Startup Success Prediction

DSCI 5240: Data Mining

Prof. Mahdi Fathi

Department of Information Technology and Decision Sciences

Group 12

SRITHA DARBHA

SRAVANTHI PERI

KOMAL SINGH BHAMRA

ARBAN MOHAMMED

UDAYA SREE DOKKA

Project Motivation/background

- A start-up is a company begun by an entrepreneur to seek, develop, and validate a scalable economic model. While entrepreneurship refers to all new businesses, including self-employment and businesses that never intend to become registered, start-up's refer to new businesses that intend to grow large beyond the solo founder. Start-ups face high uncertainty and have high rates of failure, but a minority of them do go on to be successful and influential. Start-up's play a major role in economic growth. They bring new ideas, spur innovation, create employment, thereby moving the economy. There has been an exponential growth in start-up's over the past few years. [Source of information: Wikipedia].
- Predicting the success of a start-up allows investors to find companies that have the potential for rapid growth, thereby allowing them to be one step ahead of the competition.

Dataset Description

- The start-up dataset is a second-hand dataset and has been taken from the following website: https://www.kaggle.com/manishkc06/startup-success-prediction
- The data contains industry trends, investment insights and individual company information. There are 47 columns/features.
- The target variable ‘Status’ is the field that we are predicting which takes two values.

1. Acquired : 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). It is indicated as 1 in the dataset.

2. Closed : The failure of a company is defined as the event where it had to be Closed or Shut down. It is indicated as 0 in the dataset.

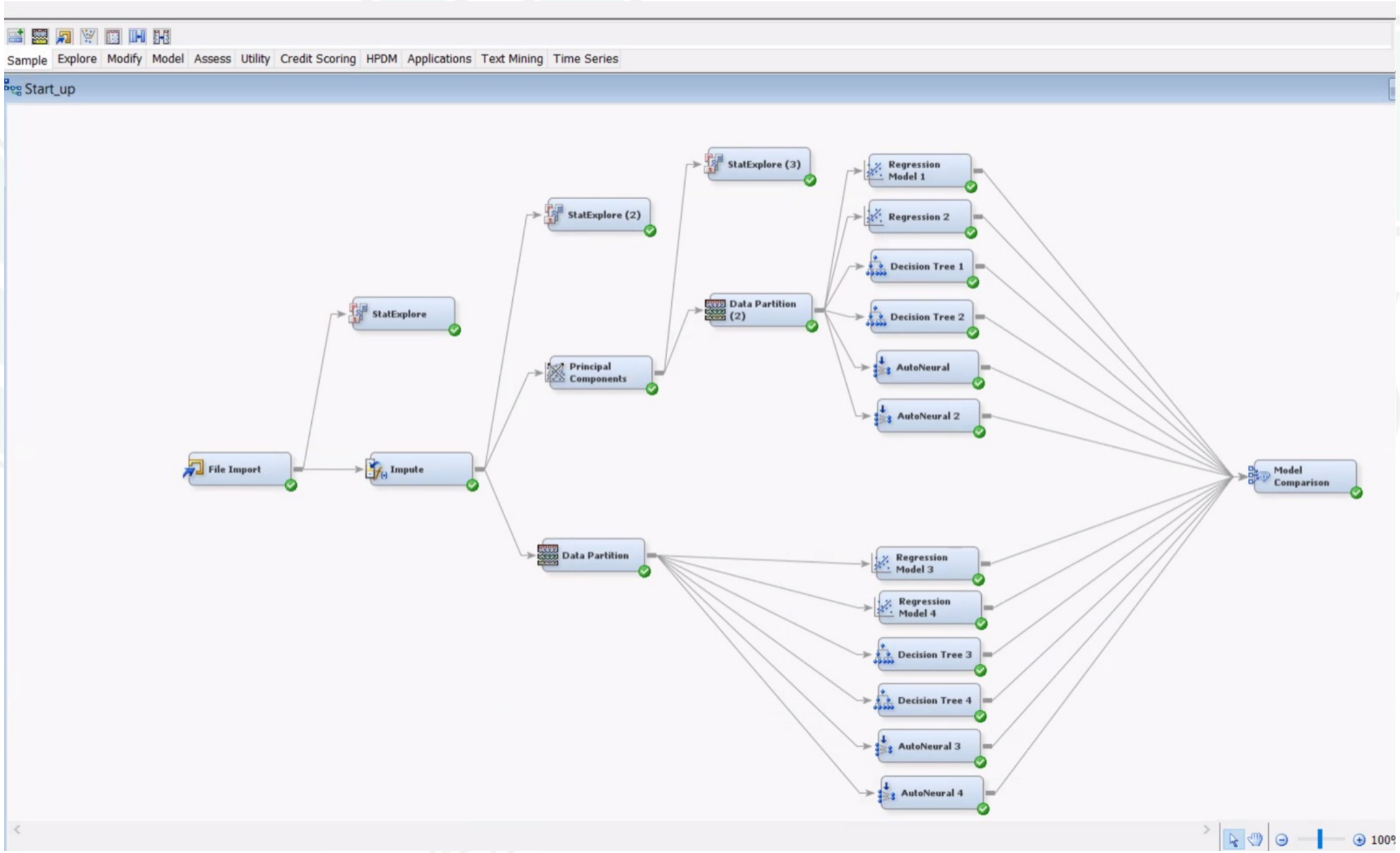
- The Class Distribution for data is Acquired : 597 and Closed : 326

