Title

The Effect of Sampling Methods on Model Performance for Classification of Imbalanced Datasets

Author

Weidner, Jeremy

Publication Date

2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

The Effect of Sampling Methods on Model Performance for Classification of Imbalanced Datasets

A thesis submitted in partial satisfaction of the requirements for the Master of Applied Statistics

by

Jeremy Elijah Weidner

2022

ABSTRACT OF THE THESIS

The Effect of Sampling Methods
on Model Performance for Classification
of Imbalanced Datasets

by

Jeremy Elijah Weidner
Master of Applied Statistics
University of California, Los Angeles, 2022
Professor Yingnian Wu, Chair

This paper applies various statistical techniques with the goal of maximizing model performance for the task of classification on a dataset with heavily imbalanced classes. A dataset is created by combining several sources into one comprehensive dataset. Exploratory data analysis will be performed to understand the available factors, their corresponding distributions and relationship to the outcome variable. Then steps will be taken to prepare the data for the task of classification. Next, a collection of different training set sampling strategies will be outlined using methods such as Random Over Sampling, Random Under Sampling and Synthetic Minority Oversampling Technique. Machine learning models such as Random Forest Classifiers will be fitted for each of the sets of parameters and the model fit will be evaluated on the test set in order to provide insight into the differences of various sampling techniques in the imbalanced classification task. Metrics used to evaluate model fit will include traditional statistical measures as well as other strategies that more closely align with the specific business problem.

The thesis of Jeremy Elijah Weidner is approved.

Miles Satori Chen

Frederic R. Paik Schoenberg

Hongquan Xu

Yingnian Wu, Committee Chair

University of California, Los Angeles

2022

This thesis is dedicated to…

My family for their unwavering support,

my friends for making the journey worthwhile,

& the city of Long Beach for making me the person I am today.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

This paper is an overview of the process of refining a solution to optimize its ability to solve a real–world business problem. Precisely, it is an investigation into what statistical methods are most useful when trying to perform a binary classification task on a dataset that contains a heavy imbalance in the outcome variable. This presents complications where some conventional metrics fail to provide the insight needed to tune a solution to fit the problem. This paper will detail the process of exploring this question through the course of data collection, exploratory data analysis, data cleaning & transformation, sampling strategies, model fitting and finally performance evaluation.

The premise of the business problem is that a company sells products to customers and can interact with them via the internet or by having an agent reach out to the potential customer during regular business hours. Prospective customers can use the company's website to enter their information and receive a personalized price estimate for a product from the company online. One of the responsibilities of the company agents is to then follow up with the potential customers who filled out price estimates to try and sell the product. The core problem is that the company agents have more positive outcomes opportunities to follow up on than they have the staff and time to be able to keep up with. Prioritizing contacting customers who are most likely to buy the product is an important part of helping the company agents be as efficient and effective as possible at helping the company succeed.

The data is collected across a period of several months, the two core components of the dataset are the information that the potential customer entered themselves to get the auto insurance price estimate as well as the corresponding web traffic associated with their visit collected via web browser cookies.

The outcome variable represents whether the price estimate resulted in the positive outcome of a product. Naturally there is a window of timing where some price estimates have not yet resulted in a positive outcome but will eventually and so for the dataset curation a 30-day grace period after the price estimate was implemented to give the full allowance of time necessary for a positive outcome to take place. Not employing the grace period would result in false negatives in our training data and that is best avoided.

Another stipulation is that the way the data is collected, the potential customer needs to go through several web pages and enter information on each to receive the price estimated rates at the end of the sequence. To remain in line with the intent of this program the dataset has been restricted to only price estimates and potential customers that completed the steps far enough to see their personal price estimate based on the information they provided. Data points where the potential customer did not make it far enough to see the estimate are excluded from the dataset. This was done because the rate of positive outcome in that cohort was well below the corresponding rate from the group that did complete the price estimate process and so the business priority was to focus on the potential customers who did complete the price estimate process.

Variables have been stripped of their names and replaced with numerical identifiers to preserve proprietary information.

