CSC 522 - Comparison of Encoding and Classification Methodologies for Prediction of Grant Applications

Priyaranjan Behera
Department of Computer Science
North Carolina State University
pbehera@ncsu.edu

Sai Sri Harsha Kunapareddy
Department of Computer Science
North Carolina State University
skunapa@ncsu.edu

Abstract

This project focuses on finding an accurate method to predict the fate of a university grant application. The problem was hosted as a Kaggle competition by University of Melbourne to improve their efficiency on grant applications. The dataset contains a mixture of continuous and categorical data and thus, to handle the categorical data attributes we employed methodologies like binary and probabilistic encoding. We created models based on classifiers such as Decision Trees, Random Forest, AdaBoost Trees, SVMs & Neural Networks and compared their effectiveness.We used several evaluation parameters like accuracy, area under the curve in ROC plots, TPR and TNR to compare the models and determine the most effective model. Furthermore, we also worked on finding the most important attributes which affects the success of a grant application.

1 Background

Around the world, the pool of funds available for research grants is steadily shrinking (in a relative sense). In Australia, success rates have fallen to 20-25 per cent, meaning that most academics are spending valuable time making applications that end up being rejected. This problem was hosted as a competition by University of Melbourne to address their inefficiencies with the grant applications. There is also a hope of discovering the most important criteria that are required to succeed in a grant application.

1.1 Problem

The university has provided a dataset containing 249 features, including variables that represent:

- Type and size of the grant: It includes the basic information of the grant application like application month, year, contract value, sponsors, etc.
- General area of study of the grant application: It includes the information on the study that is intended to be done with the grant such as research fields, courses and disciplines class and socio economic objective class.
- De-identified information of the investigators: It includes personal data of the investigators like number of publications in different levels of journals, past history in grant applications, experience, educational qualifications, etc.

The dataset contains multi-variate data with a few of them being continuous variables and a few categorical. There is a variable number of investigators in an application and thus, we need to aggregate the person data to create standard data rows which can be compared during classification. We implemented several of the classification techniques covered in the CSC522 course to arrive upon an efficient model.

1.2 Literature Survey

As the dataset contains a variable number of person attributes depending on the number of investigators in a grant application, it would lead to multiple missing data fields in applications where the investigator count is less. According to Gerhard Svolba [2], we need to create a one-row-per-subject data mart for most of the analytical methods that we need to proceed with. Accordingly, we need to aggregate the person data using mean, median, standard deviation, maximum, minimum, total. etc to create the input rows.

We looked at approaches taken to solve similar problems which contains both continuous and categorical data [6] and found that decision trees and Naive Bayes are popular choices. On the contrary, methods like neural networks, SVM, etc. cannot work well with categorical data. We can implement decision trees as well as bagged and boosted trees for more efficiency.

Daniele et al. [4] focused on the transformation of categorical data to continuous/binary data so that we can use the analytical methods which cannot process categorical data. While traditionally binary encoding is used for conversion of categorical data to binary, for categories with high cardinality a probabilistic approach is suggested.

2 Methods

The analysis of the data required extensive preprocessing steps because of its multi-variate nature and variable number of attributes. We implemented different types of classification techniques to compare the models and find the determining attributes for the classification.

2.1 Preprocessing

To handle the categorical values in the data, we indexed the attributes with numerical values after finding the unique values for each of the attributes. As per the approach suggested by Gerhard Svolba [2], we aggregated the variable number of investigators specific data for each of the application which is not null and found the minimum, maximum for all the attributes while for continuous attributes we also calculated mean, median, sum of the values to create new features. We then created plots of the attributes with respect to the output to visually inspect any correlation of the attributes with the results.

Since classification techniques like SVMs, Neural Networks, cannot handle categorical data well, we implemented binary encoding to convert the categorical data into binary attributes which can be used in the analytical techniques. Further, to handle high cardinality, we implemented a probabilistic feature creation as specified at [4].

2.2 Classification

We implemented most of the classification techniques covered in CSC522 and compared their efficiencies.

2.2.1 Decision Tree and Random Forest

According to our survey of approaches employed for similar problems, decision tree turns out to be a popular choice. This attributes to the fact the it can handle categorical values without any further processing. We also implemented bagged trees using the dataset to get a more efficient classification. We also used the error rate obtained by changing values of attributes in a random forest to determine the most important factors which decide the success of a grant application.

2.2.2 Neural Networks

Since the neural networks can handle a high number of attributes, this will serve as an ideal classifier particularly when we use the binary encoding with a high cardinality of the categorical attributes. However, we cannot estimate the determining factors of an application though this method.

