**Startup Unicorn Prediction Using Advanced**

**Machine Learning Algorithms**

**1 INTRODUCTION**

Artificial intelligence is an emerging topic and will soon be able to perform administrative decisions faster, better, and at a lower cost than humans. Machines can consistently process large amount of unstructured data to identify pattern and make predictions on future events (e.g. Agrawal and Dhar 2014; Baesens et al. 2016). In more complex and creative contexts such as innovation and entrepreneurship, however, the question remains whether machines are superior to humans. Machines fail in two kinds of situations: processing and interpreting “soft” types of information (information that cannot be quantified) (Petersen 2004) and making predictions in “unknowable risk” situations of extreme uncertainty that require intuitive decision making. In such situations, the machine does not have representative information for a certain outcome and over fits on training data at cost of the live performance of a learner (Attenberg et al. 2015).

One example where both “soft” information signals as well as “unknowable risk” are crucial, is predicting the success of early stage startups. In this case, angel investors, informal investors that devote their private equity in new ventures, face the challenge to decide if the startup at hand is worth financing or not. Angel investors often make decisions before neither the feasibility of a new product nor the existence of a market is proven (Maxwell et al. 2011). In such contexts, angel investors do simply not have enough information to assess the quality of a startup and thus predict the probability of its future success (Dutta and Folta 2016). Moreover, such information might simply not exist under conditions of extreme uncertainty and make the outcome thus unknowable. Consequently, predictions are made for ideas that serve markets, which do not yet exist or novel technologies, where feasibility is still unknown but may provide great returns of investment (Alvarez and Barney 2007). Nevertheless, identifying such unicorns, startups that are highly innovative, disrupt traditional industries, and offer tremendous return is highly relevant.

In these situations, humans are still the “gold standard” for processing “soft” signals that cannot easily be quantified into models such as creativity, innovativeness etc. (Baer and McKool 2014) and make use of an affective judgment tool to recognize pattern in previous decisions: intuition (Huang and Pearce 2015). Using one´s gut feeling proved to be a valuable strategy to deal with extreme uncertainty. However, individual human judges are tainted by bounded rationality in making predictions, which emphasizes that instead of optimizing every decision, humans tend to rely on heuristics (i.e. mental shortcuts) and thus rather focus on highly accessible information (Simon 1955; Kahneman 2011). This often leads to biased interpretation (cognitive processes that involve erroneous assumptions) and may finally result in disastrous predictions (Busenitz and Barney 1997). To solve this problem, research in the field of human computation provides a valuable solution: utilizing the “wisdom of crowds” through collective intelligence (e.g. Brynjolfsson et al. 2016; Larrick et al. 2011; van Bruggen et al. 2010). This is a suitable approach to leverage the benefits of humans in prediction tasks, such as providing subjective evaluation of variables that are difficult to measure objectively through machines (e.g. innovativeness) (Colton and Wiggins 2012) or using their prior domain-specific knowledge to make intuitive decision (Blattberg and Hoch 1990). The aggregation of knowledge and resulting predictions than eliminates the statistical errors of individual human decision makers (Larrick et al. 2011). While each of the methods might work well in separation, weargue that combining the complementary capabilities of humans and machines in a Hybrid Intelligence approach allows to make predictions in contexts of extreme uncertainty such as the case of early startup success through applying formal analysis of “hard” information as well as intuitive decision-making processing also “soft” information

The aim of this research is to develop a method to predict the probability of success of early stage startups. Therefore, we follow a design science research approach (Hevner 2007; Gregor and Hevner 2013) to develop a Hybrid Intelligence method that combines the strength of both machine intelligence such as machine learning techniques to access, process, and structure large amount of information as well as collective intelligence, which uses the intuition and creative potential of individuals while reducing systematic errors through statistical averaging in an ensemble approach (Shmueli and Koppius 2011). We, thus, intend to show that a hybrid approach improves predictions for the success of startups under extreme uncertainty compared to machine or human only methods. As we proceed our research we will empirically test our proposed method to provide and validate a practical solution for predicting firm success under conditions of extreme uncertainty.

Within the scope of this paper, we first developed a taxonomy of signals that are potential predictors for the success of early stage startups based on previous work and domain knowledge. We then designed a method that uses these predictors as input for both machine learning algorithms as well as collective intelligence to individually assess the probability of success and then weights and aggregates the results to a combined prediction outcome. Moreover, we provide an outlook on the next steps of our research project.

**Objective of the project:**

Artificial intelligence is an emerging topic and will soon be able to perform decisions better than humans. In more complex and creative contexts such as innovation, however, the question remains whether machines are superior to humans. Machines fail in two kinds of situations: processing and interpreting “soft” information (information that cannot be quantified) and making predictions in “unknowable risk” situations of extreme uncertainty. In such situations, the machine does not have representative information for a certain outcome. Thereby, humans are still the “gold standard” for assessing “soft” signals and make use intuition. To predict the success of startups, we, thus, combine the complementary capabilities of humans and machines in machine learning algorithms like Gradient Boosting, SVM, Random Forest and Decision Tree. All ML algorithms will be trained on past performance of STARTUP dataset and then this trained model can be used to predict success or failure of new STARTUP TEST DATA.

**2. LITERATURE SURVEY**

**A Model of the Internet as Creative Destroyer**

The extent to which a technological change is a creative destroyer is of interest to entrepreneurs who can exploit the opportunity and to incumbents who must defend their existing competitive advantages from the change. In the face of a technological change, an important question is: To what extent is it a creative destroyer? In this paper, we offer a model for exploring the depth and breadth of creative destruction from the Internet and the implications for wealth creation and competitive advantage. We apply the model to three groups of industries, each of which rests on one of Thompson's three categories of organizational technologies: long-linked; mediating; and intensive. The application suggests that incumbents in all industries should experience some erosion of competitive advantage. Industries with predominantly mediating technologies should experience creative destruction. Those with intensive technologies should experience more erosion of competitive advantage than those with long-linked technologies.

**Editorial —Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research**

We address key questions related to the explosion of interest in the emerging fields of big data, analytics, and data science. We discuss the novelty of the fields and whether the underlying questions are fundamentally different, the strengths that the information systems (IS) community brings to this discourse, interesting research questions for IS scholars, the role of predictive and explanatory modeling, and how research in this emerging area should be evaluated for contribution and significance.

**Discovery and creation: alternative theories of entrepreneurial action**

Do entrepreneurial opportunities exist, independent of the perceptions of entrepreneurs, just waiting to be discovered? Or, are these opportunities created by the actions of entrepreneurs? Two internally consistent theories of how entrepreneurial opportunities are formed – discovery theory and creation theory – are described. While it will always be possible to describe the formation of a particular opportunity as an example of a discovery or creation process, these two theories do have important implications for the effectiveness of a wide variety of entrepreneurial actions in different contexts. The implications of these theories for seven of these actions are described, along with a discussion of some of the broader theoretical implications of these two theories for the fields of entrepreneurship and strategic management. Copyright © 2007 Strategic Management Society.

**Deriving the Pricing Power of Product Features by Mining Consumer Reviews**

Increasingly, user-generated product reviews serve as a valuable source of information for customers making product choices online. The existing literature typically incorporates the impact of product reviews on sales based on numeric variables representing the valence and volume of reviews. In this paper, we posit that the information embedded in product reviews cannot be captured by a single scalar value. Rather, we argue that product reviews are multifaceted, and hence the textual content of product reviews is an important determinant of consumers' choices, over and above the valence and volume of reviews. To demonstrate this, we use text mining to incorporate review text in a consumer choice model by decomposing textual reviews into segments describing different product features. We estimate our model based on a unique data set from Amazon containing sales data and consumer review data for two different groups of products (digital cameras and camcorders) over a 15-month period. We alleviate the problems of data sparsity and of omitted variables by providing two experimental techniques: clustering rare textual opinions based on pointwise mutual information and using externally imposed review semantics. This paper demonstrates how textual data can be used to learn consumers' relative preferences for different product features and also how text can be used for predictive modeling of future changes in sales.

**Distilling the Wisdom of Crowds: Prediction Markets vs. Prediction Polls**

We report the results of the first large-scale, long-term, experimental test between two crowd sourcing methods – prediction markets and prediction polls. More than 2,400 participants made forecasts on 261 events over two seasons of a geopolitical prediction tournament. Some forecasters traded in a continuous double auction market and were ranked based on earnings. Others submitted probability judgments, independently or in teams, and were ranked based on Brier scores. In both seasons of the tournament, last day prices from the prediction market were more accurate than the simple mean of forecasts from prediction polls. However, team prediction polls outperformed prediction markets when poll forecasts were aggregated with algorithms using temporal decay, performance weighting and recalibration. The biggest advantage of prediction polls occurred at the start of long-duration questions. Prediction polls with proper scoring, algorithmic aggregation and teaming offer an attractive alternative to prediction markets for distilling the wisdom of crowds.

**The Gold Standard for Assessing Creativity**

The most widely used creativity assessments are divergent thinking tests, but these and other popular creativity measures have been shown to have little validity. The Consensual Assessment Technique is a powerful tool used by creativity researchers in which panels of expert judges are asked to rate the creativity of creative products such as stories, collages, poems, and other artifacts. The Consensual Assessment Technique is based on the idea that the best measure of the creativity of a work of art, a theory, a research proposal, or any other artifact is the combined assessment of experts in that field. Unlike other measures of creativity, the Consensual Assessment Technique is not based on any particular theory of creativity, which means that its validity (which has been well established empirically) is not dependent upon the validity of any particular theory of creativity. The Consensual Assessment Technique has been deemed the “gold standard” in creativity research and can be very useful in creativity assessment in higher education.

**Transformational issues of big data and analytics in networked business**

The era of big data and analytics is upon us and is changing the world dramatically. The field of Information Systems should be at the forefront of understanding and interpreting the impact of both technologies and management so as to lead the efforts of business research in the big data era. We need to prepare ourselves and our students for this changing world of business. In this discussion, we focus on exploring the technical and managerial issues of business transformation resulting from the insightful adoption and innovative applications of data sciences in business. We end by providing an overview of the papers included in this special issue and outline future research directions.

