STARTUP SUCCESS RATE PREDICTION

A Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Engineering / Electronics & Communication Engineering/ Electrical & Electronics Engineering

By

19K41A0590	Amogh Varsh Raju
19K41A0592	Anugam Saikiran
19K41A04F3	Asmath Fathima

Under the Guidance of
Dr. V. Venkataramana

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S.R.ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA) Dec-2021.

SR Engineering College
Innovation . Creativity . Entrepreneurship

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled " Startup Success Rate Prediction" is a record of bonafide work carried out by the student(s) Amogh Varsh Raju, Anugam Sai Kiran, Asmath Fathima bearing Roll No(s) 19K41A0590, 19K41A0592, 19K41A04F3 during the academic year 2021 – 2022 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering/Electronics & Communication Engineering/Electrical & Electronics Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

.

Supervisor **Head of the Department**

External Examiner

ABSTRACT

Startups have bought a tremendous impact in the world of economy. Over the past few years there has been an exponential growth. Its all about the plan of action and organization of the ideas that make it successful so as to attract the new investors and entrepreneurs. The success prediction is the vital criteria for new ventures in the way for investments , since the first step in the path of success is the potential of growing . With the developments in the model with new features using different dataset, we can make it more effectively . The results are used by enriching data and deploying these machine learning models through research and making its performance more accurate and dynamic to work.

Table of contents

1.INTRODUCTION

Startup and entrepreneurial ecosystem have become an essential element of innovation. They are rapidly growing, and need monitoring of the performance and enhancing it. Prior to the startup it's important to analyze what makes it successful. In the approach of goal we have to go beyond the data previously used with different success factors.
Predicting success is defined based on the money invested by founders and investors and the in turn benefit they achieve for the shares.
With a focus on how to explore and make a better decision of making investment strategy and gain by applying improved machine learning we can make the process more effective.

To attain success accuracy we can work on the following factors :

a. Explore new data variables, by brain storming with stakeholders and investors.
b. Implement more accurate models to enhance prediction on startup.

The insights provided will be valuable for stakeholders and venture capitals. The predictive model to explain the phenomena is the indicator of data mining that allow to achieve full potential of data.

1. Literature Survey

Machine learning models have made artificial intelligence an easy, effective identification of the patterns with a quick pace of improvement and efficient handling. The methodology used is to firstly data Preprocessing intermediate data-set and check the setup by cleaning, selection and transformation. Then , Carried out with the final data-set get the result through use of different ML models. Although the startups have brought a revolutionary change in the market yet its important to have a prior hand knowledge of what it would end up with before investing. Its a risky and unpredictable as the new Startup may or may not work. Logistic regression, SVC ,Decision tree classifier, Random forest. K-Nearest Neighbor models have been used. In the need of getting accuracy we have used different model. In proposed system our model showed accuracy rates with the highest accuracy rate in SVC.

DESIGN:

3.1 REQUIREMENT SPECIFICATION(S/W & H/W)

Hardware Requirements

- ✓ **System** : Pentium 4, Intel Core i3, i5, i7 and 2GHz Minimum
- ✓ **RAM** : 4GB or above
- ✓ **Hard Disk** : 10GB or above
- ✓ **Input** : Keyboard and Mouse
- ✓ **Output** : Monitor or PC

Software Requirements

- ✓ **OS** : Windows 8 or Higher Versions
- ✓ **Platform** : Jupiter Notebook
- ✓ **Program Language** : Python

