APPLICATION SIMPLIFICATION WITH MACHINE LEARNING TECHNIQUES

Final Report

THE UNIVERSITY OF SYDNEY

Information Technology Capstone Project

COMP5709

Johnny Peng (440278452)

ABSTRACT

Application forms are now commonly used by corporation and government organisation to acquire personal information from their applicants, which helps the organisation to determine the applicant' suitability.

However, lengthy applications create inconvenience and barriers for the applicants, while it is possible that some of the application questions do not add value to the application process, or unnecessary for a certain cohort of people.

In this research project, we have addressed this issue by developing a machine learning approach for identifying redundant application questions and optimize the application questions for different cohorts of people, so the overall application form is shortened.

Furthermore, the limitation of this research project and guidelines on generalising this approach are discussed.

TABLE OF CONTENTS

Note: The final report has a similar structure to the proposal and it is acceptable to reuse material from the proposal. For example, if your literature review perfectly covered all the relevant material then there is no need to update. More likely, you will need minor revisions to add new material your discovered during the project.

1. INTRODUCTION

General background

The focus of this research project is to develop a machine learning approach for creating shorter and cohort-based application form from the existing application data.

Motivations & Objective:

- Motivation 1: Long list of unnecessary questions in the application forms create inconvenience for the customers.
- Motivation 2 : Companies accumulated tons of application data over time, we need to find a way to turn them into something that is useful for the business.
- Objective: Creating an approach for designing shorter & cohort-based applications with existing application data

Benefits:

The cohort-based application can reduce the number of questions in the application, and hence shorten the application process and create a better customer experience.

Problem description and solutions:

Following are the main problems we have to solve to achieve our objective.

- How to partition the applicants into different clusters based on their features?
  Solution: Hirecial clustering
- How to generate a new set of cohort-based application forms base on the data in each cluster?
  Solution: Decision tree classifier
- How to evaluate the new set of cohort-based application forms against the existing application form?
  Solution: Number of questions asked & Accuracy

More detailed discussion on these problems and their solutions are in the later sessions.

2. RELATED LITERATURE

2.1 Literature Review

We found it difficult to find literature has a similar focus as our research project, however, we have found there is some research done around application data. The literature by Suryana, A., & Yulianto, E. (2019)[1] discussed the application of Association Rule on insurance data for understanding the relationship between insured, product selection and customer behaviour. Another literature by Jeong Jeong, H., Gan, G., & Valdez, E. (2018)[2] discussed the application of Association Rule on nine years of Singapore motor insurance claims and policy data to build empirical evidence of the possible associations between policyholder switching insurer after a claim and associated change in premium. These literature are focused on getting insights out of the application data, which is a different objective to ours. Hence we have also reviewed literature that is related to the methodologies we used in this research project, they will be discussed in the methodologies session.

3. RESEARCH/PROJECT PROBLEMS

3.1 Research/Project Aims & Objectives

The research objective of this research project is to develop a machine learning approach which can generate a new sets of cohort-based application form which meets the following criteria:

1. The new sets of the cohort-based application form should have the same accuracy as the existing application form.

2. The new sets of the cohort-based application form should be shorter than the existing application form.

[1] Suryana, A., & Yulianto, E. (2019). Application of Data Mining with Association Rules to Review Relationship between Insured, Products Selection and Customer Behavior. Universal Journal Of Electrical And Electronic Engineering, 6(2B), 45-61. doi: 10.13189/ujeee.2019.061405

[2] Jeong Jeong, H., Gan, G., & Valdez, E. (2018). Association Rules for Understanding Policyholder Lapses. Risks, 6(3), 69. doi: 10.3390/risks6030069

3.2 Research/Project Questions

The general framework or steps of this machine learning approach can be broken down into the following, and they are also the project questions which will be addressed as part of the research project, by answering the following questions, the project objectives can be achieved.

1. Develop and apply an unsupervised approach for partition the applicants into different clusters based on their features.

2. Develop and apply a supervised model which can generate a new set of cohort-based application forms base on the data in each cluster.

3. Evaluate the accuracy and number of questions asked for the new set of cohort-based application forms, and compare against the existing application form.