2.2.3 SVMs

SVM based classifers are known to work well with encoded data and thus, we executed the SVM based classifiers on the binary and probabilistic encoded data. We used the linear, quadratic and cubic kernels to find an effective model.

3 Plan

According to the problem statement of the competition, we aimed to find a model which has a greater accuracy while minimizing the false positives in the output.

3.1 Hypothesis

We worked on implementing models of the classifications techniques we have been familiarized with to find a model with lesser number of false positives and higher accuracy. Thus, we focused on the values of True Positive Rate, True Negative Rate, Precision and Accuracy of the models. Most importantly, we used the ROC (Receiver Operating Curve) to determine the effectiveness of the model by comparing the area under the curve for the models. This will be more helpful compared to other criteria as:

- It is insensitive to an unbalanced dataset where there is a disparity in the number of outputs of the data objects.
- We consider all the cut-offs in a model to find its effectiveness.

We worked on experiments to test the technique of conversion of categorical attributes to probabilistic values for high cardinality attributes[4]. We considered the same evaluation parameters mentioned above to compare with models which use binary encoding.

We also worked on finding the most determining factors which decides the fate of an application. For this, we used the characteristic of the random forest libraries which find the effective factors in an input by measuring the error rate when an attribute value is changed.

3.2 Experimental Design

1. *Data Indexing:* Since there are a large number of categorical attributes in the data, indexing was done to standardize them into integral values.
2. *Dimensionality Reduction and Data Creation:* Since the data has variable number of person data in each of the grant application, there was a need to aggregate the data to create similar rows for comparing applications. We used different aggregation functions like minimum, maximum, mean, median, sum of the attribute values for each of the person within a data row[2] to achieve this.
3. *Data Visualization:* Effectiveness of the attributes to determine the result of an application was studied by creating scatter plots and histograms for each of the attributes in the input matrix.
4. *Handling of Categorical Attributes:* While techniques involving decision trees and naive bayes work well with categorical values, other methods need binary or continuous inputs. Thus, we will be adopting two methods to use the data in other classifiers:
    - Binary encoding: In this technique, an indexed categorical variable with N cardinality is converted to its binarized index value.
    - Probability Substitution: In this technique, the probability of getting the output as 1 is calculated for each of the values of the categorical variable. These values are used instead of the original variable values for the classifications. [4]
5. *Classifications:* We used the different inputs obtained in the above steps to train and test the classifiers.
    - Decision Tree
    - Random Forest

- Boosted Trees
- Neural Networks
- SVMs

Primarily, libraries in Matlab will be used for this purpose where tools like 'classificationLearner', 'nntool' will be used in addition to other basic classifier classes.

6. *Factor Determination:* The random forest library explicitly determines the most important factor which decides the output. We will be using this to find out the factors which can improve the credentials of an application.

7. *Evaluation and Results:* Confusion matrices will be drawn for each of the classifications done in the experiments to find the accuracy, recall, precision and false positive rate. Additionally, ROC will be generated from the GUI interfaces in 'classificationLearner' and 'nntool'.

4 Experiments and Analysis

4.1 Pre-Processing

After the pre-processing steps which included indexing and generation of new features, we resorted to visualization of the data to find any correlation of attributes with the results (As shown in Fig.1).

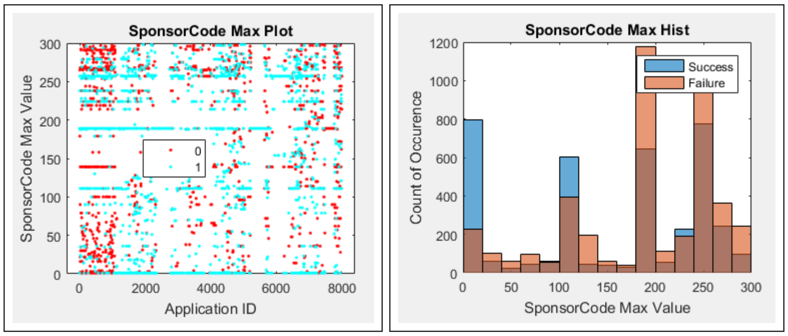

Figure 1: Scatter Plot and Histogram for Sponsor Codes.

We take an example of one of the attributes, sponsor code, for which we plot the data and see a discerning pattern. We can see that for specific sponsor codes, the chances of getting a grant is more. But we could not find any conclusive pattern in most of the plots and thus, moved on with the classification.