1.2 FLOW CHART

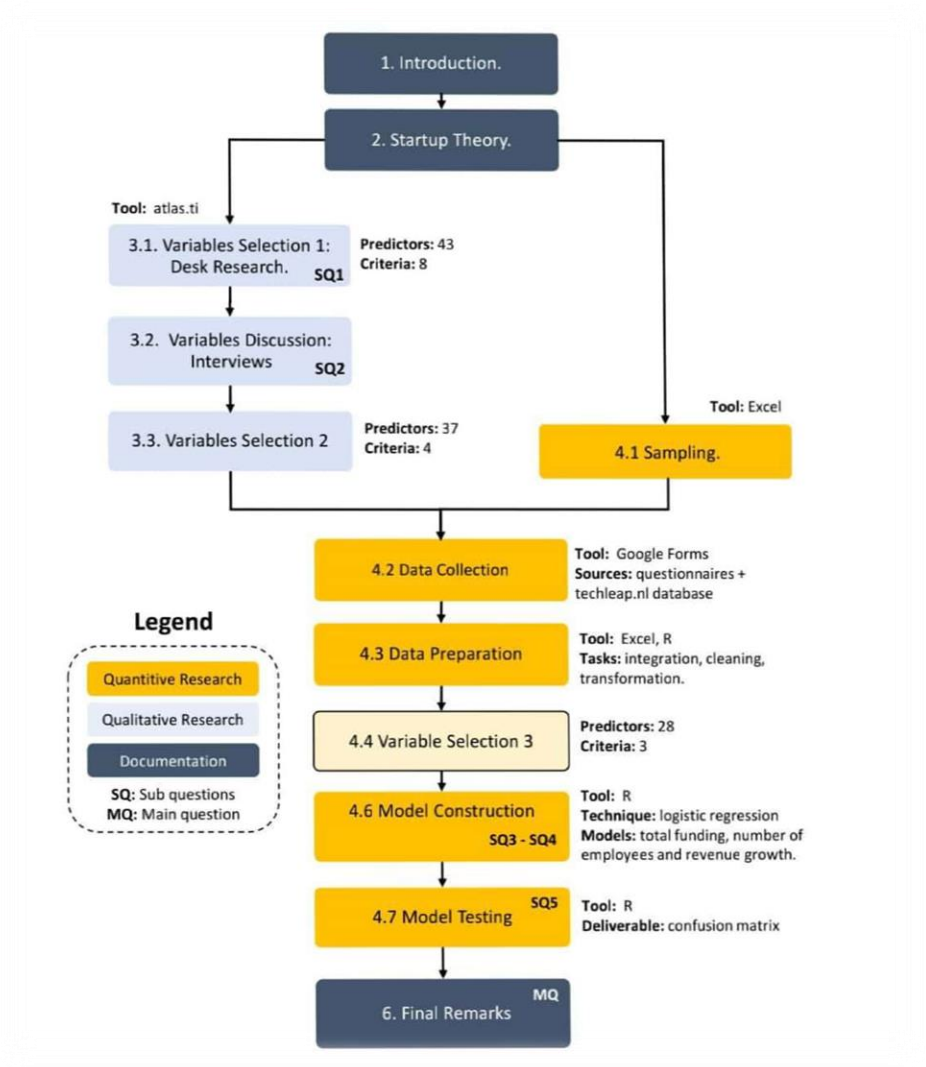

Figure 2 - flow chart

1. DATA SET

We collected the data from the Kagil Startup success prediction question. We use 7 attributes to predict whether or not a startup is successful or failure.

Labels	The Startup has marketing labels or not.	Nominal
Age first Year	Age of the company when it first received its funding	Nominal
Age first milestone	Age of the company when it reached its first milestone.	Numeric
Relationships	The company relationship rating between its users and customers.	Numeric
Milestones	The number of milestones the company has reached until now	Nominal
Avg-Participants	Amount of employees in the company including the founder and Directors.	Nominal
Is top 500	Did the startup or company reach the top places globally in top 500	Numeric