Analyzing the Data

- We have used SAS Enterprise Miner 15.1 for our project.
- We have used 4 models

1. Principal Component Analysis
2. Logistic Regression
3. Decision Tree
4. Auto Neural Networks

- We used and compared these four models to determine the most efficient model.
- The advantages and disadvantages of various models are assessed, and model comparison node results are considered in determining an efficient model.


Sample
Explore
Modify
Model
Assess
Utility
Credit Scoring
HPDM
Applications
Text Mining
Time Series
Start_up
File Import
StatExplore
Impute
StatExplore (2)
StatExplore (3)
Principal Components
Data Partition (2)
Data Partition
Regression Model 1
Regression 2
Decision Tree 1
Decision Tree 2
AutoNeural
AutoNeural 2
Regression Model 3
Regression Model 4
Decision Tree 3
Decision Tree 4
AutoNeural 3
AutoNeural 4
Model Comparison
100%

Principal Component Analysis

- We ran the Principal Component Analysis to reduce the dimensions in the data and to investigate how this effects different models.
- After running the PCA node, we observed the that variable count is reduced to 20 from 47.
- The results have been explored more in the Project document.

Logistic Regression

- After reducing the variable count, we ran Logistic Regression with all variables again to check for the R-square and Adjusted R-square values.
- We ran a total of four regression models. We ran with a combination of selecting all variables and selecting only positively correlated variables with and without running PCA. We did this till we got all significant values i.e., the p value in F- test is less than 0.05.
- The results have been explored more in the Project document.

Decision Tree

- We now used Decision Tree node to analyse the data.
- We ran a total of four Decision Tree models. We ran with a combination of selection of all variables and selecting only positively correlated variables with and without running PCA.
- The results have been explored more in the Project document.

Auto Neural Mode

- We now used Auto Neural node to analyse the data.
- We ran a total of four Auto Neural models. We ran with a combination of selection of all variables and selecting only positively correlated variables with and without running PCA.
- The results have been explored more in the Project document.

Conclusion

- We have attached a Model Comparison node to compare and determine the best model among all models.
- Based on the Misclassification rate, we found Regression 2 Model to be the best model as it has the misclassification rate of 0.21505, which is the least of all.
- Based on Average Square Error, we found Auto Neural 3 Model to be the best model with the least average square error of 0.16354.