CHAPTER 2

Exploratory Data Analysis

The overall structure of the data is such that each row represents one unique price estimate and has a corresponding unique "Row ID" to identify it. The date range of available data is price estimates given between July – October of 2021. The total record count is 54,654 and record counts by month are shown below:

Month	Record Count
July	14,776
August	15,146
September	17,142
October	7,563
Total	54,654

[Table 2.1: Record Count by Month]

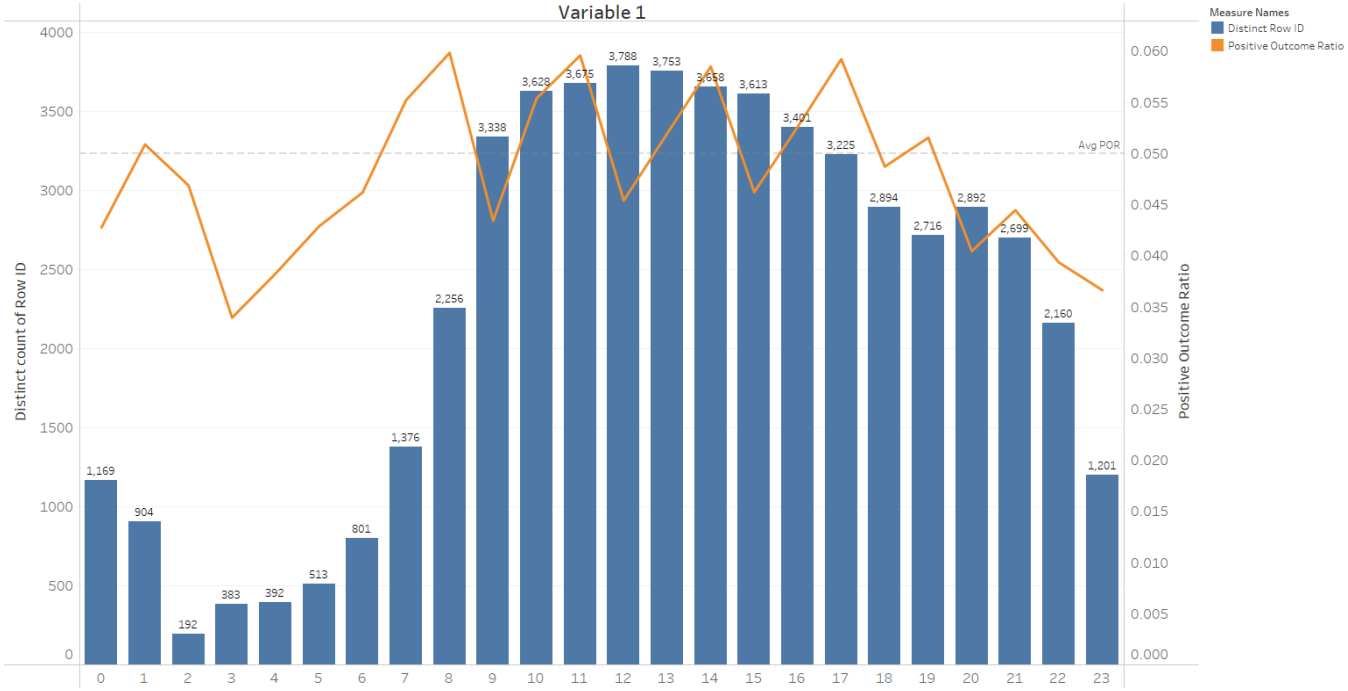
The first thing to note is that October is a partial month of data, with observations through 10/16/2021. This is due to the fact that when this dataset was being aggregated, the observance of a 30 day run-out window from the date of data aggregation to the date the price estimate was created in order to give all price estimates a fair chance to result in a positive outcome and therefore avoid false negatives in the dataset. Taking that into consideration the data is fairly stable month over month with a slight positive trend from July-September but the half-month worth of October indicating an expected slight decrease in record counts between September and October if the second half of the month were to come in on pace with the first 16 days.

The first variable to examine is the outcome variable, "Outcome Indicator", representing whether the price estimate resulted in a positive outcome. It is a binary variable, the price estimate will either result in a positive outcome or it will not. Out of the 54,654 price estimates in the dataset, only 2,733 resulted in a positive outcome. Represented as a percentage that is 5% which will be referred to as the "Positive Outcome Ratio" going

forward. This means that our dataset is rather imbalanced, where 95% of the observations do not result in a positive outcome while only 5% do result in a positive outcome.

The first group of predictors are all deconstructions of the resulting metadata from when the potential customer received the price estimate. That is then used to create several subsequent variables which are Variable 1, Variable 2 and Variable 3.