Table 1. kagil Data-set attributes detailed information

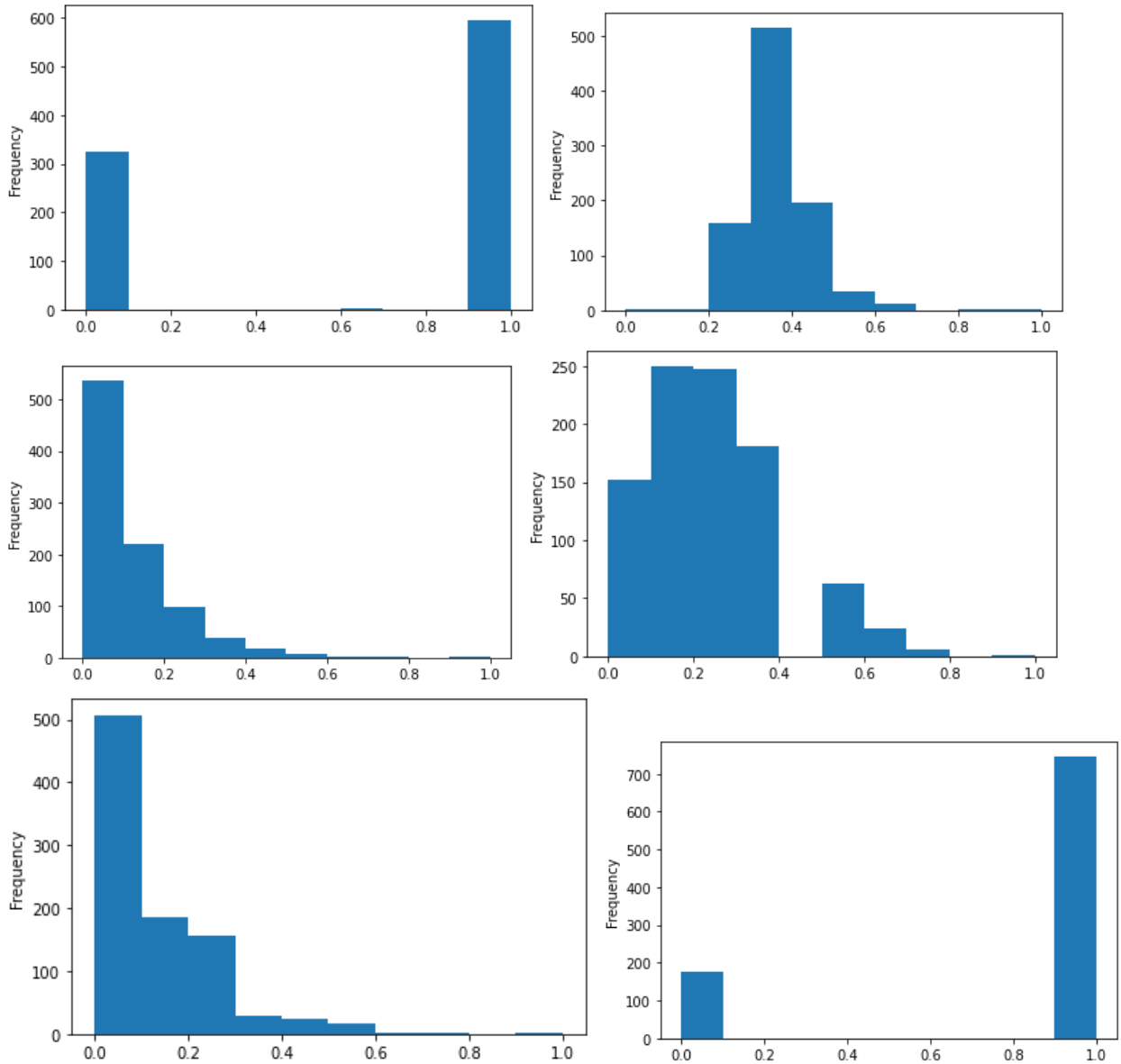

Figure 3. Visualizing attributes of the dataset

1. DATA PREPROCESSING:

After an assortment of data, The total number of rows available data is 923 rows of data for the attributes of Marketing labels,age_first_year,age_first_milestone_year, relationships, milestones, avg_participants,is_top500 and these 8 columns provide the exact necessary data which is required to be calculated , these columns have been filtered from 44 columns available to 23 columns and the best columns have been selected those are these 8 columns

Labels.	Boolean(The company has,marketing label or not,marketing partner or not, marketing partner is present or not.)
age_first_year.	Float(The age of the company:When it received its first funding,first investment ,first transaction)
Age_milestone_year.	Float(The age of the company when it received:its first success,its first threshold limit,its first profit,its first pitch,its first user base.)
Relationships	Float(The number of people who are working with the company and are purchasing from the company.)
Milestones	Float(The number of milestones reached by the company)
Avg_participants	Float(The number of employees along with the founder and management team of the company)
Is_top500	The company is in top 500 companies world wide or is available in the magazines or is it well known.