**Rate or Trade? Identifying Winning Ideas in Open Idea Sourcing**

Information technology (IT) has created new patterns of digitally-mediated collaboration that allow open sourcing of ideas for new products and services. These novel sociotechnical arrangements afford finely-grained manipulation of how tasks can be represented and have changed the way organizations ideate. In this paper, we investigate differences in behavioral decision-making resulting from IT-based support of open idea evaluation. We report results from a randomized experiment of 120 participants comparing IT-based decision-making support using a rating scale (representing a judgment task) and a preference market (representing a choice task). We find that the rating scale-based task invokes significantly higher perceived ease of use than the preference market-based task and that perceived ease of use mediates the effect of the task representation treatment on the users’ decision quality. Furthermore, we find that the understandability of ideas being evaluated, which we assess through the ideas’ readability, and the perception of the task’s variability moderate the strength of this mediation effect, which becomes stronger with increasing perceived task variability and decreasing understandability of the ideas. We contribute to the literature by explaining how perceptual differences of task representations for open idea evaluation affect the decision quality of users and translate into differences in mechanism accuracy. These results enhance our understanding of how crowd sourcing as a novel mode of value creation may effectively complement traditional work structures.

**The Business Model DNA: Towards an Approach for Predicting Business Model Success**

Business models have gained much interest in the last decade to analyze the potential of new business ventures or possible innovation paths of existing businesses. However, the business model concept has only rarely been used as basis for quantitative empirical studies. This paper suggests the concept of a Business Model DNA to describe the characteristics of specific business models. This concept allows to analyze business models in order to identify clusters of business models that outperform others and calculate future prospects of specific business models. We used 181 startups from the USA and Germany and applied data mining techniques, i.e. cluster analysis and Support Vector Machines, to classify different business models in regards to their performance. Our findings show that 12 distinct business model clusters with different growth expectations and chances of survival exist. We can predict the survival of a venture with an accuracy of 83.6 %.

**Crowd-Squared: Amplifying the Predictive Power of Search Trend Data**

Big Data generated by crowds provides a myriad of opportunities for monitoring and modeling people's intentions, preferences, and opinions. A crucial step in analyzing such “big data” is selecting the relevant part of the data that should be provided as input to the modeling process. In this paper, we offer a novel, structured, crowd-based method to address the data selection problem in a widely used and challenging context: selecting search trend data. We label the method “crowd-squared,” as it leverages crowds to identify the most relevant terms in search volume data that were generated by a larger crowd. We empirically test this method in two domains and find that our method yields predictions that are equivalent or superior to those obtained in previous studies (using alternative data selection methods) and to predictions obtained using various benchmark data selection methods. These results emphasize the importance of a structured data selection method in the prediction process, and demonstrate the utility of the crowd-squared approach for addressing this problem in the context of prediction using search trend data.

**3 .SYSTEM ANALYSIS**

**3.1 Existing System**

One way towards understanding predictions in uncertain situations is to examine the mental processes that underlies the cognitive decision-making process. A theory that is particularly helpful in this context is the dual process theory of decision making. The underlying assumption of this theory is that people make use of two cognitive modes, one is characterized by intuition (system 1) and one by deliberate analytical predictions (system 2) (Tversky and Kahneman 1983; Kahneman 2011).

Predicting the success of early stage ventures is extremely complex and uncertain because frequently just vague ideas are prevalent, prototypes do not yet exist and thus the proof of concept is still pending. Moreover, such ideas might even not have a market yet, but offer great potential of growth in the future (Alvarez and Barney 2007). Consequently, the decision-making context is highly uncertain as neither possible outcomes nor the probability of such are known. This fact can be explained through two concepts: information asymmetry and unknowable risk (Alvarez and Barney 2007; Huang and Pearce 2015)

Information asymmetry describes situation, in which forecasters have incomplete information to decide (Spence 1974). When perfect information is absent, decision makers tend to search for various indicators that signal the likeliness of future outcomes (Morris 1987). In our context, such signals include both “hard” signals that can be easily quantified and categorized (e.g. industry, technology, team size) as well as “soft” signals (e.g. innovativeness, personality of entrepreneur). Humans then try to apply formal analysis to gather signals that support them in making deliberate, rule-based system 2 decisions (Kahneman 2011). On the other hand, unknowable risk defines situations in which a decision maker cannot gather information that signal a potential outcome or make decisions based on formal analysis because the simply not exist. This may be best compared to the error term of a statistical Bayesian model. Unknowable risk covers unexpected events that describe a deviation from status quo (Kaplan and March 1988). In our context, this means for instance identifying a unicorn startup that gains enormous return that only few would have expected. Formal analytics are not working in these contexts, as representative cases might be missing in previous experience. In such situations, where humans “don’t know what they don’t know”, decision making is mainly based on intuition (system 1) rather than formal analysis (Tversky and Kahneman 1983; Huang and Pearce 2015). Thus, predicting the success of early stage startups is a challenging task and the costs of misclassification are high as they might lead to disastrous funding decisions or missing valuable chances for return (Attenberg et al. 2015). Previous research in the context of early stage ventures provides strong evidence the best performance in terms of accuracy are provided by combining both types of predictions: analytical (system 2) and intuitive (system 1) (Huang and Pearce 2015).

**PROPOSED SYSTEM**

The objective of the project is to predict whether a startup which is currently operating turn into a success or a failure. The success of a company is defined as the event that gives the company's founders a large sum of money through the process of M&A (Merger and Acquisition) or an IPO (Initial Public Offering). A company would be considered as failed if it had to be shutdown.

To implement this project and predict success or failure of startup we have used advanced machine learning algorithms such as Gradient Boosting, SVM, Random Forest and Decision Tree. All ML algorithms will be trained on past performance of STARTUP dataset and then this trained model can be used to predict success or failure of new STARTUP TEST DATA.

**MODULES**

To implement this project we have designed following modules

**Upload Startup Dataset:** using this module we will upload dataset to application

**Preprocess Dataset:** using this module we will process data to remove missing values and then split dataset into train and test where ML algorithms will take 80% dataset records to trained themselves and 20% records will be used to perform prediction and calculate accuracy

**Run Decision Tree Algorithm:** using this module we will trained decision tree algorithm

**Run SVM & Random Forest Algorithm:** using this module we will trained Random Forest and SVM algorithms

**Run Gradient Boosting Algorithm:** using this module we will trained Gradient Boosting algorithm

**Comparison Graph:** using this module we will evaluate performance of each algorithms and plot comparison graph of accuracy, precision, recall and FSCORE

**Predict Startup Status from Test Data:** using this module we will upload test data and then ML algorithms will analyse test data and predict whether that company or startup data show signs of success of failure

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**



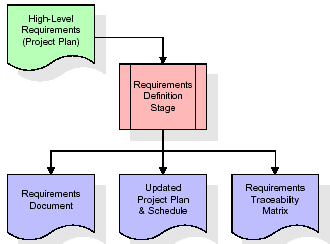
SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

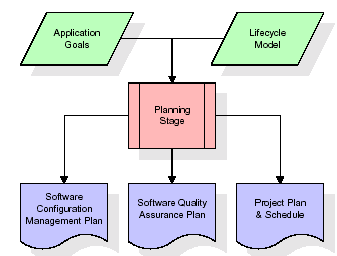
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

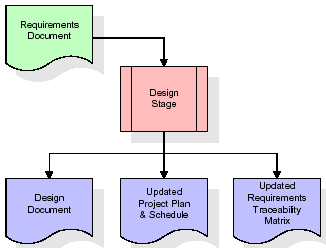
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

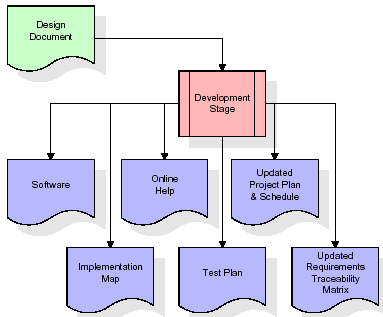
**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

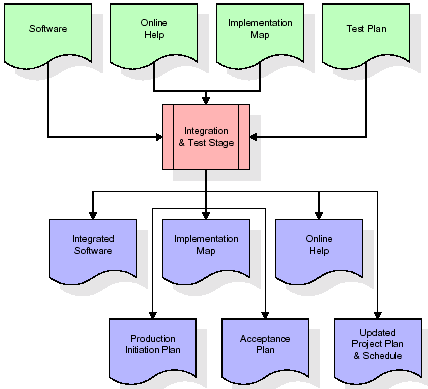
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

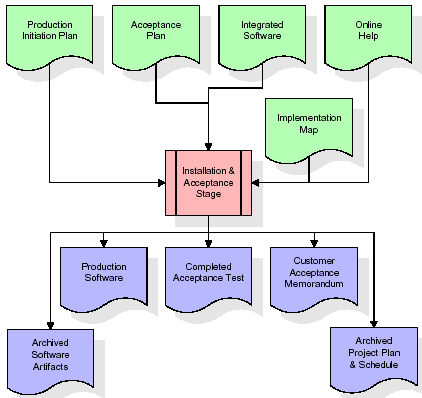


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labour data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behaviour of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms *what* must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**Operating Environment**

Windows XP.

**HARDWARE REQUIREMENTS:**

# Processor - Pentium –IV

* Speed - 1.1 Ghz
* RAM - 256 MB(min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows7/8
* Programming Language - Python