Variable 1 has discrete values that range from 0-23. This variable can be analyzed using the graph below:

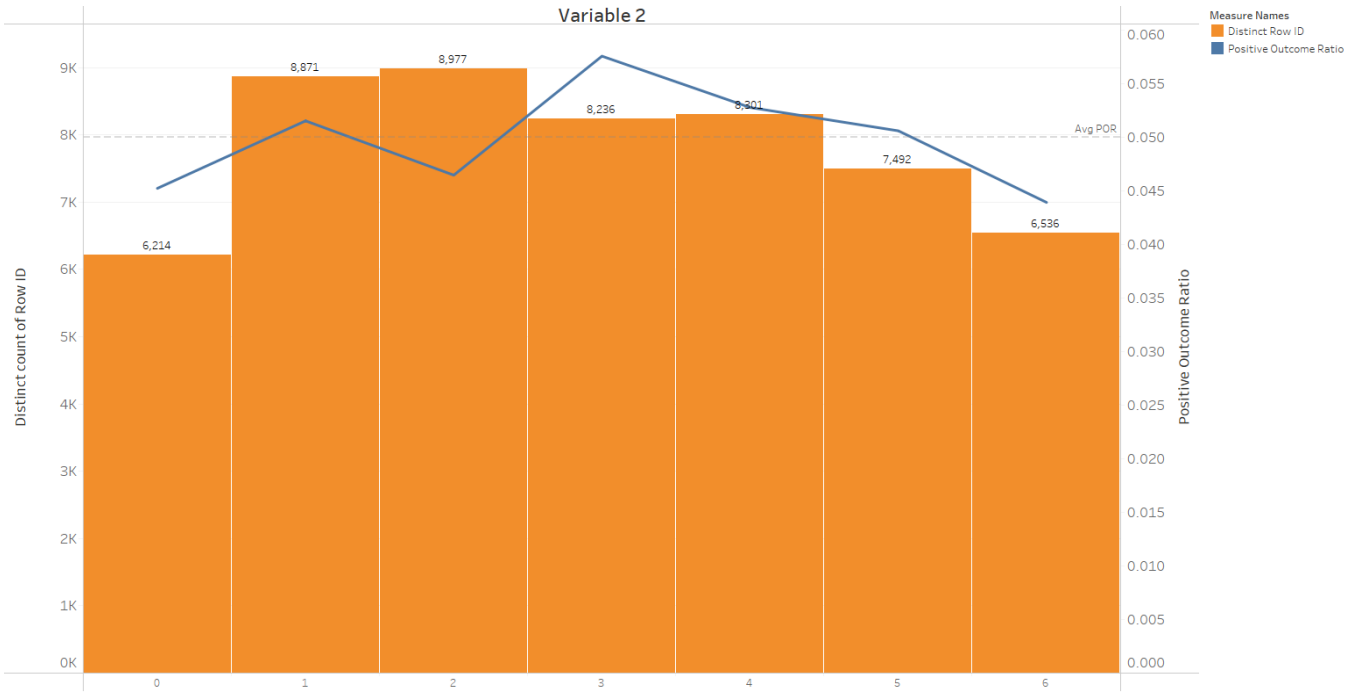


[Figure 2.1: Distribution of Variable 1]

First by looking at the volume we see some clear patterns, most noticeably that there is a lower volume of price estimates being generated in the window of values 23 & 0-7. Then the record count rises to its peak and then tails off throughout the range of 8-22. The orange line corresponding to the Positive Outcome Ratio has significant variability and may be easier to comment on if Variable 1 were bucketed in some way. Due to the low number of observations for some of the values within the range it is hard to put much importance into the

corresponding Positive Outcome Ratio for that value but it does appear that generally the Positive Outcome Ratio is higher for the higher volume values in the range 8-17.

Variable 2 has discrete values that range from 0-6. A visual representation of Variable 2 is below:



[Figure 2.2: Distribution of Variable 2]

Generally more row ID's are created during the middle of the range of values, with 1 and 2 having the most volume and then tailing off gently towards the end of the range. Values 0 and 7 have the lowest volume as well as the lowest Positive Outcome Ratios. We can see relative to the reference line of the dataset's overall Positive Outcome Ratio that row ID's generated on Variable 2 values of 3, 1, 4 and 5 (in order from highest to lowest) are all above the average Positive Outcome Ratio while 2, 0 and 7 are all below the average.

Next is Variable 3 which is a binary indicator variable that represents whether the row ID was generated between the Variable 1 range of 8-21. When examining the data through this lens we see the following:

Variable 3	Count Row ID	Count Positive outcome	Positive Outcome Ratio
Yes	42,837	2,216	5.17%
No	11,790	517	4.39%

[Table 2.2: Variable 3 Counts]

This is consistent with some observations from Variable 1, Row ID's created during the Variable 1 ranges of 8-21 have a slightly higher Positive Outcome Ratio (by 0.78%).