Table 2. Attributes Validation.

So, from the above graph, we can predict that the startup can be successful or not.

Correlation Matrix

A correlation matrix is simply a table that displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a scatterplot. coefficients for different variables.

How it is calculated?

A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable (X_i) in the table is correlated with each of the other values in the table (X_j)... The diagonal of the table is always a set of ones because the correlation between a variable and itself is always 1. Let's perform the Correlation matrix to understand the relation between the dependent variable and the independent variable and within the independent variable.

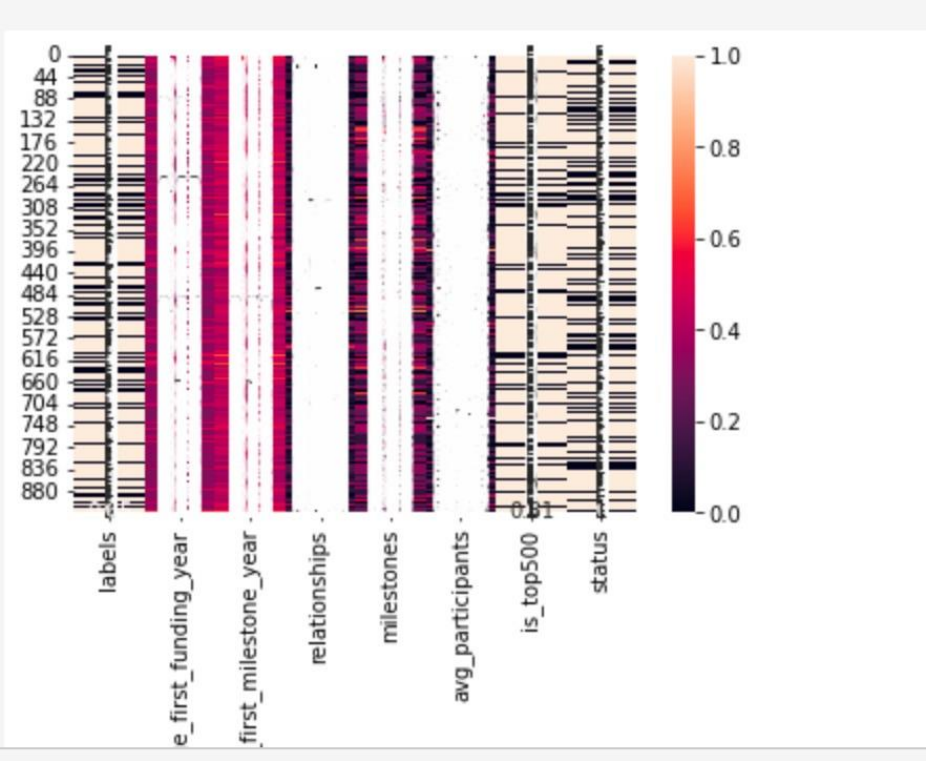

Figure 7 Correlation Matrix

Enough EDA performs on the data to evaluate the data-set and gather knowledge about the data. Let's perform some Machine Learning model and Experimentation to create a model that helps us to achieve our goal we state in the problem definition.

2. METHODOLOGY:

This section talks about the algorithms used for the project. We used three different algorithms like Logistic Regression, Random forest, K-Near Neighbor.

Logistic Regression

Logistic regression is a supervised algorithm for study classification. The likelihood of a destination variable was predicted. The nature of the target or dependent variable is dichotomous, meaning that only two possible classes are available[11].