**4. SYSTEM DESIGN**

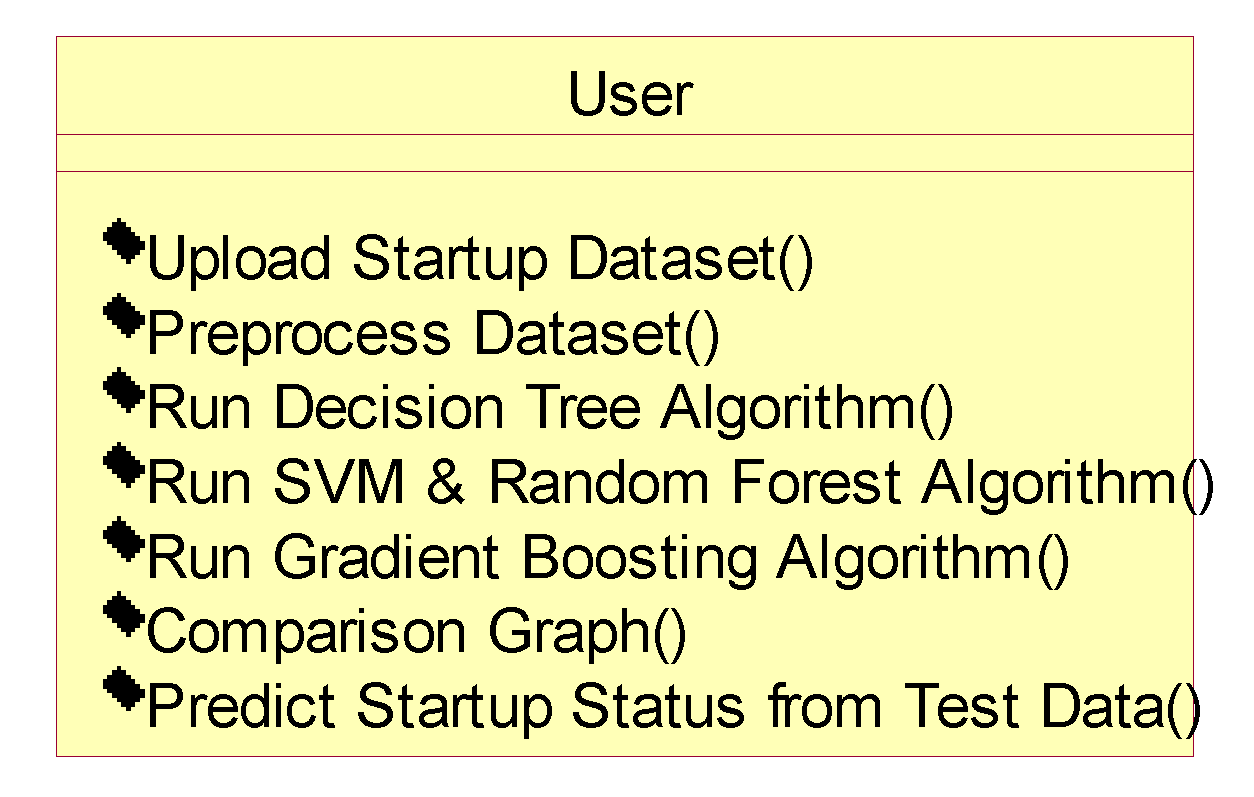
**UML Diagram:**

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

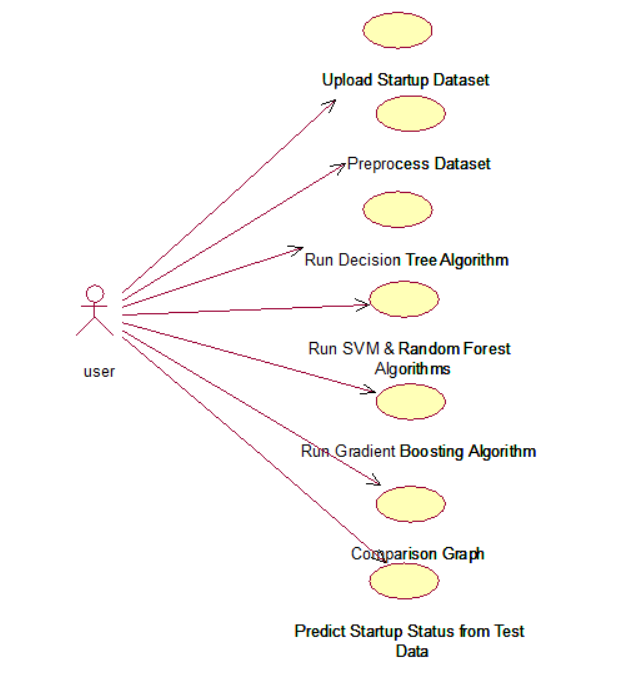
**Class Diagram:**



**Use case Diagram:**

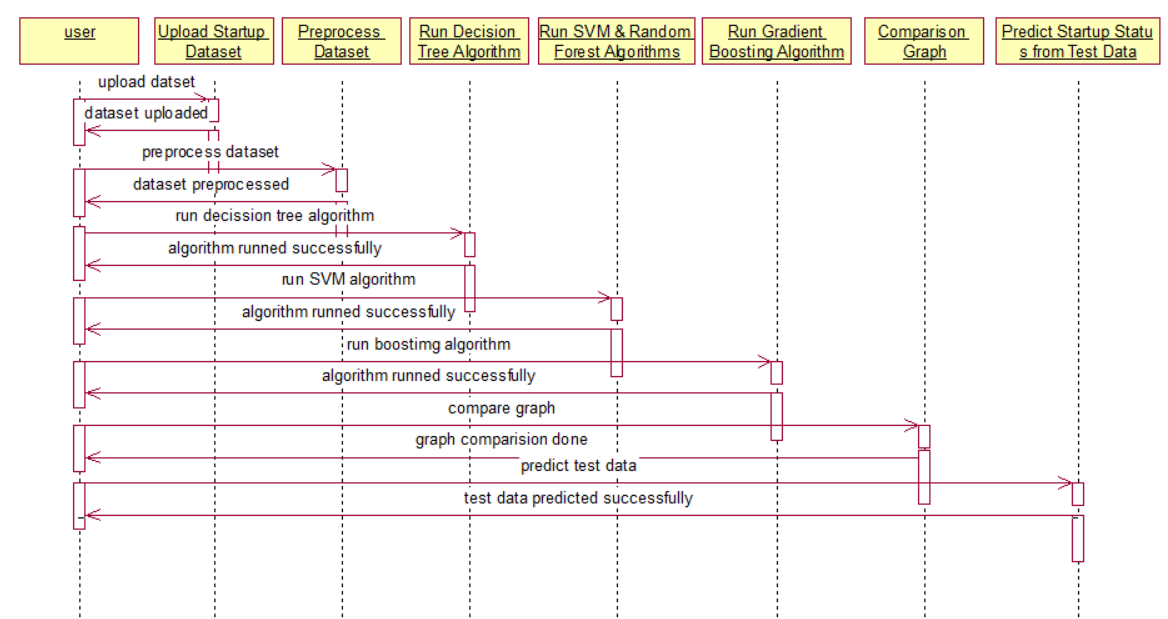
A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

**Use case Diagram:**



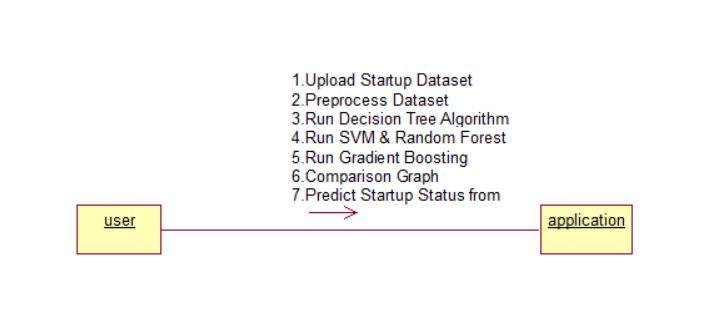
**Sequence diagram:**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.



**Collaboration diagram:**

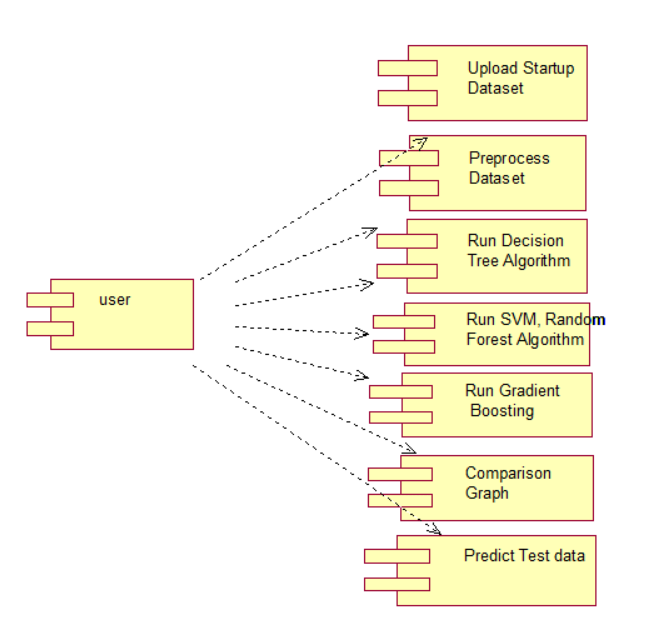
A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.



**Component Diagram:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

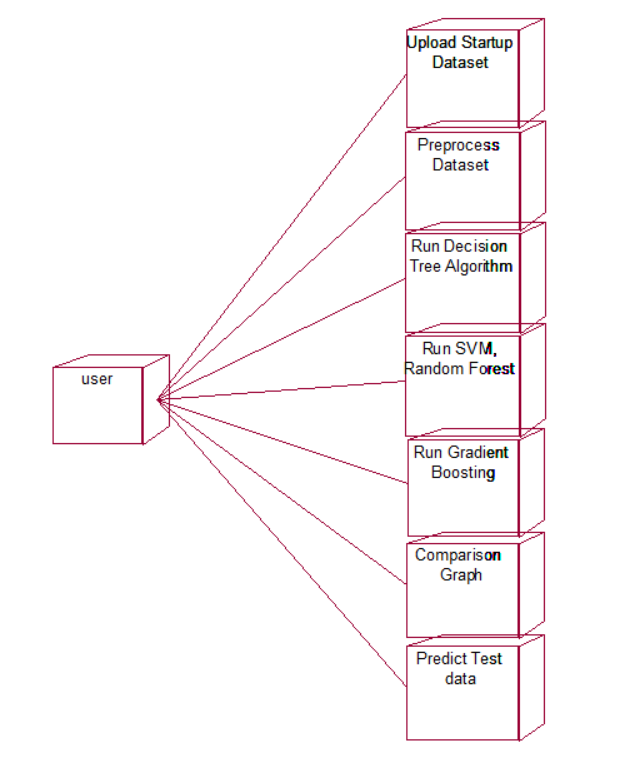
****



**Deployment Diagram:**

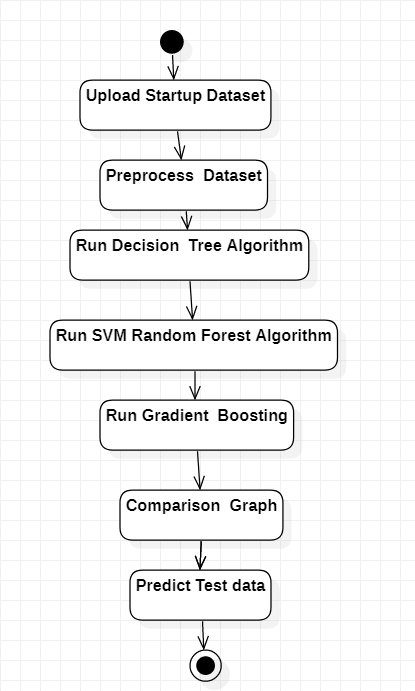
A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**Activity Diagram:**