Next available predictor is Variable 5 which is a numerical variable specifically measured in dollars. Variable 5 has observations corresponding to three tiers, representing the robustness of the product offering. The tiers are low, medium and. Summary statistics for the various tiers are in the table below:

Variable 5 Tier	Mean	Standard Deviation	Minimum	25th Percentile	50th Percentile	75th Percentile	Maximum
Low	2,077	1,457	282	1,047	1,728	2,617	54,205
Medium	3,153	1,784	526	1,963	2,688	3,842	65,356
High	3,514	1,925	585	2,216	3,017	4,274	66,325

[Table 2.3: Variable 5 Summary Statistics]

We can see that the relationships between the tiers across the various metrics is fairly consistent, with an increase across the board from low to medium and then to high. It is likely that using all of these three tiered options in model fitting is not necessary and we can just select one to better represent the overall phenomenon without relying on redundant variables.

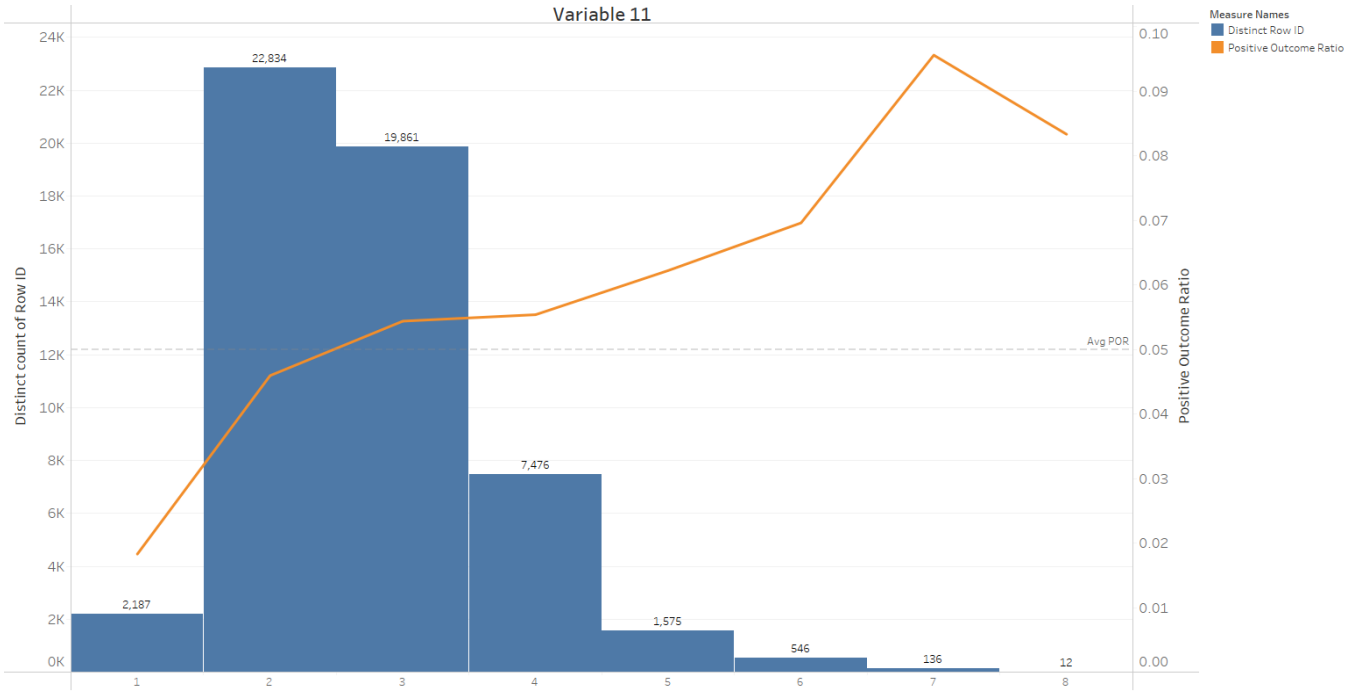
Next up is the Variable 4. This is generated by the company website and corresponds to criteria that the potential customer has entered in the steps leading up to the Variable 5 value being displayed. There are several components that factor into Variable 4's number value and the components are cumulative. Summary statistics for Variable 4 are below:

Mean	Standard Deviation	Minimum	25th Percentile	50th Percentile	75th Percentile	Maximum
13,918	18,418	16	300	6,136	21,391	245,446

[Table 2.4: Variable 4 Summary Statistics]

The first thing to note is the presence of some high outliers, because conceptually this variable should be less than the numerical value of Variable 5 for that individual price estimate. It is most likely that these high outliers and the subsequently high standard deviation are a result of errors in the data collection and it is probably best that these observations are removed before model fitting.