3.3 Research/Project Scope

We will be aiming to address all research questions and project objectives by the end of this research project. However, as the project progress, unexpected difficulties may arise, and hence the level of completion may be varied.

4. METHODOLOGIES

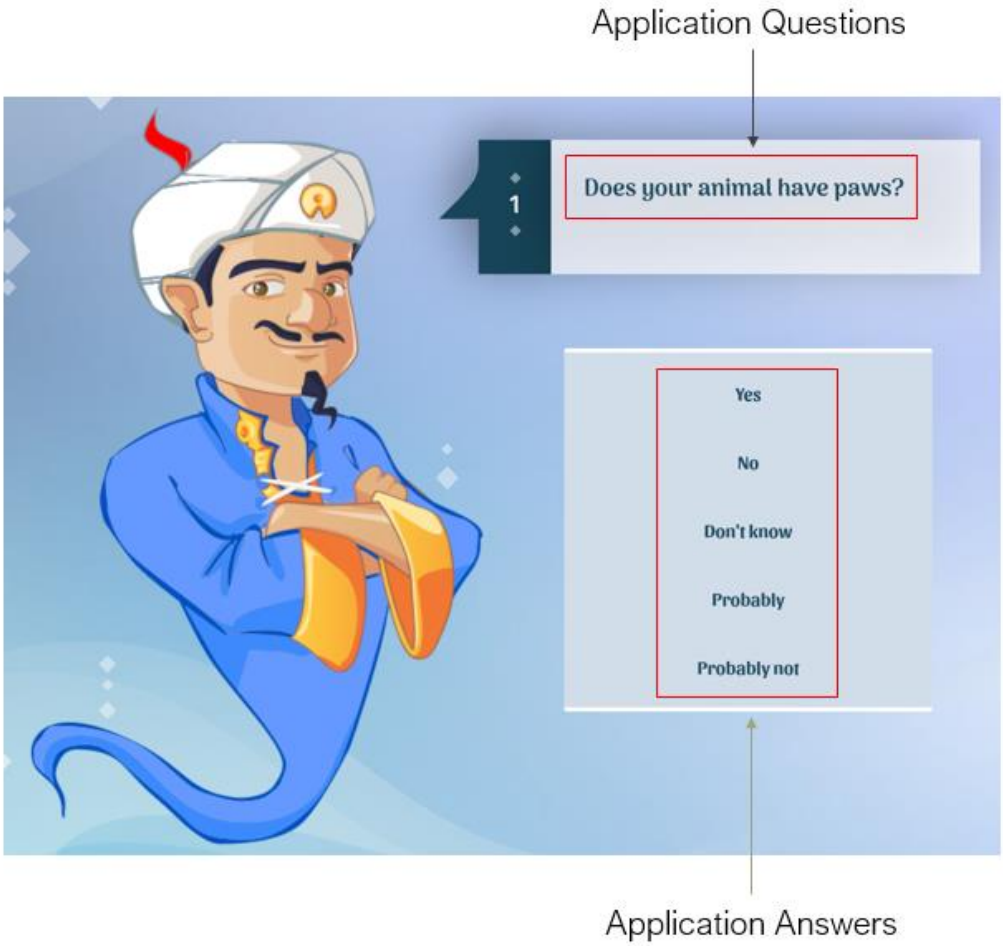
4.1 Data Collection

Before we can collect the data, we will formally define the application data below.

Application data generally comprise the following three components:

1. Application Questions – Questions designed for the applicants to fill in, which will help the service provider to understand the applicant' situation and determine the application outcome.
2. Application Answers – Responses from the applicant to the application questions, note that depending on the circumstances, not all application questions will be answered.
3. Application Outcomes – Service provider's decision on whether to go ahead with providing the service to the applicants.

Overall, the main characteristics of the application data are high-dimensional, and highly sparse since each data points will have different features.



Applications are generally designed to acquire personal data from the customer, and hence application data usually cannot be found from a public source.

Alternatively, we have found an interesting web-based game call “Akinator”, which can be potentially treated as a public source of application data, and we can collect them with a web bot. Akinator is a gene who can guess any animals in players’ mind after asking 25 questions to the player. At the end of each question, depending on the answers given by the player, the follow-up question may vary. There are only 5 possible answers, “Yes”, “No”, “Don’t know”, “Probably”, and “Probably not” can be chosen as the response to the questions from Akinator. After asking a certain number of questions, the Akinator will give his best guess.