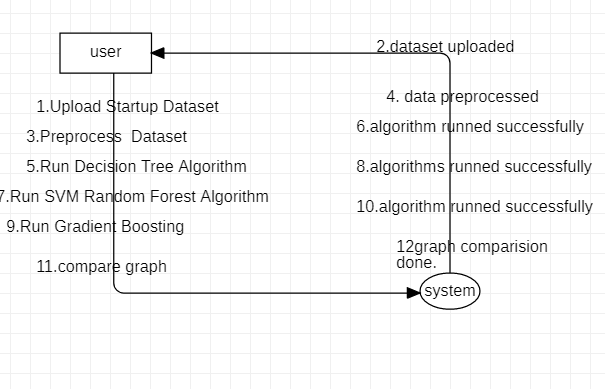
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent



**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

**Main.py**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

import os

import pandas as pd

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

import pickle

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn import svm

main = tkinter.Tk()

main.title("Startup Unicorn Prediction Using Advanced Machine Learning Algorithms") #designing main screen

main.geometry("1300x1200")

global filename

global X, Y

global X\_train, Y\_train

global classifier

global accuracy, precision, recall, fscore

global X\_train, X\_test, y\_train, y\_test

global dataset

def uploadDataset():

global filename

global dataset

filename = filedialog.askopenfilename(initialdir = "Dataset")#uploading dataset

dataset = pd.read\_csv(filename) #reading dataset from loaded file

dataset.fillna(0, inplace = True)

text.delete('1.0', END)

text.insert(END,filename+' Loaded\n\n')

text.insert(END,str(dataset.head()))

label = dataset.groupby('labels').size()

label.plot(kind="bar")

plt.show()

def preprocessDataset():

global dataset

global X, Y

global X\_train, X\_test, y\_train, y\_test

text.delete('1.0', END)

X = dataset[['age\_first\_funding\_year', 'age\_last\_funding\_year', 'relationships', 'funding\_rounds',

'funding\_total\_usd', 'milestones', 'is\_CA', 'is\_NY', 'is\_MA', 'is\_TX',

'is\_otherstate', 'is\_software', 'is\_web', 'is\_mobile', 'is\_enterprise',

'is\_advertising', 'is\_gamesvideo', 'is\_ecommerce', 'is\_biotech',

'is\_consulting', 'is\_othercategory', 'has\_VC', 'has\_angel', 'has\_roundA',

'has\_roundB', 'has\_roundC', 'has\_roundD', 'avg\_participants',

'is\_top500', 'age\_first\_milestone\_year','age\_last\_milestone\_year','labels'

]]

Y = dataset['labels']

X = X.values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=0)

text.insert(END,"Total records available in dataset: "+str(X.shape[0])+"\n")

text.insert(END,"Total features/columns found in dataset: "+str(X.shape[1])+"\n\n")

text.insert(END,"Dataset train and test split details\n\n")

text.insert(END,"Training dataset split 80% total records are: "+str(X\_train.shape[0])+"\n")

text.insert(END,"Testing dataset split 20% total records are: "+str(X\_test.shape[0])+"\n")

def calculateMetrics(y\_test,predict,name):

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

a = accuracy\_score(y\_test,predict)\*100

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

text.insert(END,name+" Accuracy : "+str(a)+"\n")

text.insert(END,name+" Precision : "+str(p)+"\n")

text.insert(END,name+" Recall : "+str(r)+"\n")

text.insert(END,name+" FSCORE : "+str(f)+"\n\n")

def runDecisionTree():

global accuracy, precision, recall, fscore

global X\_train, X\_test, y\_train, y\_test

accuracy = []

precision = []

recall = []

fscore = []

text.delete('1.0', END)

dt\_cls = DecisionTreeClassifier(max\_depth=2,criterion="entropy")

dt\_cls.fit(X\_train, y\_train)

predict = dt\_cls.predict(X\_test)

calculateMetrics(y\_test,predict,"Decision Tree Algorithm")

def runRF():

global classifier

global accuracy, precision, recall, fscore

global X\_train, X\_test, y\_train, y\_test

rf\_cls = RandomForestClassifier(n\_estimators=2,criterion="entropy",max\_features="sqrt")

rf\_cls.fit(X\_train, y\_train)

predict = rf\_cls.predict(X\_test)

calculateMetrics(y\_test,predict,"Random Forest Algorithm")

classifier = rf\_cls

svm\_cls = svm.SVC(probability=True)

svm\_cls.fit(X\_train, y\_train)

predict = svm\_cls.predict(X\_test)

calculateMetrics(y\_test,predict,"SVM Algorithm")

def runGB():

global accuracy, precision, recall, fscore

global X\_train, X\_test, y\_train, y\_test

gb = GradientBoostingClassifier()

gb.fit(X\_train, y\_train)

predict = gb.predict(X\_test)

calculateMetrics(y\_test,predict,"Gradient Boosting Algorithm")

def graph():

df = pd.DataFrame([['Decision Tree','Precision',precision[0]],['Decision Tree','Recall',recall[0]],['Decision Tree','F1 Score',fscore[0]],['Decision Tree','Accuracy',accuracy[0]],

['Random Forest','Precision',precision[1]],['Random Forest','Recall',recall[1]],['Random Forest','F1 Score',fscore[1]],['Random Forest','Accuracy',accuracy[1]],

['SVM','Precision',precision[2]],['SVM','Recall',recall[2]],['SVM','F1 Score',fscore[2]],['SVM','Accuracy',accuracy[2]],

['Gradient Boosting','Precision',precision[3]],['Gradient Boosting','Recall',recall[3]],['Gradient Boosting','F1 Score',fscore[3]],['Gradient Boosting','Accuracy',accuracy[3]],

],columns=['Parameters','Algorithms','Value'])

df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')

plt.show()

def predict():

text.delete('1.0', END)

global classifier

testfile = filedialog.askopenfilename(initialdir = "Dataset")

testdata = pd.read\_csv(testfile)

testdata.fillna(0, inplace = True)

testdata = testdata[['age\_first\_funding\_year', 'age\_last\_funding\_year', 'relationships', 'funding\_rounds',

'funding\_total\_usd', 'milestones', 'is\_CA', 'is\_NY', 'is\_MA', 'is\_TX',

'is\_otherstate', 'is\_software', 'is\_web', 'is\_mobile', 'is\_enterprise',

'is\_advertising', 'is\_gamesvideo', 'is\_ecommerce', 'is\_biotech',

'is\_consulting', 'is\_othercategory', 'has\_VC', 'has\_angel', 'has\_roundA',

'has\_roundB', 'has\_roundC', 'has\_roundD', 'avg\_participants',

'is\_top500', 'age\_first\_milestone\_year','age\_last\_milestone\_year','labels'

]]

testdata = testdata.values

predict = classifier.predict(testdata)

for i in range(len(testdata)):

if predict[i] == 1:

text.insert(END,"Startup Test Values: "+str(testdata[i])+" =====> Predicted As SUCCESS\n\n")

else:

text.insert(END,"Startup Test Values: "+str(testdata[i])+" =====> Predicted As FAILURE\n\n")

font = ('times', 16, 'bold')

title = Label(main, text="Startup Unicorn Prediction Using Advanced Machine Learning Algorithms")

title.config(bg='darkviolet', fg='gold')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=50,y=120)

text.config(font=font1)

font1 = ('times', 12, 'bold')

uploadButton = Button(main, text="Upload Startup Dataset", command=uploadDataset)

uploadButton.place(x=50,y=550)

uploadButton.config(font=font1)

preprocessButton = Button(main, text="Preprocess Dataset", command=preprocessDataset)

preprocessButton.place(x=360,y=550)

preprocessButton.config(font=font1)

dtButton = Button(main, text="Run Decision Tree Algorithm", command=runDecisionTree)

dtButton.place(x=580,y=550)

dtButton.config(font=font1)

rfButton = Button(main, text="Run SVM & Random Forest Algorithm", command=runRF)

rfButton.place(x=50,y=600)

rfButton.config(font=font1)

gbButton = Button(main, text="Run Gradient Boosting Algorithm", command=runGB)

gbButton.place(x=360,y=600)

gbButton.config(font=font1)

graphButton = Button(main, text="Comparison Graph", command=graph)

graphButton.place(x=50,y=650)

graphButton.config(font=font1)

predictButton = Button(main, text="Predict Startup Status from Test Data", command=predict)

predictButton.place(x=360,y=650)

predictButton.config(font=font1)

main.config(bg='turquoise')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

**Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

**Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

**System Testing**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

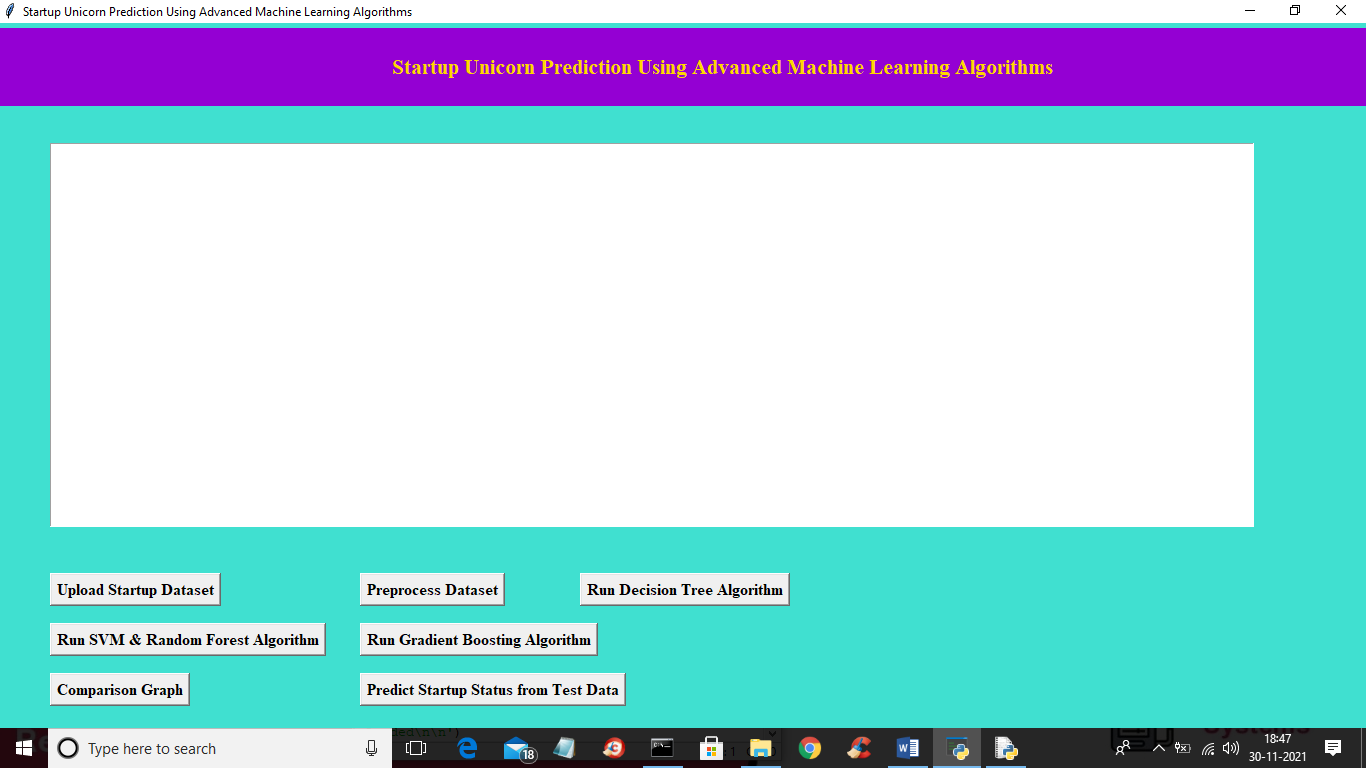
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