As previously mentioned the numerical value of Variable 4 is the result of cumulative components which are represented separately as Variable 11. A histogram for the Variable 11 is below:



[Figure 2.3: Variable 11 Distribution]

We can see a minimum of 1 and a maximum of 8, with most price estimates falling in the 2-3 range for Variable 11. The high outlier problem is slightly less apparent here when the count of price estimates is tallied up by the Variable 11 counts, but we can see that there is a long right tail. We can see a positive relationship here between the Variable 11 value and the percentage of those price estimates that result in positive outcomes. The

high Variable 11 values such as 7 or 8 are based on a small number of observations so we have insufficient evidence to have much confidence in the POR readings for those values.

The next variable is composed of discrete values between 1-4 and will be referred to as Variable 13. An overview of the variable is below:

Variable 13	Count Row ID	Count Positive outcome	Positive Outcome Ratio
1	36,103	1,722	4.77%
2	15,584	867	5.56%
3	2,239	116	5.18%
4	701	28	3.99%

[**Table 2.5: Variable 13 Summary Statistics**]

Variable 13 cannot have a value smaller than 1, and no records had a value larger than 4. One is the most common Variable 13 value with over double the observations of the next highest Variable 13 value of two. It seems as though Variable 13 values of 2-3 have the marginally higher than average Positive Outcome Ratios where price estimates with Variable 13 values of 1 followed by those with a Variable 13 value of 4 have a subsequently lower Positive Outcome Ratios.

The next available factor is also represented as discrete integers and will be referred to as Variable 14. It is a good indication of whether or not the potential customer has an existing relationship with the company at the time of receiving the price estimate. This variable has a range of zero to three in the dataset and the corresponding summary is found below in the table:

Variable 14	Count Row ID	Count Positive outcome	Positive Outcome Ratio
0	37,812	1,676	4.43%
1	13,921	891	6.40%
2	2,889	166	5.74%
3	5	0	0.00%

[Table 2.6: Variable 14 Summary Statistics]

Price estimates with Variable 14 values of one and two show higher than average Positive Outcome Ratios while those with values of zero show a lower than average Positive Outcome Ratio while having the highest number of observations. Variable 14 values of three only has five price estimates does not have enough observations to lead to much insight. This seems to suggest that the company has better success with customers it already has a relationship with at the time of the price estimate being generated.

The next variable is another collection of discrete integers and will be referred to as Variable 12. The variable's most common value is zero which accounts for 87% of all price estimates. The summary statistics are in the table below:

Variable 12	Count Row ID	Count Positive outcome	Positive Outcome Ratio
0	47,445	2,428	5.12%
1	6,890	294	4.27%
2	290	11	3.79%
3	2	0	0.00%

[Table 2.7: Variable 12 Summary Statistics]

This variable shows a negative relationship with the outcome variable, each increase in the value of Variable 12 shows that the cohort is less likely to have their price estimate result in a positive outcome. Only Variable 12 values of zero is above the average Positive Outcome Ratio while observations with Variable 12 values of 3 do not have enough observations to yield meaningful metrics.