Model Comparison Result

Fit Statistics

Model Selection based on Valid: Misclassification Rate (_VMISC_)

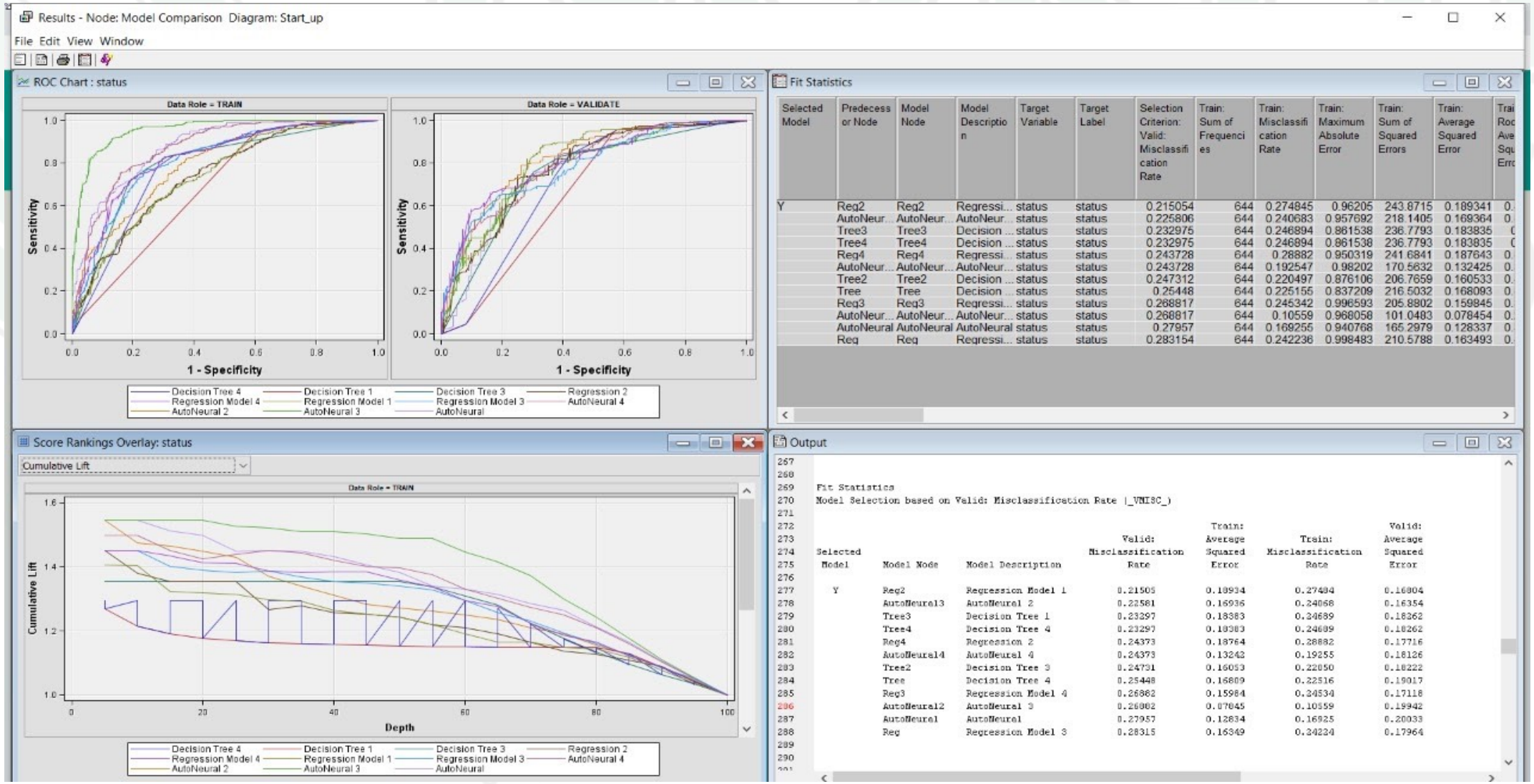
Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Reg2	Regression Model 1	0.21505	0.18934	0.27484	0.16804
	AutoNeural3	AutoNeural 2	0.22581	0.16936	0.24068	0.16354
	Tree3	Decision Tree 1	0.23297	0.18383	0.24689	0.18262
	Tree4	Decision Tree 4	0.23297	0.18383	0.24689	0.18262
	Reg4	Regression 2	0.24373	0.18764	0.28882	0.17716
	AutoNeural4	AutoNeural 4	0.24373	0.13242	0.19255	0.18126
	Tree2	Decision Tree 3	0.24731	0.16053	0.22050	0.18222
	Tree	Decision Tree 4	0.25448	0.16809	0.22516	0.19017
	Reg3	Regression Model 4	0.26882	0.15984	0.24534	0.17118
	AutoNeural2	AutoNeural 3	0.26882	0.07845	0.10559	0.19942
	AutoNeural	AutoNeural	0.27957	0.12834	0.16925	0.20033
	Reg	Regression Model 3	0.28315	0.16349	0.24224	0.17964