When we considering a real-life situation, answers like “Don’t Know”, Probably” and “Probably not” are generally not acceptable in applicaitons, it is the applicant’s responsibility to ensure the accuracy of their answers. Hence, we can further simplify the data by making the following assumptions:

1. Only “Yes” or “No” will be selected as the answers by the web bot.
2. All guess from Akinator will be treated as the correct guess.

Regarding the second assumption, what we found is that the accuracy of Akinator is around 60-80% after playing the game for a few times, however we will just treat them as if they correct, since as long as the data structure is the same as application data, then we can develop a generalisable approach from this toy dataset.

4.2 Data Analysis

Once the data are collected, the next step is to analyse the data and perform data cleaning, what we found is that there are some data points is associated with animals that does not exists in real life or has extincted, which is add complexity for the clustering step later on, hence we have filtered those data out. Also, we have found "Akinator" will just pick a default animal as his answer when he is not sure about the correct answer, hence we have excluded those data points with default answers (e.g. animals that have guessed by Akinator more than 5 times).

After we have done the data cleaning, this game essential forms the source of application data by providing the following three main compoents of application data:

1. Application Questions – 609 unique questions asked by Akinator.
2. Application Answers – Answers selected by the web bot. Possible value includes "Yes", "No", and N/A.
3. Application Outcome – 95 unique animal guessed by Akinator.

4.3 Methods

The methodology to address each research questions are shown below:

<u>Research Question 1 – Unsupervised Model – Hirecial Clustering (Ward's Method):</u>

Firstly, the clustering method will not be applied to the original dataset. Instead, the clustering method will be applied to the latent dataset derived from the original dataset. If a question was asked for a sample (e.g. non-missing) in the original dataset, then the corresponding feature column in the latent dataset will have a value of 1, and a value of 0 if this question was not asked for this sample (e.g. missing) in the original dataset.

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5		Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Sample 1	Yes	Yes		Yes		------>	1	1	0	1	0
Sample 2	Yes		Yes		No	------>	1	0	1	0	1

After applying a clustering method on this latent dataset, we will be able to group the samples that got similar questions together. Then we can allocate a cohort label to these groups by domain knowledge. For example, if the labels for certain clusters are all certain

breed of dog, then we can assign the label of "dog" to this cluster. Once we have the clusters and cluster labels ready, we can train individual decision tree for different clusters, so that the number of questions asked can be reduced for people who only want to play this game with customized questions for a particular group of animals. For example, if we have a cluster label of "dog" and given all the samples labels are at breed level, then we can introduce a shorter question sets for people who only interested in dogs and want to play a "Guess what breed is the dog I am thinking" game.

More importantly, this idea can be generalized for many different types of application data. For example, if we applied this approach for life insurance application data, and having the cluster labels representing life insurance products, then we will be able to have different applications for different products.

There are plenty of clustering methods available, however, due to time constraints we are unable to test all of them, hence we have chosen Hirecial Clustering for this research project simply based on the following reasons:

- Easy to interpret – the output of heretical clustering is a dendrogram which is very easy for humans to interpret.
- No need to specify the total number of clusters – Unlike K-mean clustering, hierarchial clustering allows the modeller to cut the dendrogram at any level, and hence the user can easily inspect the output and decide what kind of clusters will make business senses and allowing the incorporation of business knowledge.

There are plenty of linkage functions available as well, but after reviewing the following linkage function we found that Ward's Method produce the most sensible clustering, as in the commonality in each cluster is very obvious to identify.

Research Question 2 – Supervised Model – Decision Tree (CART):

The decision tree model was chosen to be the supervised model for this research project because it outputs transparent decision rules, and these rules can be treated as an application form, which fits very well with our research objectives.

The main challenge for training on application data is the sparsity of the application data, hence we will need a decision tree algorithm that can handle missing values.