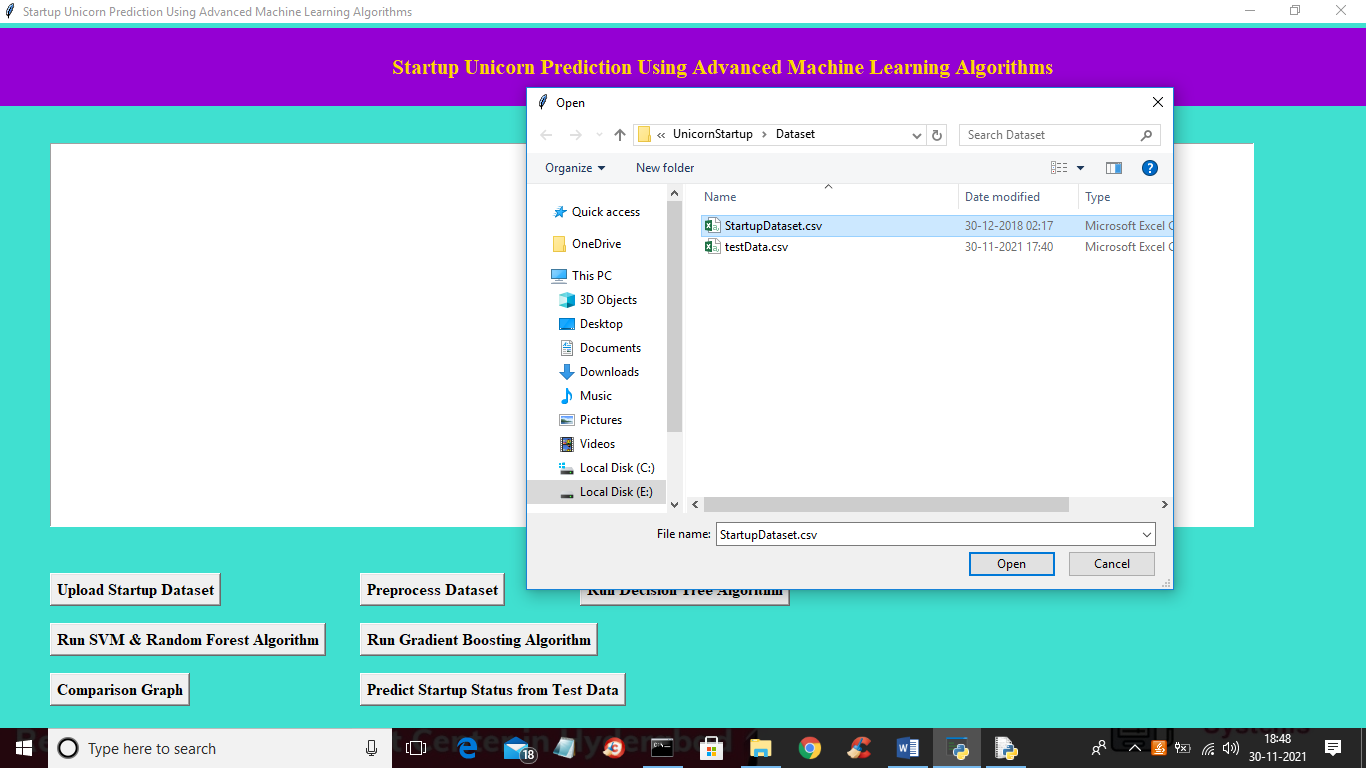
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Expected** | **Actual** |
| 01 | Upload Startup Dataset | Test whether the Startup Dataset is uploaded  or not | If Startup Dataset  Is not uploaded | we cannot do further operations | Startup Dataset Is uploaded we can do further operations | High | High |
| 02 | Pre-process Dataset | Verify the Dataset is Pre-processed  or not | Without loading the dataset | We cannot Pre-process Dataset | We Can Pre-process  Dataset successfully | High | High |
| 03 | Run Decision Tree Algorithm | Verify the Run Decision Tree will run or not | Without training model | we cannot Run Decision Tree Algorithm | we can run Run Decision Tree |  |  |
| 04 | Run SVM & Random Forest Algorithms | Verify the Run SVM & Random Forest Algorithms will run or not | Without training model | we cannot Run SVM & Random Forest Algorithms | we can Run SVM & Random Forest Algorithms | High | High |
| 05 | Run Gradient Boosting Algorithm | Verify the Gradient Boosting Algorithm will run or not | Without training model | we cannot run Gradient Boosting Algorithm | we can run Gradient Boosting Algorithm | High | High |
| 06 | Comparison Graph | compare Accuracy Comparison Graph | Without  Comparing Graph | We cannot get accuracy results | We can get accuracy results | High | High |
| 07 | Predict Startup Status from Test Data | Predict the results from test data | Without predicting result | we cannot do further operations | we can do further operations | High | High |

**7. SCREENSHOTS:**

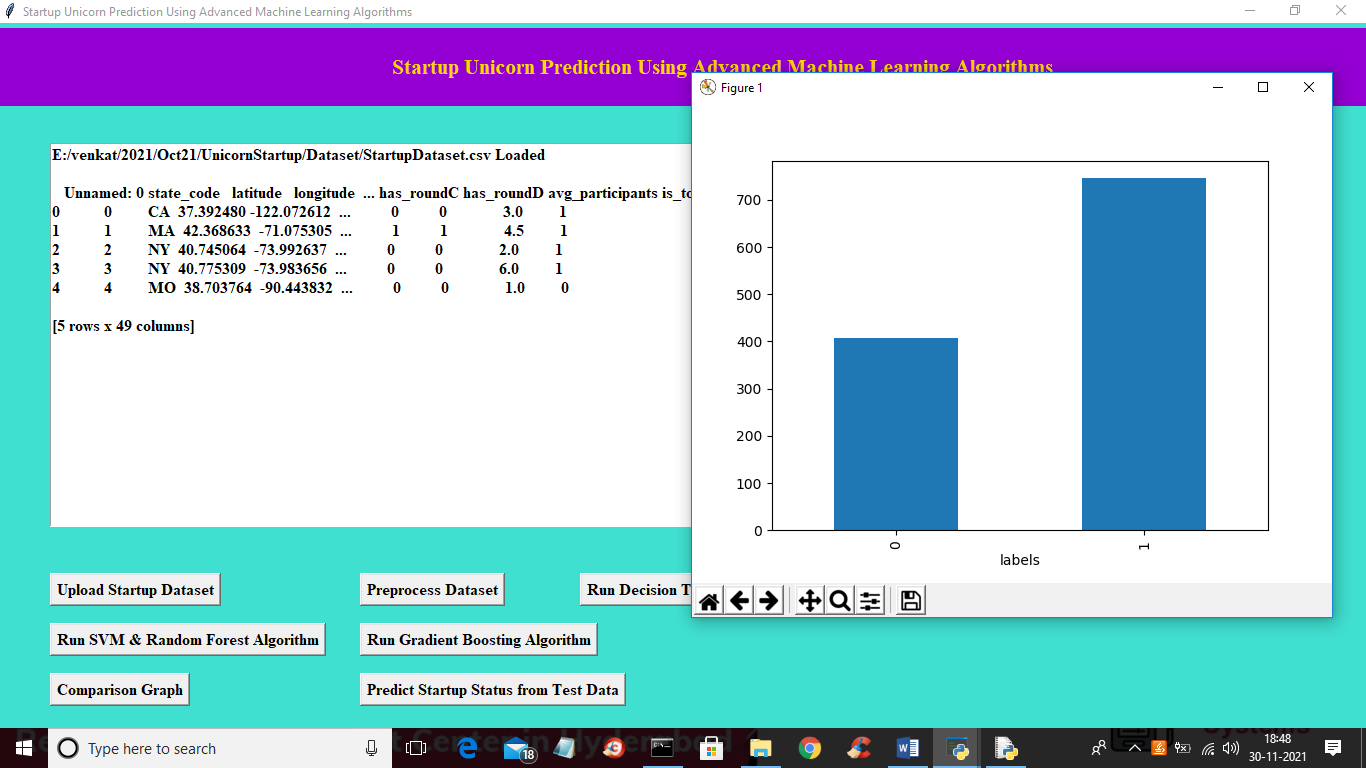
To run project double click on ‘run.bat’ file to get below screen