4.2 Establishing the Baseline

We implemented a simple one node decision tree to obtain one of the attributes, the mean of number of unsuccessful grants, at the root node. Thus, the classification done with the help of a decision stub with this variable will act as baseline for this problem. Based on the tree obtained, any data object with the variable value greater than 0.813333 was classified as 0 and others as 1. We obtained an accuracy of 72.1% from this and we aimed to design models with greater accuracy.

4.3 Comparison of Classification Methologies

Since our data did not have too many attributes, dimensionality reduction was not required. For the decision tree and SVM classifiers we opted for a 5-fold cross validation. After the classification models were implemented, we analyzed the confusion matrix to deduce the evaluation parameters and compare the models. One of the confusion matrices along with TPR/FNR rate which was obtained from the random forest model for categorical data is listed at Fig.2.

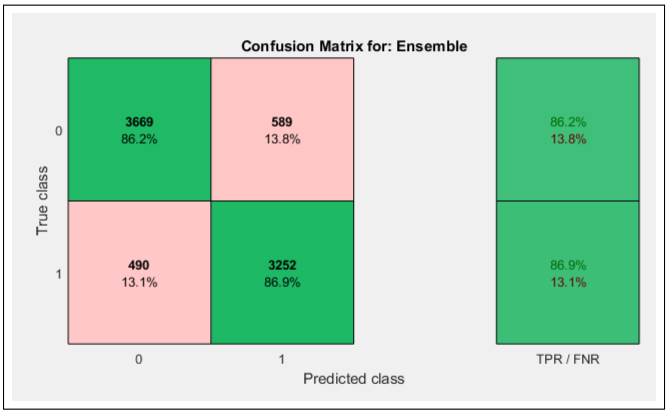

Figure 2: Confusion Matrix and TPR/FNR of Random Forest on Categorical Data.

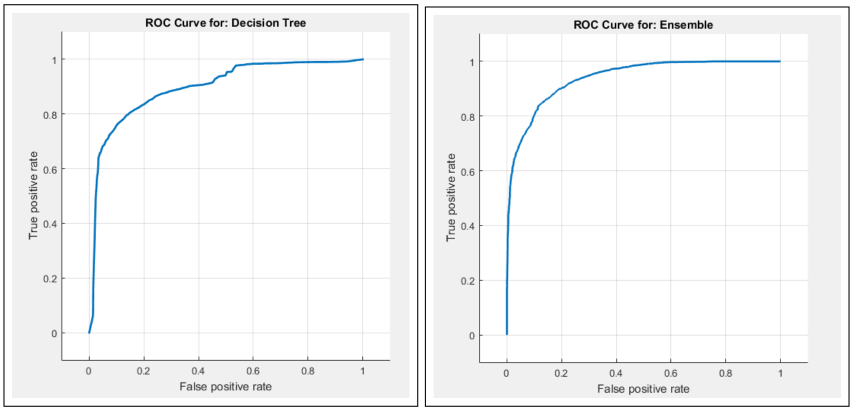

Figure 3: ROC Plot for Decision Tree and Random Forest.

We also compared the models according to the area under the curve of the ROC curve. This will help us to determine the effectiveness of the model in spite of any imbalance in the number of data objects with a specific output.

By comparing the models which used decision tree and random forest for categorical data based on the ROC curve(Fig.3), we find that the area under the curve for random forest is higher and thus, random forest is a better model in this scenario where we take the data without any encoding.

4.4 Comparison of Encoding Methodologies

Similarly, we compared the encoding methodologies by using the same classification methodology for each of the encodings and evaluating the parameters. Fig. 4 provides the comparison between the encodings where we have taken quadratic SVM as the classifier. We see that binary encoding works a little better in this scenario.

4.5 Feature Importance Analysis

In addition, we found the important features which determine the outcome of the application. The feature importance is calculated by comparing the error obtained when a single attribute is changed by a specific value in the data object. The data obtained from this dataset is plotted in Fig.5. From the plot we can see that features Contract Value, Sponsor Code, Application Month, Application Year, Grant Category are the most important factors.