After reviewed the existing classic decision tree algorithm ID3[3], C4.5[4], and CART[5], and we have found that ID3 algorithm is unable to handle missing values, whereas both C4.5 and CART decision tree algorithm has their own way of handling missing values. CART is preferred as the C4.5 handle the missing values by letting the samples follow the implied probability distribution of the primary splits, regardless of the values in other features. Whereas CART decision tree used surrogate splits which will consider the features other than the features that used for primary splits.

We will be evaluating the generated application forms using the following metrics:

- Number of splits - representing the number of questions asked in the application form, the less number of questions needed to be asked to get the answer, the shorter the application form.
- Accuracy – This essentially equals to

$$\frac{\text{number of correct predictions}}{\text{number of total prediction}}$$

The new application forms have to achieve the same accuracy as the existing application form (Akinator), which means an accuracy of 1 or close to 1.

Holdout Dataset

We will not be using holdout/test dataset to evaluate the accuracy of our dataset, because the holdout dataset is not that useful for our purpose, because imagine if we implemented this new set of application forms, all the new application data we collected should not have missing values. whereas if we test this new application using the past application data, we may have the issue of missing values in holdout data and hence underestimate the accuracy of the new application form. As long as the existing application data is large enough to cover most of the application scenario, then the new application form will be able to cover most of the application scenario as well. Also, human inspection of the new application form is another useful check that can validate the quality of the new application forms.

[3] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). Classification and regression trees. Leo Breiman.
[4] Quinian, R. (1993). C4.5. Morgan Kaufmann.
[5] Quinlan, J. (1986). Induction of decision trees. Machine Learning, 1(1), pp.81-106.

5. RESOURCES

5.1 Hardware & Software

Hardware - includes a personal computer, monitor, and a GPU (if required).

Software - We will be using Python & R mainly for this research project, other open-source packages such as Selenium will also be used for this research project.

5.2 Roles & Responsibilities

The student: Responsible for delivering the promised technical outputs.

Academic Supervisor: Responsible for providing technical guidance to the student when required.

6. EXPECTED OUTCOMES

6.1 Project Deliverables

· An generalisable approach of generating shorter cohort-based application form based on the existing application data.

· Results that can prove the cohort-based application generated by such approach is shorter and as accurate as the existing application form.

6.2 Implications

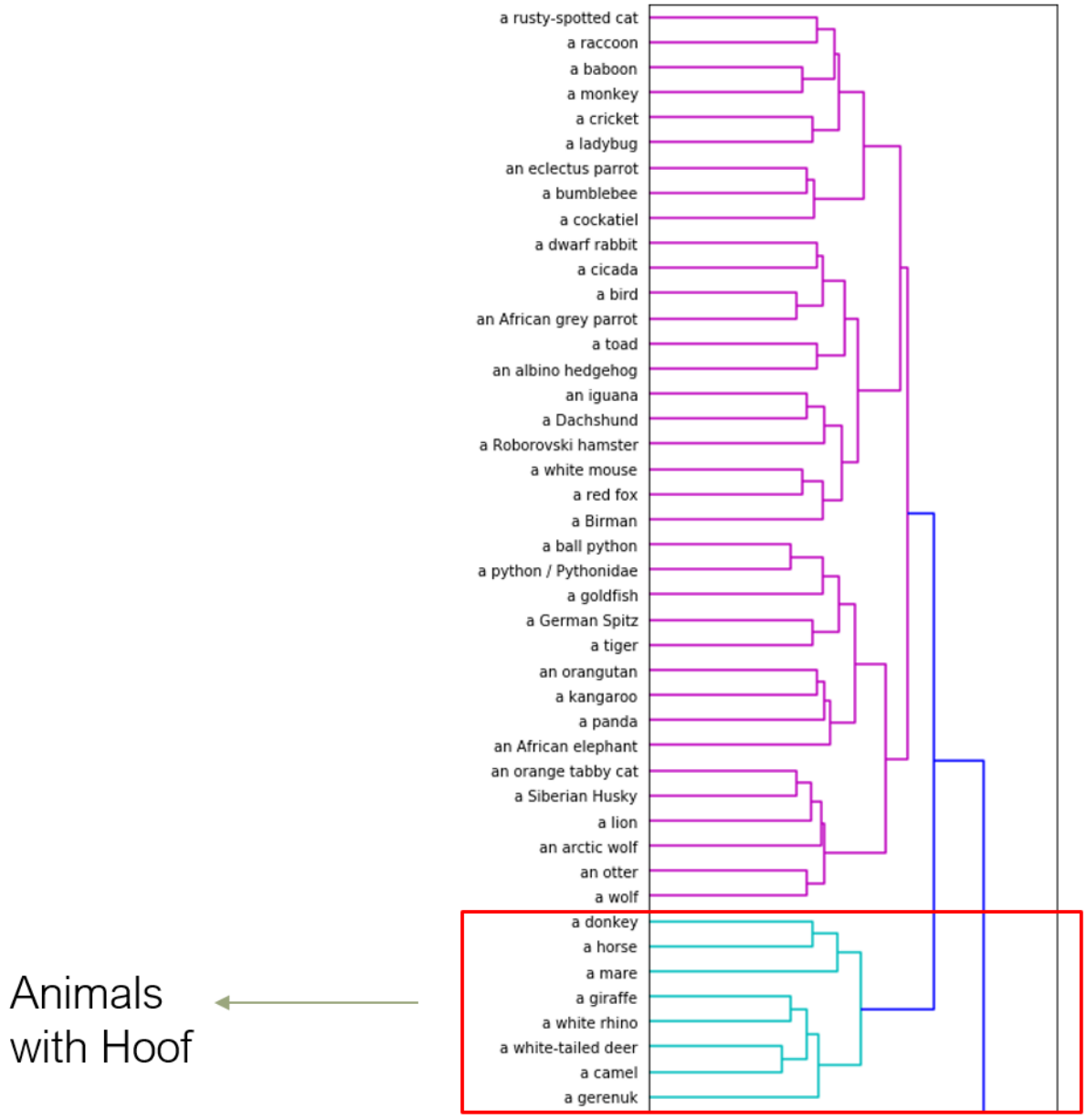
By addressing the research questions and meeting the project objectives, many industry that uses application form and store application data can be benefit from it, and hence benefiting the downstream customer/applicants.