Steps for Logistic Regression:

Step 1 : Data Preprocessing step

Step 2 : Fitting Logistic Regression to the training set

Step 3 : Predicting the test Result

Step 4 : Test accuracy to the result

Step 5 : visualizing the test set result

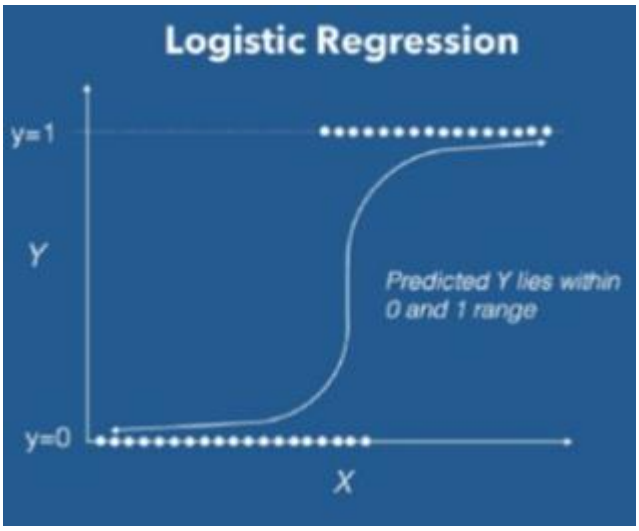

Figure 8. Logistic Regression

K-Nearest Neighbor

The **k-Nearest-Neighbors (KNN)** method of classification is one of the simplest methods in machine learning and is a great way to introduce yourself to machine learning and classification in general. At its most basic level, it is essentially classification by finding the most similar data points in the training data and making an educated guess based on their classifications. Although very simple to understand and implement, this method has seen wide application in many domains, such as in **recommendation systems**, **semantic searching**, and **anomaly detection[12]**.

As we would need to in any machine learning problem, we must first find a way to represent data points as **feature vectors**. A feature vector is our mathematical representation of data, and since the desired characteristics of our data may not be inherently numerical, preprocessing and feature-engineering may be required to create these vectors. Given data with **N** unique features, the feature vector would be a vector of length **N**, where entry **I** of the vector represents that data point's value for the feature **I**. Each feature vector can thus be thought of as a point in **R^N**.

Now, unlike most other methods of classification, KNN falls under **lazy learning**, which means that there is **no explicit training phase before classification**. Instead, any attempts to generalize or abstract the data is made upon classification. While this does mean that we can immediately begin classifying once we have our data, there are some inherent problems with this type of algorithm. We must be able to keep the entire training set in memory unless we apply some type

of reduction to the data-set, and performing classifications can be computationally expensive as the algorithm parse through all data points for each classification. **For these reasons, KNN tends to work best on smaller data-sets that do not have many features.** Once we have formed our training data-set, which is represented as an **M** x **N** matrix where **M** is the number of data points and **N** is the number of features, we can now begin classifying[12]. The gist of the KNN method is, for each classification query, too;

Two important decisions must be made before making classifications. One is the value of **k** that will be used; this can either be decided arbitrarily, or you can try **cross-validation** to find an optimal value. The next, and the most complex, is the **distance metric** that will be used. There are many different ways to compute distance, as it is a fairly ambiguous notion, and the proper metric to use is always going to be determined by the data-set and the classification task. Two popular ones, however, are **Euclidean distance** and **Cosine similarity**. Euclidean distance is probably the one that you are most familiar with; it is essentially the magnitude of the vector obtained by subtracting the training data point from the point to be classified.

$$E(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

General formula for Euclidean distance

Another common metric is Cosine similarity. Rather than calculating a magnitude, Cosine similarity instead uses the difference in direction between two vectors.

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

General formula for Cosine similarity

Choosing a metric can often be tricky, and it may be best to just use cross-validation to decide unless you have some prior insight that leads to using one over the other. For example, for something like word vectors, you may want to use Cosine similarity because the direction of a word is more meaningful than the sizes of the component values. Generally, both of these methods will run at roughly the same time and will suffer from highly-dimensional data.