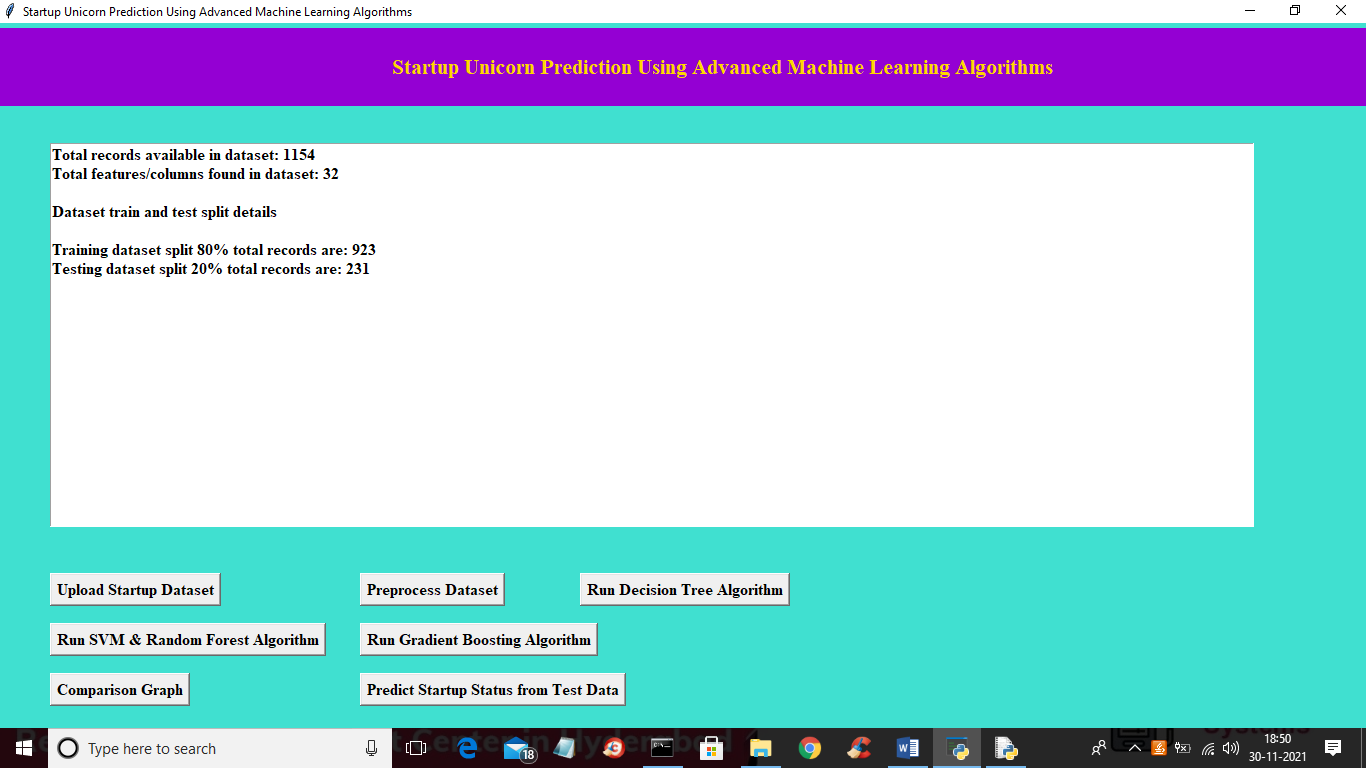
In above screen click on ‘Upload Startup Dataset’ button to upload dataset and to get below screen



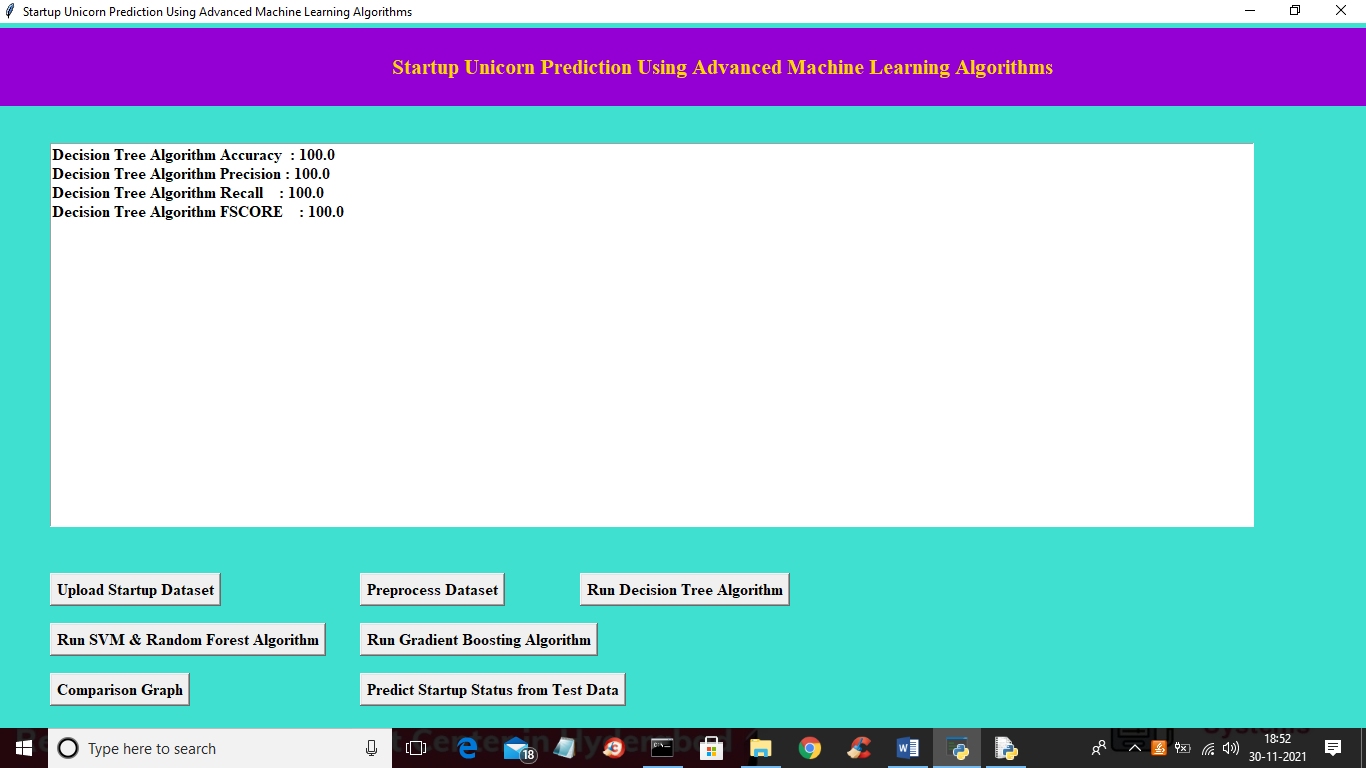
In above screen selecting and uploading ‘StartupDataset.csv’ file and then click on “Open” button to load dataset and to get below screen



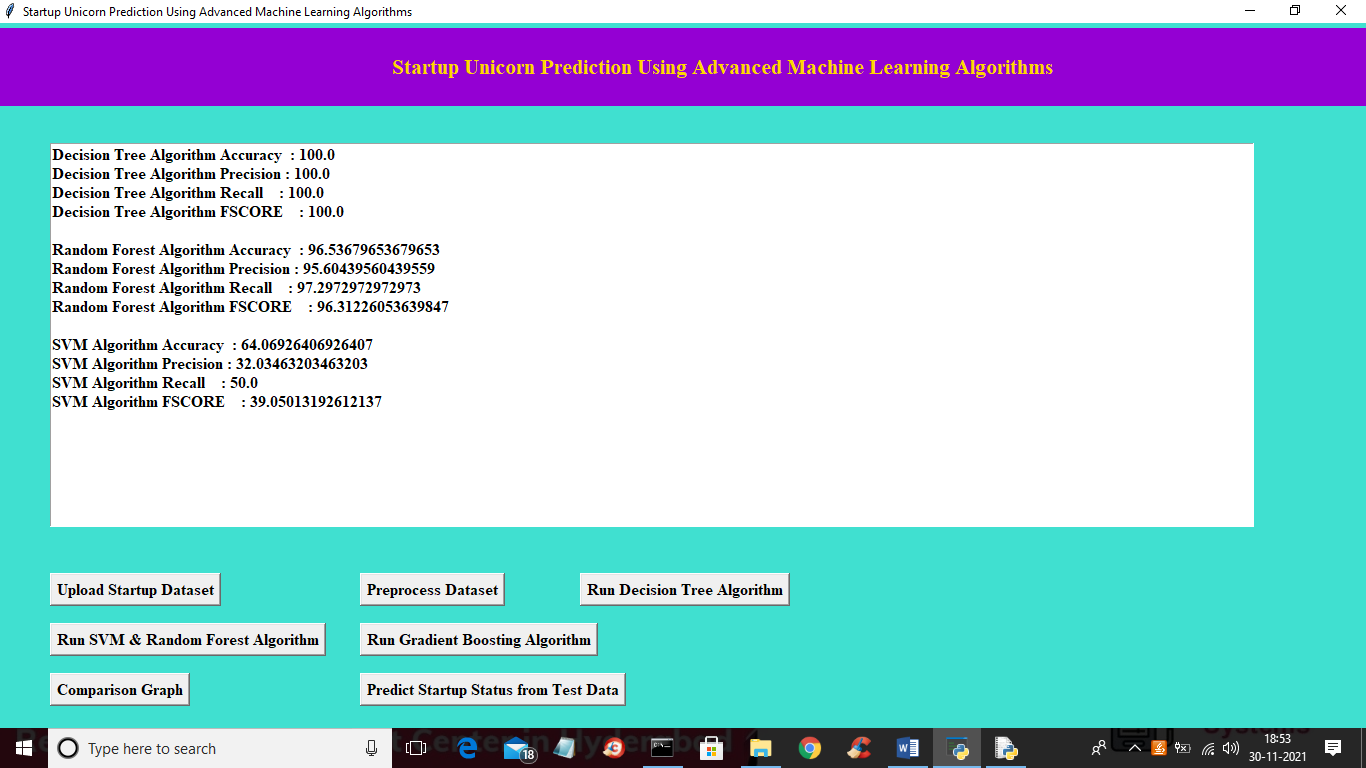
In above screen dataset loaded and I am displaying few values from dataset and in graph x-axis represents values 0 (failure) and 1 (success) and y-axis represents number of records available under that label. Now close above graph and then click on ‘Preprocess Dataset’ button to remove missing values and then split dataset into train and test part