7. RESULTS

Research Question 1 – Unsupervised Model – Hirecial Clustering (Ward's Method):

Based on the output from the hierarchical clustering with Wards' method, there are 4 clusters in total, but only two of them have obvious commonality between the in-cluster samples. The first cluster consists of all the animals with hoof (Hoof cluster), and another cluster consists of dogs and cats (Dog & Cats cluster).

More clusters can be found if we have more data or have the time to look at the clustering in dimensions other than the animal name.

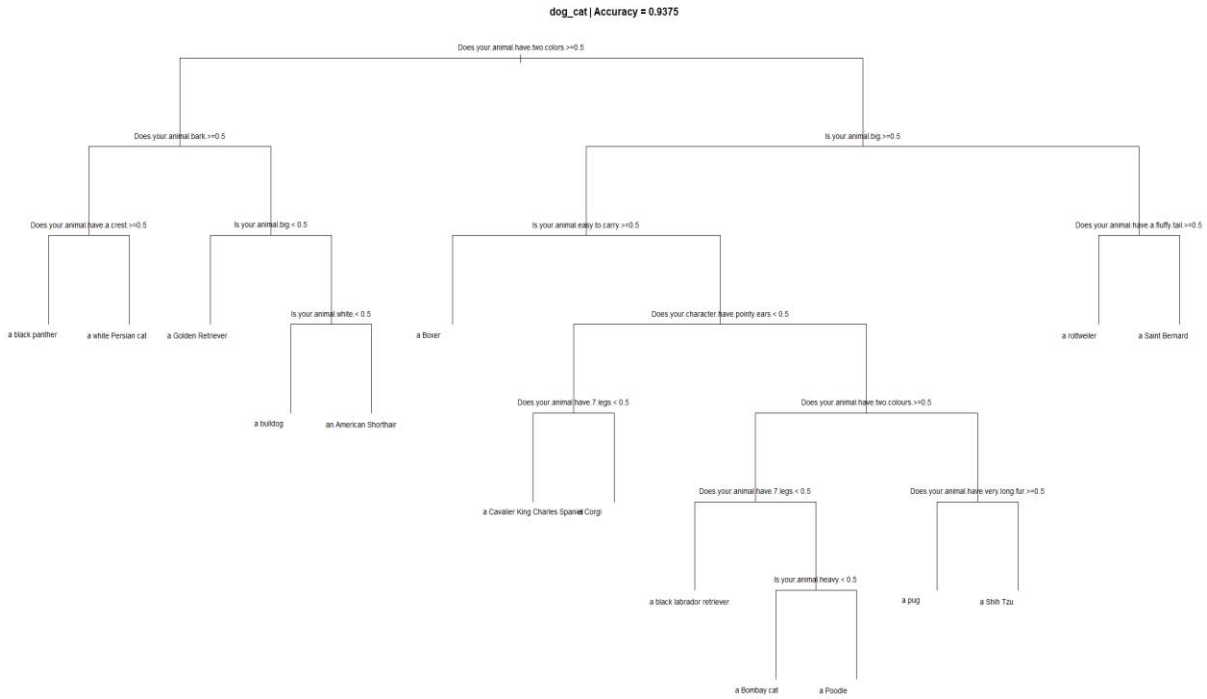


Hoof cluster

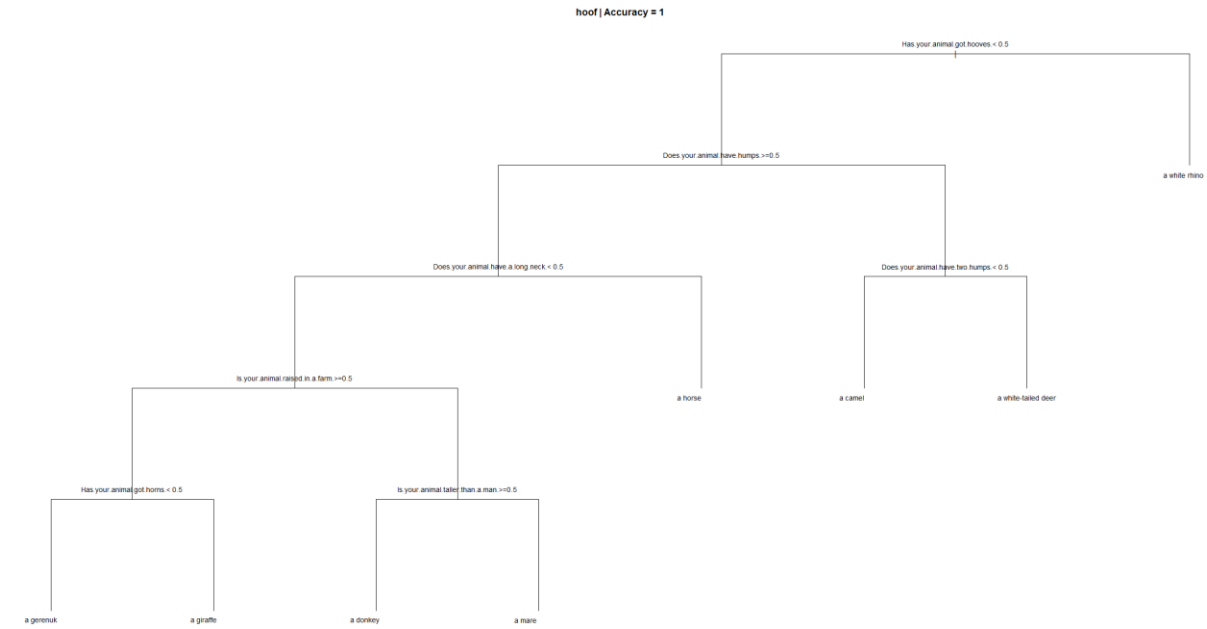
Dog & Cats cluster

Research Question 2 – Supervised Model – Decision Tree (CART):

The next step is to run the CART decision tree on these two clusters separately.



The decision tree above is fitted using the data points in the dog & cat cluster. The result shows that the accuracy of this new application form for dogs and cats is 15/16 = 93.75% and the average number of splits on all the samples is 4.625, which implies this new application form we are able to reduce the average questions asked from 25 (Akinator) to 4.625, while we have only got 1 application wrong. Even if we have to ask a prefilter question to decide whether the applicant can use this application, on average we just need to ask 5.625 questions.