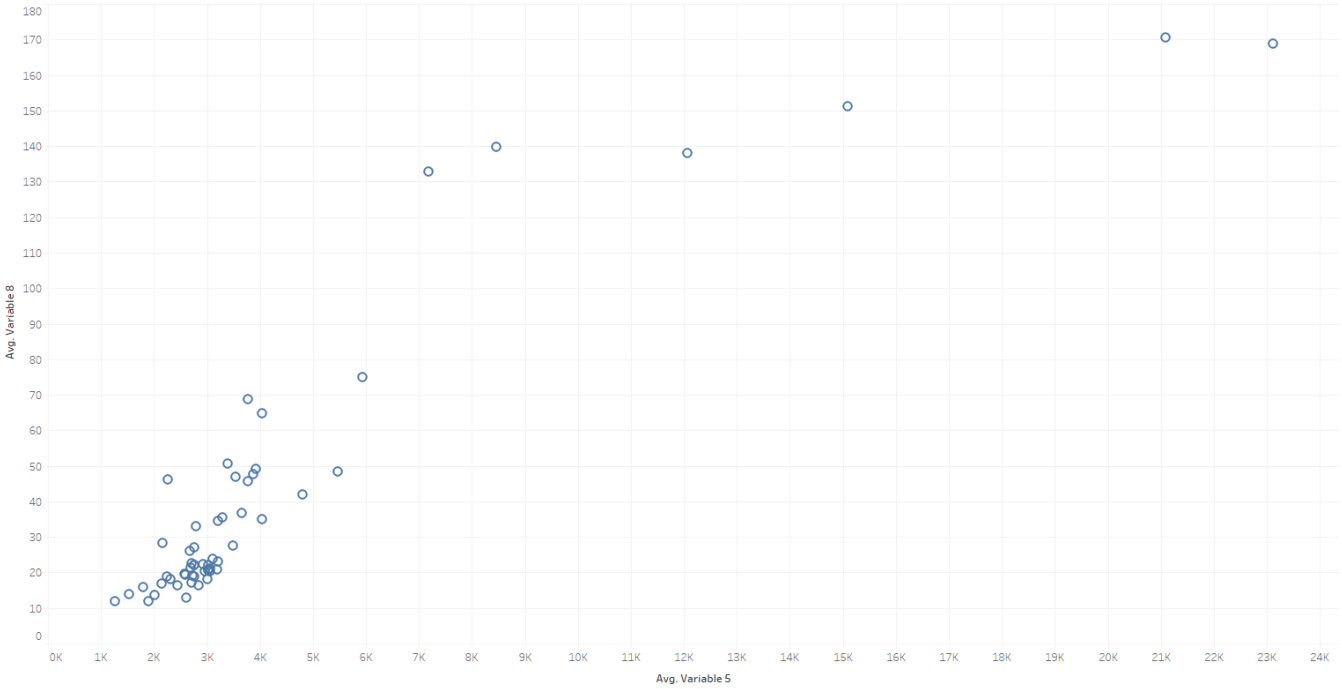
The next variable is another discrete numerical variable and will be referred to as Variable 6. Positive Outcome Ratio summary information is displayed in the table below:

Variable 6	Count Row ID	Count Positive outcome	Positive Outcome Ratio
1	42,129	1,934	4.59%
2	9,104	588	6.46%
3	2,528	155	6.13%
4	865	56	6.47%
6	1	0	0.00%

[Table 2.8: Variable 6 Summary Statistics]

Variable 6 values of one which make up the majority of the dataset is actually the only group with a significant number of observations that is below the average Positive Outcome Ratio. All other Variable 6 price estimate categories such as two, three or four have Positive Outcome Ratios that are at least one percent above the average.

Another piece of information available to us from the customer's price estimate information is Variable 8 which is a continuous numerical variable. To examine this variable we can take an average of the values for all observations based on Variable 18 which is a categorical variable with 58 distinct values in our data set. Examining the average Variable 8 grouped by Variable 18 against the average of Variable 5 gives us some insight as to the distribution of this variable.



[Figure 2.4: Scatter Plot of Variable 8 by Variable 18]

Mean	Standard Deviation	Minimum	25th Percentile	50th Percentile	75th Percentile	Maximum
24.58	11.42	6	18	21	27	292

[Table 2.9: Variable 8 Summary Statistics]

From this summary we see that Variable 8 is another skewed variable in our dataset, there is the main cluster of observations in the low twenties and then some extremely high outliers.