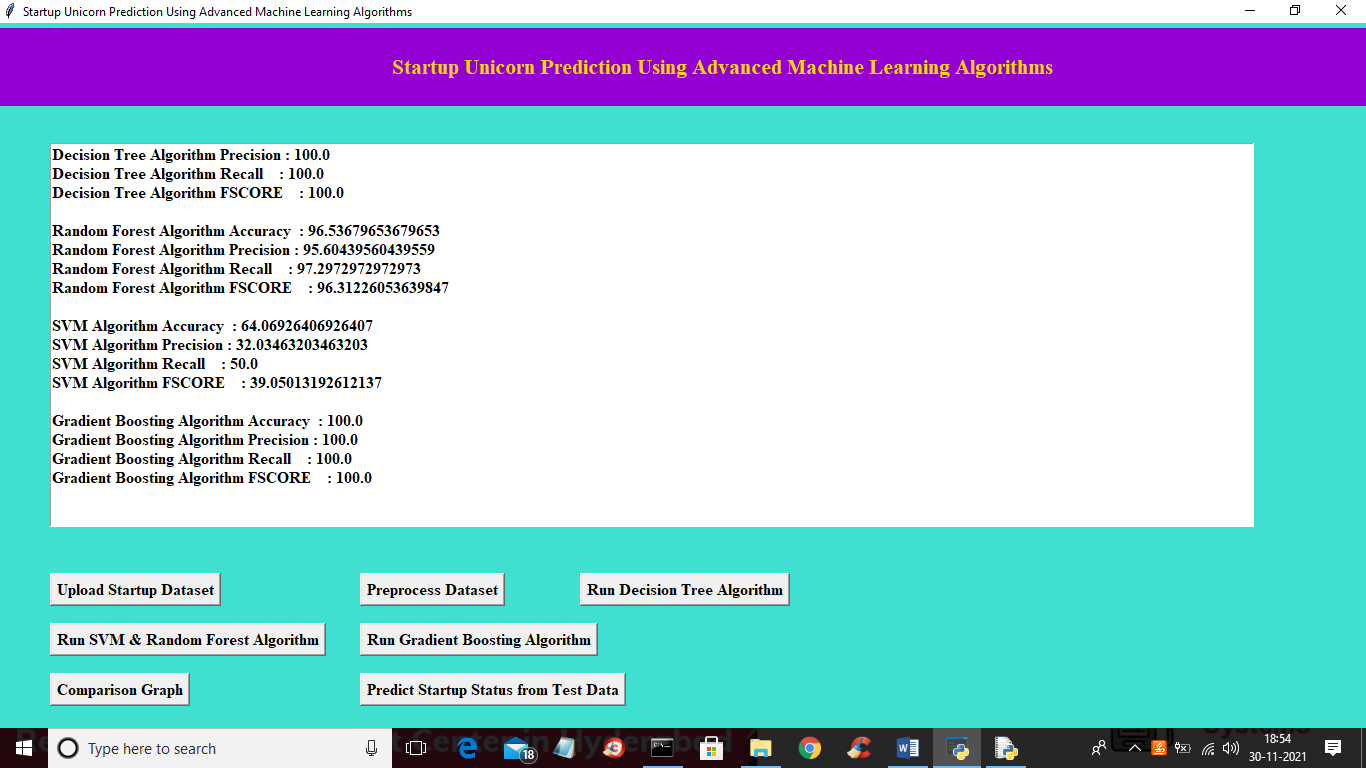
In above screen we can see dataset contains 1154 total records and each record contains 32 features/columns and then application split dataset into 80% (923 records) for training and remaining 231 records will be used to test trained algorithms prediction performance in terms of accuracy. Now train and test data is ready and now click on ‘Run Decision Tree Algorithm’ button to train Decision Tree with above dataset and get below output



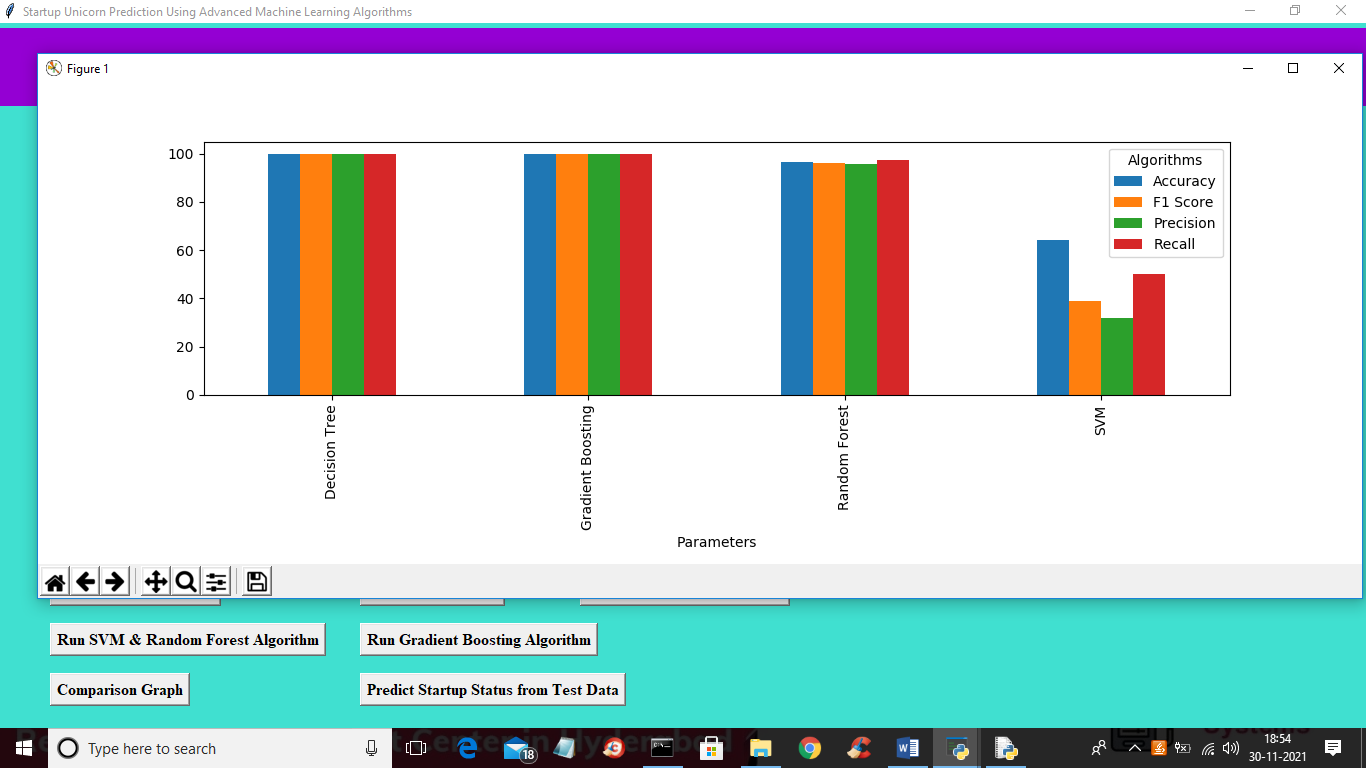
In above screen with decision tree we got 100% accuracy and now click on ‘Run SVM & Random Forest Algorithms’ button to get below output



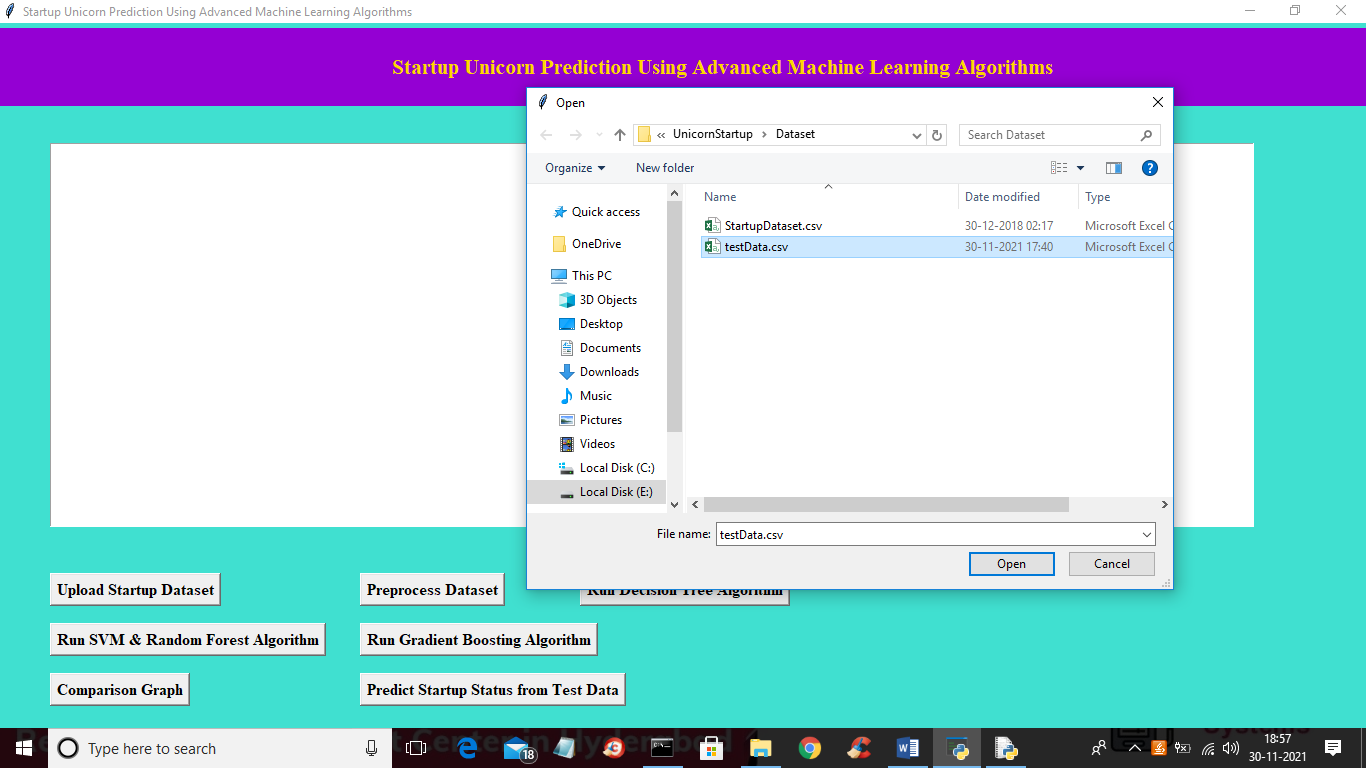
In above screen with random forest we got 96% accuracy and with SVM we got 64% accuracy and now click on ‘Run Gradient Boosting Algorithm’ button to train gradient boosting with above dataset and to get below output