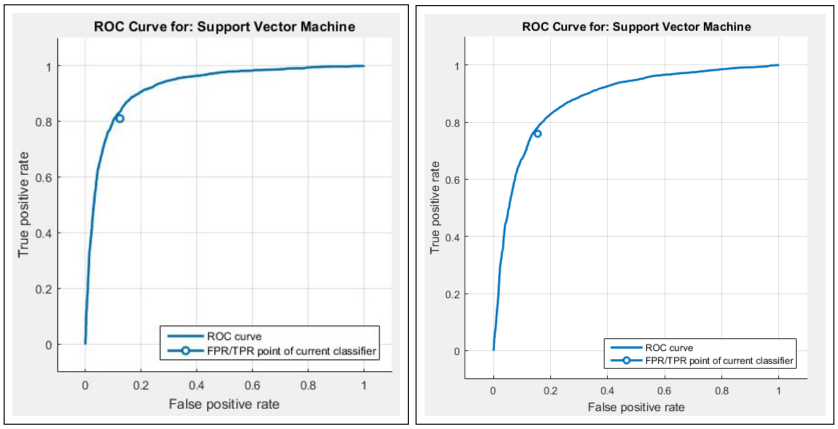

Figure 4: ROC Plot for Quadratic SVM Classifier with Binary(L) and Probabilistic(R) Encoding.

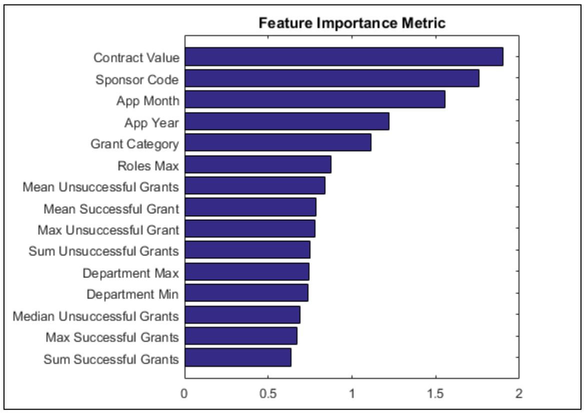

Figure 5: Feature Importance Matrix Generated By Random Forest.

4.6 Experiment Results

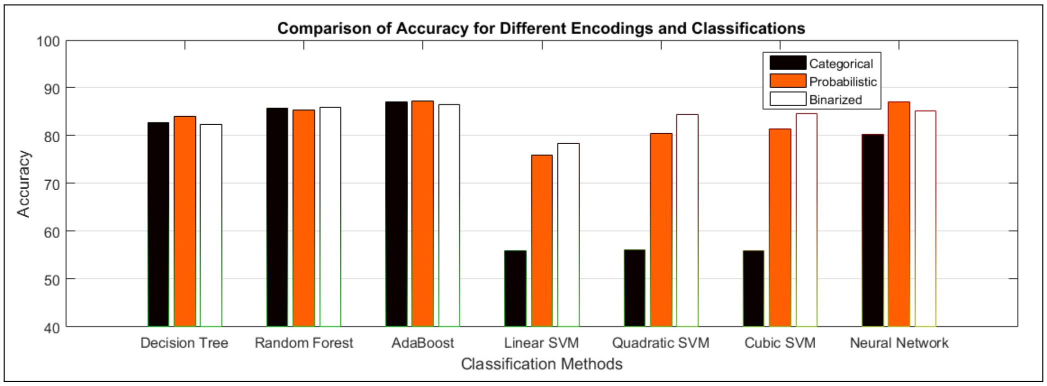

Figure 6: Accuracy of the classification models implemented.

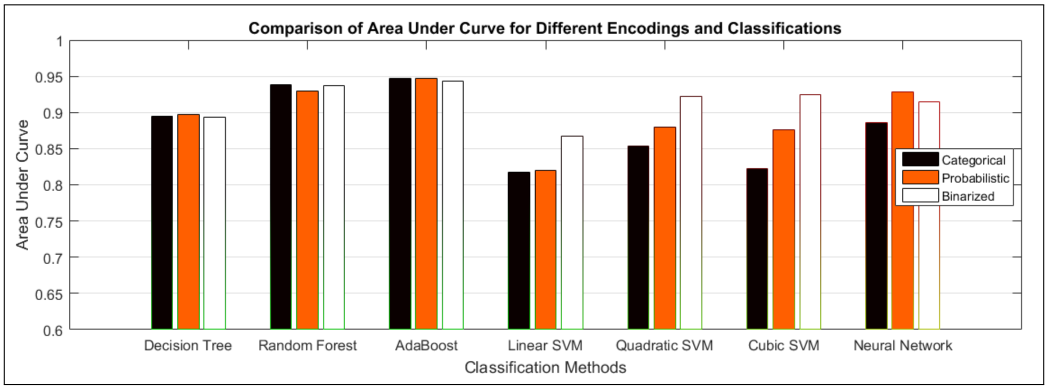

Figure 7: Area Under Curve of the ROC curves for all the implemented classification models.

