DATA PROGRAMMING PROJECT – 2 REPORT

1. Motivation and Problem: First, you need to describe why this problem is important and why people should care enough to pay you to solve it. Then provide your problem statement in very clear terms, as there should not be any ambiguity.

Several factors determine if a startup will survive in the long run. While about 305 million startups are created all over the world each year, 90% of all startups fail [1], and only 2 in 5 startups will be profitable. These statistics pose investors with the question of whether to invest or not in a particular startup. While the answer to this question may not be 100% empirical and rather involve unquantifiable reasons like market situation, law, soft skills, etc., the analysis of some factors, viz. age of the startup, its industry, its milestones, etc., can help perform an initial screening of to-be failure startups, that is the startups which should not be invested in.

Traditionally, domain experts analyse all such quantifiable and unquantifiable factors, in a highly time and resource intensive process, to decide whether to invest in a startup or not. The experts often review several funding applicants before selecting a small number of winners. Our model addresses this problem by identifying trends in the startup world history that suggest eventual failure, then classifying a given startup as showing such trends or not. Based on results from the model, investors can allocate more resources and efforts towards scrutinizing the startups predicted by the model to fail, and less towards those predicted to succeed, thereby optimising the scrutiny process and allowing for more startups to be efficiently analyzed.

Therefore, the problem statement boils down to -
"Predicting whether a startup will fail or not."

Because this is a one-time investment, it is sensible to pay to create a model that would allow an investment firm to operate with a limited number of domain experts and explore more investing opportunities in a short span of time. In the long run, this approach could be modified to include real time factors from the market, and would prove more profitable than hiring a larger number of domain experts on a fixed payroll.

• **Solution:** Second, you need to provide a high-level description as to what is your solution to solve this problem, and why your approach is the right solution.

Since selecting the winners from among the hundreds of companies vying for investment is a difficult undertaking, it would be almost impossible to engage enough experts to evaluate every startup. Therefore, using historical startup data and different indicators that help us estimate the chances of a startup failing, a machine learning model can assist them in shortlisting investment opportunities that require a higher level of risk analysis. Details of the features available and used can be found in reports for Project 1 and Project 0.

We intend to solve this problem through the Supervised Learning approach by training a model based on the history of startups which were either acquired, achieved IPO, or still functional (classified as *successful startups*), or closed (classified as *failed startups*). The trained model will subsequently be applied to forecast the *success* or *failure* of startups that are already up and running.

The classification model will assist an investor in shortlisting startups seeking funding, with a quick and important insight of whether the startup will fail or not, based on its current trend. Equipped with this information, investors can look at other factors, then use their domain expertise to determine whether to invest or not.

• **How it works**: Third, you need to explain how your approach works. Start with providing an architecture of your approach to give a big picture. Then, you need to provide ample details of the components of your approach that will explain their inner workings. Also, you need to provide justifications for why you chose those specific techniques or algorithms. Explain what steps you took to improve the model, e.g., tuning models, manipulating data.

A 5-step standard operating procedure for analytics was followed, as mentioned in Project 0, to provide a concrete data-driven solution. However, revised details of the process as uncovered by doing are given below-

(1) **Prepare** : Collect the data required to model the problem and to test the model from an open source- Kaggle [2].
(2) **Process** : Further details in Report for Project 1, 2).
   a. Cleaned the data (removed errors, outliers, empty values, etc) and made it reliable for an ML model.
   b. Performed encoding of categorical features.
   c. Derived new features from existing features.
   d. Performed oversampling via Synthetic Minority Over-sampling Technique to bring down the imbalance in target variable using smote.
(3) **Analyse** :
   1. Performed exploratory descriptive analysis of the data, followed by feature selection.
   2. Since the problem is of binary classification, an array of both linear and non-linear classifier machine learning models were used to train and test the model.
      a. Decision Tree
      b. Random Forest
      c. K-Nearest Neighbors
      d. Naive Bayes
      e. Logistic Regression
      f. SVM
      g. Perceptron
   3. Refine model using Ensemble techniques -

    a. Adaboost
    b. Gradient Boost
    c. Voting Classifier
4. Hyper parameter tuning via grid search. Then refine, re-train and re-test the model until satisfactory results.