After doing all of the above and deciding on a metric, the result of the KNN algorithm is a decision boundary that partitions **R^N** into sections. Each section (coloured distinctly below) represents a class in the classification problem. The boundaries need not be formed with actual training examples — they are instead calculated using the distance metric and the available training points. By taking **R^N** in (small) chunks, we can calculate the most likely class for a

hypothetical data-point in that region, and we thus colour that chunk as being in the region for that class.

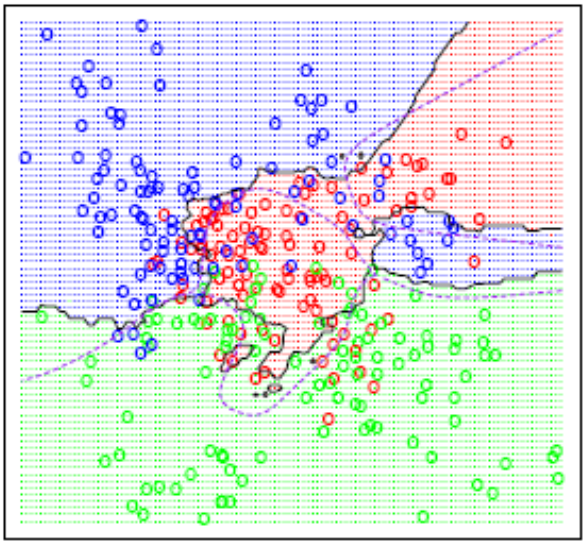

Figure 9. - KNN

The following steps are shown below : -

Step 1 − We need a dataset to implement any algorithm. So we need to load both training and test data during the first stage of KNN.

Step 2 − The next step is to select the K value, i.e. the closest datasets. K can be any integer. Any integer.

Step 3 − The following applies to each point in the test data

3.1 - To calculate, using any of the methods namely the distance Euclidean, Manhattan or Hamming data from the test data and each row of training data. Euclidean is the most common way of calculating distance.

3.2 − Sort them now in ascending order based on the distance value.

3.3 − Next the top K rows of the classified array will be chosen.

3.4 − The t class will now be assigned

Random forest:

Random forest is used for both regression and classification-based applications. This algorithm is flexible and easy to use. Most of the times this algorithm gives accurate results even without hyper tuning the parameters. It builds many decision trees which on merging

forms as a forest. While building the decision trees, adds more randomness to the model. This algorithm searches for the best feature in the random subset of features, which results in the formation of a better model[13].

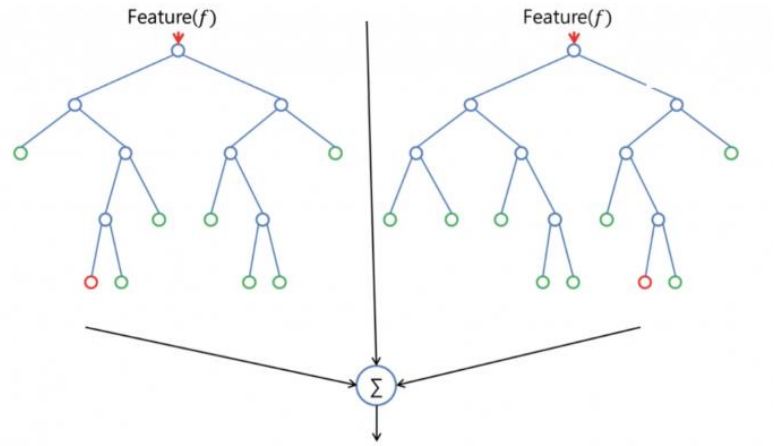

Figure 10: Random forest classifier

With the help of the sklearn library, we can measure the relative importance of each feature in prediction. By finding the feature importance, we can drop the features that have less importance in the prediction process. The main limitation of random forest is that many trees can make the algorithm too slow and ineffective for real-time predictions. In general, these algorithms are fast to train but quite slow to create predictions once they are trained. A more accurate prediction requires more trees, which results in a slower model. In most real-world applications, the random forest algorithm is fast enough but there can certainly be situations where run-time performance is important and other approaches would be preferred[13].