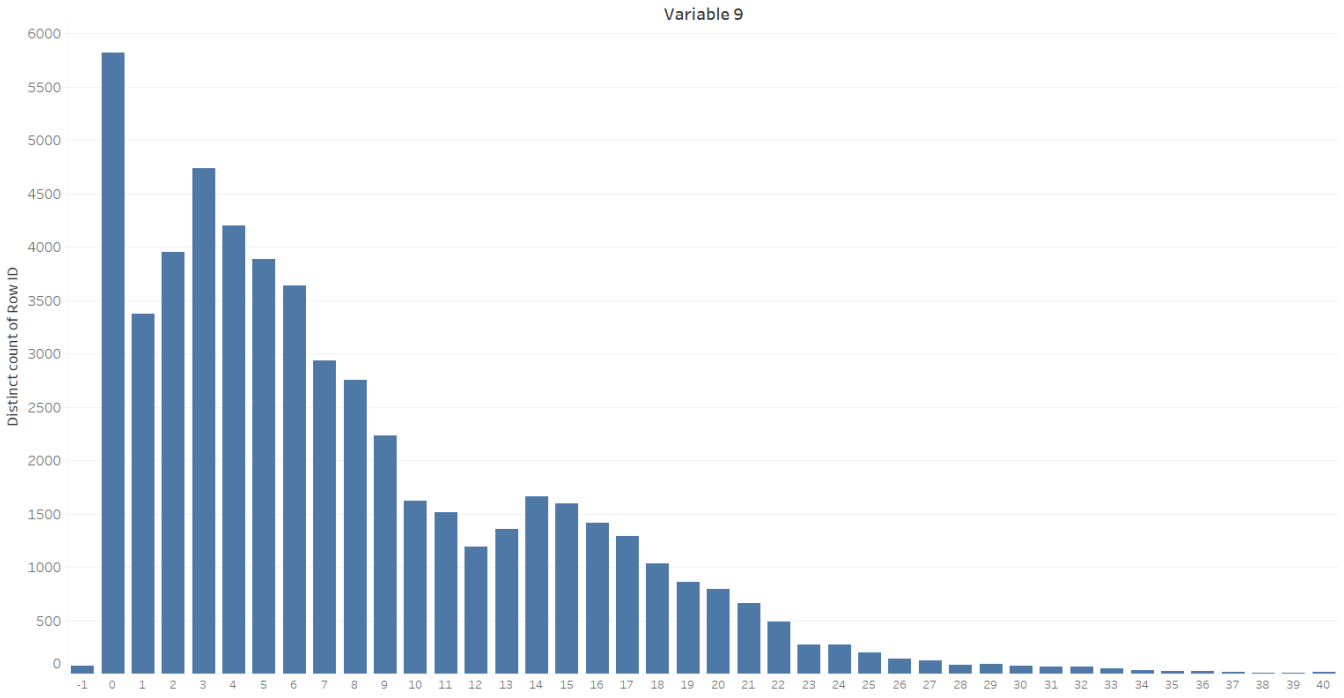
Another continuous numerical variable in the dataset is referred to as Variable 7. It has a fairly wide range of values from a minimum of 346 to a max of 59,364. We can see the distribution below:

Mean	Standard Deviation	Minimum	25^{th} Percentile	50^{th} Percentile	75^{th} Percentile	Maximum
11,667	2,822	346	12,501	12,501	12,501	59,364

[Table 2.10: Variable 7 Summary Statistics]

There are 41,053 observations that have exactly 12,501 as the Variable 7 value for that price estimate. This may have to do with the way the data collection prompt is provided to potential customers as they navigate through the website. This sort of pattern in a continuous numerical variable casts doubt on the reliability of it for model fitting and should be monitored closely.

The next variable is another discrete numerical variable, this one will be referred to as Variable 9. It has the following distribution:



[Figure 2.5: Variable 9 Distribution]

With summary statistics below:

Mean	Standard Deviation	Minimum	25th Percentile	50th Percentile	75th Percentile	Maximum
7.69	6.69	-1	3	6	12	40

[Table 2.11: Variable 9 Summary Statistics]

The variable has a long right tail with the main cluster of observations being < 10. Given the business context of this variable this range of values is reasonable.

Next is Variable 10 which represents the total number of visits to the website that customer had relating to that Row ID. The relevant statistics for this variable are represented in the table below:

Variable 10	Count Row ID	Count Positive outcome	Positive Outcome Ratio	Percent of Total
1	50,594	2,577	5.09%	92.62%
2	812	53	6.53%	1.49%
3	27	1	3.70%	0.05%
4	2	0	0%	0.00%

[Table 2.12: Variable 10 Summary Statistics]

We see here that this Variable 10 predictor is skewed, with 92% of all price estimates having a value of 1. This makes sense as the price estimate process is fairly short and most people should be expected to complete it in one visit.

Next is Variable 16 which contains values as a percentage out of 100. This variable tracks how far through the total number of steps in the online price estimate process the potential customer made it through. The data set contains several versions of the website and so the step labels in the web tracking data vary and this metric was created to track how far a potential customer made it out of the total number of steps at that particular version of the web page. Summary statistics for the Variable 16 are below:

Mean	Standard Deviation	Min	25th Percentile	50th Percentile	75th Percentile	Max
59.9%	2.79%	3.84%	58.33%	60.00%	61.53%	100%

[Table 2.13: Variable 16 Summary Statistics]

Quartile 1 POR	Quartile 2 POR	Quartile 3 POR	Quartile 4 POR
3.62%	4.68%	5.88%	5.50%

[Table 2.14: Positive Outcome Ratio by Quartile of Variable 16]

The data is centered around the 60% mark, this is in part because the data pipeline leading up to this point filters out records that do not reach a certain base percentage. There are however some low outliers, these are probably the result of errors in the collection of the data and those records should be removed.