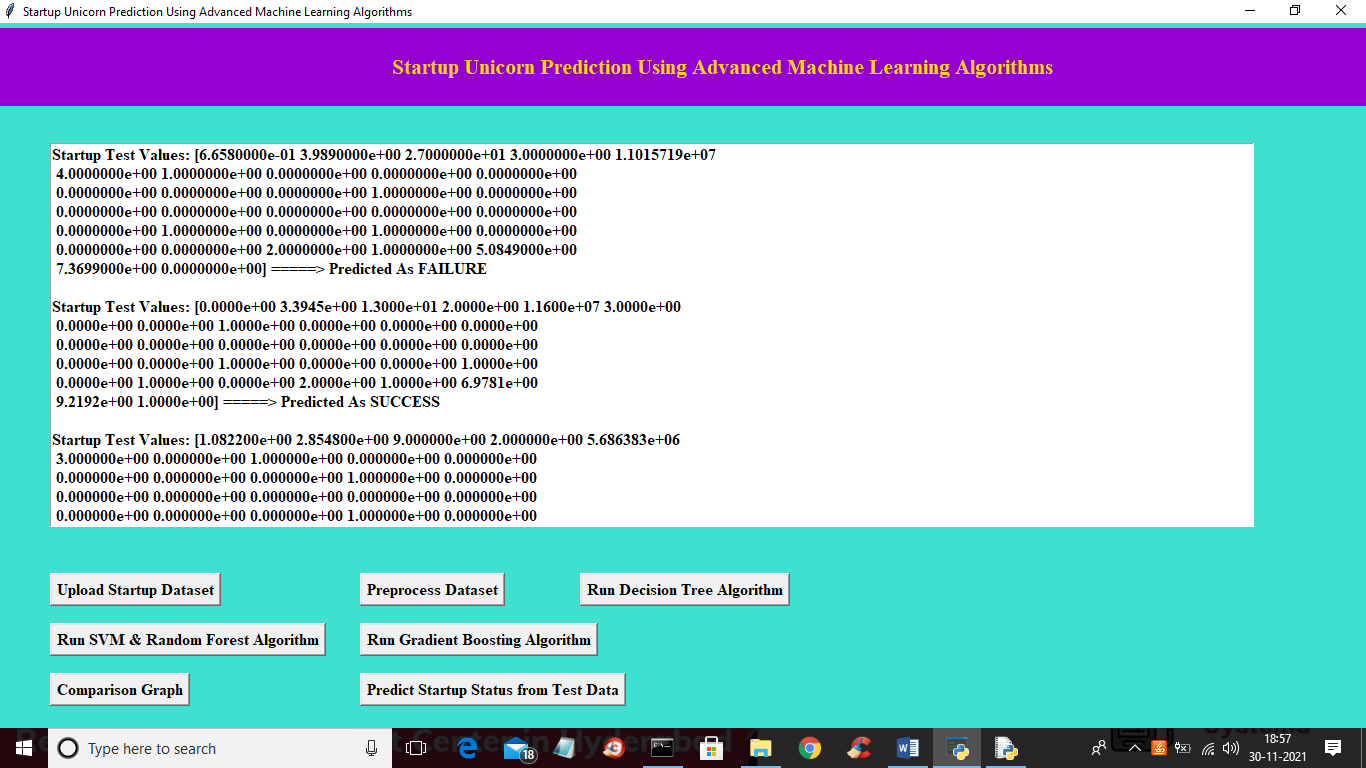
In above screen with Gradient Boosting also we got 100% accuracy and now click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithms names and y-axis represents accuracy, precision, recall and FSCORE and in above graph different colour bar graph represents different metrics for each algorithm. In above graph we can see Decision Tree and Gradient Boosting giving high performance compare to other algorithms. Now close above graph and then click on ‘Predict Startup Status from Test Data’ button to upload test data and to get below output



In above screen selecting and uploading ‘testData.csv’ file and then click on “Open’ button to load test data and to get below prediction result



In above screen inside square bracket we can see Startup test values and based on test values ML algorithms perform prediction. We can see prediction result for each test record after ====🡺 arrow symbol as FAILURE or SUCCESS

**8. CONCLUSION**

Predicting the success of early stage startups is a challenging task and the costs of misclassification is high as it might lead to disastrous funding decisions are missing valuable chances for return. To make predictions in such contexts, we propose to combine the complementary capabilities of machine and human intelligence. While machines are particularly beneficial in consistently processing large amount of “hard” signals that indicate the success of a new venture humans are superior in interpreting “soft” signals such as the personality of an entrepreneur or the innovativeness of a new product. Moreover, humans can leverage their intuition to identify valuable startups that cannot be find by relying on previous data. To overcome the constraints of bounded rationality of individuals, we thus suggest leveraging collective intelligence. To reach our aim, we developed a preliminary Hybrid Intelligence method that we will initially evaluate as we proceed our research. In the next steps, we will then also test its applicability for other outcome variables in the context of startups (e.g. growth, survival rate etc.) and other contexts of extreme uncertainty (e.g. innovation in general). Moreover, we intend to assess the relevance of accuracy and transparency with potential users of this method and if they are more willing to take advice when human sources are included (e.g. Önkal et al. 2009). We expect our research to make several contributions to both academia and practice. First, we provide a taxonomy of potential predictors that can be generalized for modelling startup success predictions (e.g. Böhm et al. 2017). Second, this research adds to literature on predictive research in IS and data analytics (e.g. Chen et al. 2012) by introducing a new method for predicting uncertain outcomes under limited information and unknowable risk by combining collective and machine intelligence in a Hybrid Intelligence method. This approach allows to complement formal analysis of “hard” information and intuitive predictions based on “soft” information. Such hybrid method might be valuable for other settings of extreme uncertainty as well. Consequently, our research will offer prescriptive knowledge in this vein that might be generalizable for data science methods in general (Gregor and Jones 2007). Third, we contribute to previous work on collective intelligence (e.g. Malone et al. 2009; Wooley et al. 2010) by proposing novel applications of machines and crowd. We argue that our proposed approach can augment the capabilities of collective intelligence in general. While we use a parallel approach in this paper, further research might explore how machine intelligence might be leveraged as feedback for the crowd and thus point towards more collaborative interactive approaches (e.g. Calma et al. 2016). Finally, we provide a useful solution for a practical prediction problem that may support angel investors in making decisions and potentially reduce the frequency of bad investment decisions.

**9.REFERENCES**

Armstrong, J. S. 2001. Principles of forecasting: a handbook for researchers and practitioners: Springer Science & Business Media.

Atanasov, P. D., Rescober, P., Stone, E., Swift, S. A., Servan-Schreiber, E., Tetlock, P. E., Ungar, L., and Mellers, B. 2017. “Distilling the wisdom of crowds: Prediction markets versus prediction polls,” Management science (63:3), pp. 691–706.

Attenberg, J., Ipeirotis, P., and Provost, F. 2015. “Beat the Machine: Challenging Humans to Find a Predictive Model's “Unknown Unknowns”,” Journal of Data and Information Quality (JDIQ) (6:1), pp. 1-15.

Baer, J., and McKool, S. S. 2014. “The gold standard for assessing creativity,” International Journal of Quality Assurance in Engineering and Technology Education (IJQAETE) (3:1), pp. 81–93.

Baesens, B., Bapna, R., Marsden, J. R., Vanthienen, J., and Zhao, J. L. 2016. “Transformational issues of big data and analytics in networked business,” MIS Quarterly (40:4), pp. 807–818.

Baum, J. A. C., and Silverman, B. S. 2004. “Picking winners or building them? Alliance, intellectual, and human capital as selection criteria in venture financing and performance of biotechnology startups,” Journal of Business Venturing (19:3), pp. 411–436.

Baum, J. A. C., and Silverman, B. S. 2004. “Picking winners or building them? Alliance, intellectual, and human capital as selection criteria in venture financing and performance of biotechnology startups,” Journal of Business Venturing (19:3), pp. 411–436.

Blattberg, R. C., and Hoch, S. J. 1990. “Database models and managerial intuition: 50% model+ 50% manager,” Management Science (36:8), pp. 887–899.

Blohm, I., Riedl, C., Füller, J., and Leimeister, J. M. 2016. “Rate or trade? identifying winning ideas in open idea sourcing,” Information Systems Research (27:1), pp. 27–48.

Böhm, M., Weking, J., Fortunat, F., Müller, S., Welpe, I., and Krcmar, H. 2017. “The Business Model DNA: Towards an Approach for Predicting Business Model Success,” Proceedings of the WI 2017.

Bradley, A. P. 1997. “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” Pattern recognition (30:7), pp. 1145–1159.

Breiman, L. 2001. “Random forests,” Machine learning (45:1), pp. 5–32.

Brynjolfsson, E., Geva, T., and Reichman, S. 2016. “Crowd-squared: amplifying the predictive power of search trend data,” MIS Quarterly (40:4), pp. 941–961.

Burton-Jones, A., and Weber, R. 2014. “Building conceptual modeling on the foundation of ontology,” Computing Handbook, Third Edition: Information Systems and Information Technology. Chapman and Hall.

Busenitz, L. W., and Barney, J. B. 1997. “Differences between entrepreneurs and managers in large organizations: Biases and heuristics in strategic decision-making,” Journal of Business Venturing (12:1), pp. 9–30.

Carneiro, N., Figueira, G., and Costa, M. 2017. “A data mining based system for credit-card fraud detection in e-tail,” Decision Support Systems (96:3), pp. 91–101.

Chen, H., Chiang, R. H. L., and Storey, V. C. 2012. “Business intelligence and analytics: From big data to big impact,” MIS Quarterly (36:4), pp. 1165–1188.

Colton, S., and Wiggins, G. A. (eds.) 2012. Computational creativity: The final frontier? IOS Press.

Cowgill, B. 2017. Automating Judgement and Decision-making: Theory and Evidence from Résumé Screening. http://www.sole-jole.org/17446.pdf. Accessed 6 September 2017.

Cowgill, B., and Zitzewitz, E. 2015. “Corporate prediction markets: Evidence from google, ford, and firm x,” The Review of Economic Studies (82:4), pp. 1309–1341.

Creamer, G. G., Ren, Y., Sakamoto, Y., and Nickerson, J. V. 2016. “A Textual Analysis Algorithm for the Equity Market: The European Case,” The Journal of Investing (25:3), pp. 105–116.

Dutta, S., and Folta, T. B. 2016. “A comparison of the effect of angels and venture capitalists on innovation and value creation,” Journal of Business Venturing (31:1), pp. 39–54.

Ægisdóttir, S., White, M. J., Spengler, P. M., Maugherman, A. S., Anderson, L. A., Cook, R. S., Nichols, C. N., Lampropoulos, G. K., Walker, B. S., and Cohen, G. 2006. “The meta-analysis of clinical judgment project: Fifty-six years of accumulated research on clinical versus statistical prediction,” The Counseling Psychologist (34:3), pp. 341–382.

Einhorn, H. J. 1972. “Expert measurement and mechanical combination,” Organizational Behavior and Human Performance (7:1), pp. 86–106.