The decision tree above is fitted using the data points in the hoof cluster. The result shows that the accuracy of this new application form for dogs and cats is 8/8 = 100% and the average number of splits on all the samples is 3.75, which implies this new application form we are able to reduce the average questions asked from 25 (Akinator) to 3.75, while still able to achieve an accuracy of 1.

For the samples that are not in these two clusters, the decision tree accuracy for them is nowhere close to 100%, around 75% to 80%, so those samples will need to go through the old application form, however, we probably can generate more clusters if we have more data points available.

Overall, the main things that affected the decision tree result are the samples in the clusters. If the clusters grouping from the hierarchical clustering does not make sense and do not have many commonalities between the in-cluster samples, then the downstream decision tree will also have a very poor accuracy.

8. DISCUSSION & DEPLOYMENT

Generalise to the real-world problem

Now we have shown that this approach can generate new sets of shorter cohort-based application form with accuracy that is almost as good as the existing application. This is good enough for our toy dataset, but how robust is this approach when solving a real-world problem?

Following are some of the major differences in the real-world situation, and their implications.

1. Fewer classes – There will be just "Approved", "Declined", "Refer to Underwriter" etc. which is much less than 95 class of animals.

   The accuracy of the decision tree should improve with fewer classes.

2. More possible answers – Apart from "Yes" or "No", we may have numerical answers. For example, questions like "How old are you?" and "How many years have you been driving?" will require numerical answers from the customers.

   The decision tree may split more than once on the same feature, but when we implementing the new application forms, we should incorporate them as one question.

3. Much more data – Corporations accumulated a lot of application data over time, definitely will have more than what we have in the toy dataset (95 data points).

   More data implies more clusters can be generated from the hierarchical clustering, and hence more customized cohorts-based application form can be generated.

Business Acumen & Cluster Label

It is important to note the following when we deploy this approach to a business problem:

1. Keep the pre-filtering simple - The purpose of the cluster label is so that the internal staff can pre-filter and select the right cohort-based application for customers. Hence the cluster label should be things that are very easily accessible and obvious. For example, application by gender can be easily identified when the internal staff is engaging with the customers, and the internal staff can easily select the right cohort-based application form for the customer. If the cluster label is very complex, then it is equivalent to asking a lot of additional questions to the customer.
2. Avoid a large number of clusters – The more clusters we used, the more types of cohort-based application form will be generated, which also means more costly for the internal staff to find the right application form for the customers.
3. Apply human inspection – After the new application forms are generated, it is very useful to work with relevant business units to inspect and modify the new application form with logical induction and business knowledge.

9. LIMITATIONS AND FUTURE WORKS

The following limitations of our research project may undermine the validity of our results:

1. Scalability - The research dataset is relatively small, and we have not yet tested it on a larger dataset, the results may vary on a larger dataset.
2. Authenticity – The research data are collected and inferred from a web-based game, which may be an oversimplified representation of the real-world application data.
3. Overestimated Improvement – We have demonstrated that this approach can reduce the number of questions asked from 25 to just 3, 4 questions on average. However, Akinator is likely covering slightly more animals than what is covered in our sampled research data (95 unique animals), so if we have to cover the same number of animals, we may need to ask more than just 3,4 questions. However it should still be reasonably lower than 25 questions since we have run the web bot for a very run time, it's unlikely Akinator is covering ridiculously more animals than what we already have.

Hence, more research should be done around generalising this approach to the real-world business cases, where we can test this approach on a larger and more realistic dataset.

REFERENCES

1. Jeong Jeong, H., Gan, G., & Valdez, E. (2018). Association Rules for Understanding Policyholder Lapses. Risks, 6(3), 69. doi: 10.3390/risks6030069

2. Suryana, A., & Yulianto, E. (2019). Application of Data Mining with Association Rules to Review Relationship between Insured, Products Selection and Customer Behavior. Universal Journal Of Electrical And Electronic Engineering, 6(2B), 45-61. doi: 10.13189/ujeee.2019.061405

3. Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and regression trees*. Leo Breiman.

4. Quinian, R. (1993). *C4.5*. Morgan Kaufmann.

5. Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1(1), pp.81-106.