(4) **Share** : Convert the analysis into insightful and useful visualisations for the client and clearly state the suggested actions.
(5) **Act** : Based on the result of analysis, either implement the suggested actions, or approach the problem from a different perspective.

• **Evaluation, Outcomes and Discussion:** Fourth, you need to provide results of the models based on the evaluation metrics you chose. Explain which models are performing better and why this might be the case. Provide a discussion as to why the model you pick will perform well, and state if there may be situations it may not perform well. What are key takeaways from your analysis and models?

For this problem,
True Positive: Predicted closure of a startup and the startup actually getting closed.
False Positive: Predicted that the startup closes, but the actual outcome is that the startup is a success.
True Negative:Predicted startup is going to survive when the startup actually is successful.
False Negative: Predicted that the startup is going to succeed but the actual outcome being the startup getting closed.

Since, the cost of false negatives is high as the investors would lose money if they invest in companies that are likely to get closed. We used recall as an accuracy metric for comparing models.

Below are the classification reports of all the fitted classification models-

Model1 : Random Forest. (1- startup getting closed)

	precision	recall	f1-score	support
0	0.98	1.00	0.99	6603
1	0.33	0.04	0.06	171

Model 2 : SVM. (1- startup getting closed)

	precision	recall	f1-score	support
0	0.97	1.00	0.99	6603
1	0.00	0.00	0.00	171

Model 3 : AdaBoost Classifier (1- startup getting closed)

	precision	recall	f1-score	support
0	0.98	0.99	0.98	6603
1	0.21	0.12	0.15	171

Model 4 : Gradient Boost Classifier (1- startup getting closed)

	precision	recall	f1-score	support
0	0.98	0.98	0.98	6603
1	0.14	0.12	0.13	171

Model 5 : KNN (1- startup getting closed)

	precision	recall	f1-score	support
0	0.98	0.90	0.94	6603
1	0.09	0.37	0.14	171

Model 6 : Logistic Regression (1- startup getting closed)

	precision	recall	f1-score	support
0	0.97	1.00	0.99	6603
1	0.00	0.00	0.00	171

Model 7 : Perceptron Classifier (1- startup getting closed)

	precision	recall	f1-score	support
0	0.98	0.95	0.96	6603
1	0.07	0.16	0.10	171

Model 8 : Naive Bayes Classifier (1- startup getting closed)

	precision	recall	f1-score	support
0	0.99	0.65	0.78	6603
1	0.06	0.80	0.10	171

Of all the models fitted the model with best recall is Naive Bayes Classifier.

Why will the model you picked perform well?

Since our model is built to minimise the cost of false negatives, the selected Naive Bayes model with 0.8 recall for class 1 *(startup will fail)* will perform well. This is helpful because the consequence of identifying a startup as *successful* is that lesser resources will be allocated by the investor in properly determining whether to invest or not, which might lead to overlooking some indicators of failure that are not considered by the model. Therefore, a high recall model for class 1 will perform well.

Are there situations when the model may not perform well?

Given the inherent nature of the startup world, that only 10% of all startups will succeed, the model is likely to detect most startups which are likely to fail, and maintain efficiency. However, when offered a situation where the model is made to predict the fate of several startups which in reality, are likely to succeed, many of them will be predicted as likely to fail. This might result in loss of efficiency as a resource intensive scrutiny process might be used to determine whether a true successful startup is worthy of investing or not.

Citations:

[1] Team, E. M. B. (2022, October 25). *106 must-know startup statistics for 2022*. Embroker. Retrieved November 29, 2022, from https://www.embroker.com/blog/startup-statistics/#:~:text=03-,Startup,About%2090%25%20of%20startups%20fail.&text=10%25%20of%20startups%20fail%20within%20the%20first%20year.&text=Across%20all%20industries%2C%20startup%20failure,be%20close%20to%20the%20same.&text=Failure%20is%20most%20common%20for,70%25%20falling%20into%20this%20category.

[2] Polcari, F. (2021). Startup Success Prediction, Version 17. Retrieved September 20, 2022 from https://www.kaggle.com/code/fpolcari/startup-success-prediction.