With the aid of the following steps we can understand how the Random Forest algorithm works.

Step 1 − First, select random samples from a particular dataset.
Step 2 - Next, for each sample this algorithm will build a decision tree. Then every decision tree will predict the result.
Step 3 − Every predicted result will be voted in this step.
Step 4 − Finally, the final prediction result will be selected as the most voted prediction result.

Support Vector Machine SVM

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N - the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum

distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

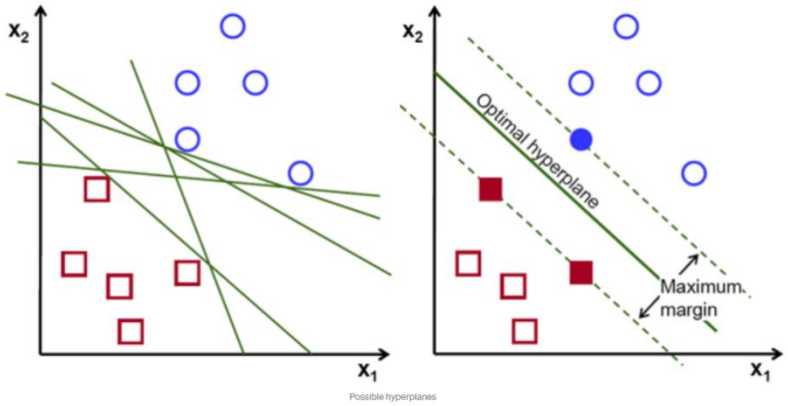

Figure 11 - SVM

XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly **efficient**, **flexible** and **portable**. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

Navies Bayes

A naive Bayes classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points. Popular uses of naive Bayes classifiers include spam filters, text analysis and medical diagnosis. These classifiers are widely used for machine learning because they are simple to implement. Naive Bayes is also known as simple Bayes or independence Bayes.

Neural Networks

Artificial neural networks, usually simply called neural networks, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain

3. RESULTS:

ALGORITHM	ACCURACY RATE
KNN	0.72(72%)
SVC	0.70(72%)
Decision Tree	0.89(89%)
Random Forest	0.90(90%)
Logistic regression	0.90(90%)

Table 6. Accuracy

The algorithms that are used in the given model are K-Nearest Neighbor algorithm, random forest, SVC,Decision tree. On comparision of the above algorithms accuracy rate for the SVC is more than the other algorithms. The above table shows the accuracy rates of the algorithms used.

4. CONCLUSION:

The main purpose of the project is to make an efficient model that would classify success of start up companies based on the input features provided .With the use of binary classification we can predict whether a company would be successful or not. We have used 5 models - Logistic regression, KNN, Random Forest Classifier, Decision Tree Classifier and SVC.SVC provides better accuracy compared to others we used 4 main metrics for prediction : age of first funding , age of last funding year, age at which company achieved its first milestone, is it in top 500 or not. The model with best accuracy is used to find the success rate.

8.REFERENCES:

[1] https://startup-prediction.herokuapp.com/

[2]https://www.google.com/url?sa=t&source=web&rct=j&url=https://github.com/RamkishanPanthena/Startup-Success-Prediction&ved=2ahUKEwi91ZfQgtH0AhWV_XMBHVoXCvQQFnoECDIQAQ&usg=AOvVaw298k4PtURYlpuEIQfVG8JG

[3]https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3878-2019.pdf&ved=2ahUKEwi91ZfQgtH0AhWV_XMBHVoXCvQQFnoECA4QAQ&usg=AOvVaw3TTNRVoEVuMz7qCWqODIuk

[4] Graham,P.(2012). Startups equal growth. Web page.

[5]https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.kaggle.com/manishkc06/startup-success-prediction&ved=2ahUKEwjN96DNn9H0AhUWIbcAHTW9CpsQFnoECC8QAQ&usg=AOvVaw0eN-0y_TinT96XAKORBM5C

[6] https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148