social networks**: Data on relationships and preferences. Machine learning to extract value from data.