Classification Method	Encoding Method	Accuracy (In %)	Area Under Curve	TPR (In %)	TNR(In %)
Decision Tree	Categorical Data	82.8	0.895	86.3	79.7
	Binary Encoded	82.3	0.893	84.6	80.1
	Probability Encoded	84.0	0.897	82.5	85.3
Random Forest	Categorical Data	85.8	0.938	86.1	85.6
	Binary Encoded	85.9	0.937	85.1	86.5
	Probability Encoded	85.4	0,930	82.6	87.9
AdaBoost Tree	Categorical Data	87.1	0.947	87.4	86.8
	Binary Encoded	86.6	0.943	85.2	87.8
	Probability Encoded	87.3	0.947	86.5	87.9
Linear SVM	Categorical Data	56.0	0.819	9.6	96.7
	Binary Encoded	78.4	0.867	73.7	82.6
	Probability Encoded	75.9	0.821	71.5	79.9
Quadratic SVM	Categorical Data	56.1	0.839	9.5	97.1
	Binary Encoded	84.5	0.922	81.0	87.7
	Probability Encoded	80.5	0.880	75.9	84.5
Cubic SVM	Categorical Data	55.9	0.806	9.1	97.1
	Binary Encoded	84.6	0.925	80.9	87.7
	Probability Encoded	80.8	0.875	77.0	84.2
Neural Network	Categorical Data	80.3	0.886	78.0	81.3
	Binary Encoded	85.1	0.915	82.5	87.5
	Probability Encoded	87.1	0.928	85.1	88.8

Table 1: Evaluation Parameters for all the implemented Classification Models.

The evaluation parameters for all the classification models that have been implemented are listed at Table. 1. We can also visually compare the effectiveness of the models based on the plots of accuracy and area under curve for each of the models in Fig. 6 and 7. The area under the curve in these cases mostly correlates with the accuracy. We can see that accuracies for SVMs are low when there is no encoding as all the data objects are classified as a single output (In this case 0). This is the reason TNR here is high while TPR is too low.

5 Conclusion

The experiments were done using a few of the classification techniques that were covered in the CSC522 courses. We trained the models using binary and probabilistic encodings of the categorical data as well as without using any encoding. Evaluation parameters like accuracy, area under of curve in ROC plot, TPR, TNR were used to compare the models.

From the results obtained we found that:

- Tree based classification methods work better with categorical data, while bagging and boosting of the trees gives us a more efficient model.
- SVMs fare poorly when they are trained with categorical data without any encodings. We saw improvements in the models with encoding, but they couldn't match the effectiveness of the tree-based classifiers for this problem.
- Neural network provides us a fairly good results without any encodings. But it provides a significant rise in the effectiveness when encodings were introduced.
- TPR/TNR determine the effectiveness of a model when there is a huge imbalance in the outputs of the classifier. In cases of SVMs when the classifier predicted all values as 0, we got a very high TNR value while TPR was low, signifying a poor model
- Based on our analysis, for this particular problem, AdaBoost along with probabilistic encoding gave use the model with the highest effectiveness which has an accuracy of 87.3% compared to the baseline of 72.1%.

6 Future Work

The dataset contains de-identified person data of the investigators in each of the applications. In case of grant applications, there might be individuals or combination of individuals who are more likely to receive a grant. Thus, we can create a Bayes Network based probabilistic classification model which can leverage the individuals' attributes to create a more effective model.

References

[1] Top 10 algorithms in data mining. Knowl. Inf. Syst. 14, 1 (December 2007), 1-37. DOI=http://dx.doi.org/10.1007/s10115-007-0114-2

[2] Efficient One-Row-per-Subject Data Mart Construction for Data Mining, Gerhard Svolba, PhD, SAS Austria

[3] Reliable Early Classification on Multivariate Time Series with Numerical and Categorical Attributes, Cao, Tru et al.

[4] Daniele Micci-Barreca. 2001. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. SIGKDD Explor. Newsl. 3, 1 (July 2001), 27-32. DOI=http://dx.doi.org/10.1145/507533.

[5] Getting Started with Kaggle Data Science Competitions. Loren Shure

[6] Predict Grant Applications - Kaggle Competition, https://www.kaggle.com/c/unimelb

Dataset:

`https://drive.google.com/a/ncsu.edu/file/d/0B5j0qZXYCftuSVRJRWNmd2FLR0k/view?usp=sharing`

Git Repository:

`https://github.ncsu.edu/pbehera/CSC522_PredictGrantApplications.git`