Fit Statistics

Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Reg2	Regression Model 1	0.21505	0.18934	0.27484	0.16804
	AutoNeural3	AutoNeural 2	0.22581	0.16936	0.24068	0.16354
	Tree3	Decision Tree 1	0.23297	0.18383	0.24689	0.18262
	Tree4	Decision Tree 4	0.23297	0.18383	0.24689	0.18262
	Reg4	Regression 2	0.24373	0.18764	0.28882	0.17716
	AutoNeural4	AutoNeural 4	0.24373	0.13242	0.19255	0.18126
	Tree2	Decision Tree 3	0.24731	0.16053	0.22050	0.18222
	Tree	Decision Tree 4	0.25448	0.16809	0.22516	0.19017
	Reg3	Regression Model 4	0.26882	0.15984	0.24534	0.17118
	AutoNeural2	AutoNeural 3	0.26882	0.07845	0.10559	0.19942
	AutoNeural	AutoNeural	0.27957	0.12834	0.16925	0.20033
	Reg	Regression Model 3	0.28315	0.16349	0.24224	0.17964

Model Comparison Result



Alternative Work

- We also performed the same project using Python.
- We used four algorithms: Logistic Regression, Decision Tree, Random Forest, Naïve Bayes. Out of the four algorithms, we found Random Forest as the better performing model. It has 78% accuracy.
- From the output, we have Acquired (1) start-ups has more f-1 score (0.83) than the f-1 score (0.67) of Closed (0) start-ups but precision of Closed start-ups is more than the Acquired start-ups.
- The results have been explored more in the Project document.

References

1. https://www.kaggle.com/manishkc06/startup-success-prediction.
2. https://documentation.sas.com/doc/en/emref/15.1/titlepage.htm.
3. https://analyticsconsultores.com.mx/wp-content/uploads/2019/03/Decision-Trees-for-Analytics-Using-SAS-Enterprise-Miner-B.-de-Ville-P.-Neville-SAS-2013.pdf.
4. https://medium.com/hackernoon/logistic-regression-using-sas-enterprise-guide-3ffb7774f765.
5. https://bykelly93.wordpress.com/2016/04/28/sas-enterprise-miner-logistic-regression/.
6. https://www.mwsug.org/proceedings/2009/stats/MWSUG-2009-D02.pdf.
7. https://communities.sas.com/t5/SAS-Data-Mining-and-Machine/Principle-component-analysis-in-Enterprise-Miner/td-p/480335.
8. https://www.lexjansen.com/wuss/2007/AnalyticsStatistics/ANL_Matignon_DataMining.pdf.
9. https://www.datalab-crm.de/wp-content/uploads/2017/07/sas-enterprise-miner-101369.pdf.
10. https://realpython.com/logistic-regression-python/
11. https://www.datacamp.com/community/tutorials/decision-tree-classification-python.
12. https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
13. https://www.datacamp.com/community/tutorials/random-forests-classifier-python
14. https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/
15. https://www.analyticsvidhya.com/blog/2021/01/a-guide-to-the-naive-bayes-algorithm/

THANK YOU