Following the distribution of Variable 16 we can divide the observations into quartiles and get the Positive Outcome Ratio for each of those quartiles. This is shown in the table above, we can see that the Positive Outcome Ratio increases from quartile 1 to quartile 3 with the 4th quartile having a slightly lower Positive Outcome Ratio.

The next available variable is Variable 17 which is measured in minutes expressed as an integer. Summary statistics are available in the table below:

Mean	Standard Deviation	Min	25th Percentile	50th Percentile	75th Percentile	Max
5.04	6.21	0	2	3	6	117

[Table 2.15: Variable 17 Summary Statistics]

Variable 17 Bin	0	1-3	4-7	8-10	10+
Positive Outcome Ratio	3.13%	4.24%	5.72%	6.96%	5.76%

[Table 2.16: Positive Outcome Ratio by Variable 17 Bins]

This variable shows some outliers on each end but a cluster of observations around the 5-minute mark. We can also probably believe that there are some data collection issues here which may explain the lower outliers as it should be impossible to have a valid observation in 0 minutes. Additionally, the upper outliers, including the max of 117 minutes are probably due to data collection errors as well. It is probably worth considering trimming outlier observations here as well as potentially binning the data to help a model overcome some of the variability in this metric.

When we put the Variable 17 values into bins and take the Positive Outcome Ratio of those bins we can get an idea of the relationship between Variable 17 values and the chance of a positive outcome. The Positive Outcome Ratio grows from the 0 minute bin up until the 8-10 minute bin which has the max value at 6.96%. Then there is a slight decrease for the 10+ minute bin.

That is the end of the exploratory data analysis section in which all available features have been evaluated. The insights gained from this process will help move forward with the effort in several ways. First, where features have shown that they have records which are either invalid values or just outliers there will be an opportunity to use the insights from the exploratory data analysis to eliminate these from our record set that will

be used for the evaluation runs so that these invalid or outlier values do not negatively impact the model training and evaluation process. Then, using the insights on which variables have imbalances in their distributions and the corresponding relationship to the outcome variable, steps can be taken to transform these features into a format that will give the model a better chance at being able to make accurate classifications using that information.

CHAPTER 3

Data Cleaning and Transformation

Based on the findings from the exploratory data analysis several data cleaning measures were performed. First observations were removed that have a Variable 4 value of >= 50,000. Then rows were removed that had several core columns with null values. Then observations with a Variable 8 value of 65 or over were removed. The final dataset for model fitting contains 46,268 rows.

Then the following transformations were applied. Variable 17 values were binned with values following within the following ranges: 0, 1-3, 3-7, 7-10, 10+. Variable 6 is transformed into a categorical where values of 1 are coded as “single” and values > 1 are coded as “multi”. Variable 14 is transformed into a categorical where values of 0 are coded with “no” and values > 0 are coded with “yes”.

Then one-hot encoding[**1**] was applied to the following fields:

- Variable 17
- Variable 6
- Variable 1
- Variable 2
- Variable 3
- Variable 14

The result of this process is that when the final data set will be fed to the model these categorical variables that have the one hot coding will no longer be represented by one categorical variable but will now have one feature per category of that variable, and will have a value of 1 or 0 based on whether that individual record’s feature value falls within that category that the new column represents.

A standard scaling process[**2**] was then applied to the following fields:

- Variable 4
- Variable 5
- Variable 7
- Variable 8
- Variable 9

The scaling process will apply to each observation the process of subtracting the mean and dividing by the standard deviation for each numerical variable which will reduce the overall spread of the distribution. The goal of performing this transformation is to make the numerical variables more standard and have the model training process be less susceptible to outliers.

This leaves us with the final set of 17 post-transformation features for model fitting and evaluation:

- Outcome Indicator
- Variable 10
- Variable 11
- Variable 13
- Variable 16
- Variable 17 (categorical)
- Variable 6 (categorical)
- Variable 14 (categorical)
- Variable 1 (categorical)
- Variable 2 (categorical)
- Variable 3 (categorical)

- Variable 12 (categorical)
- Variable 4 (scaled)
- Variable 5 (scaled)
- Variable 7 (scaled)
- Variable 8 (scaled)
- Variable 9 (scaled)

These 16 variables will be the data elements that are used in the model fitting and evaluation. The categorical variables are fed to the model in the form of one column per category value in that variable due to the one-hot encoding procedure.

CHAPTER 4

Sampling & Modeling Approach

The model fitting strategy is to test the effects of various sampling approaches and analyze the resulting effects on a model's ability to accurately make predictions on our imbalanced set of observations. The goal of this section is to define and detail those approaches. All the different trials will follow the same structure of applying a train test split[**3**] to the dataset and randomly creating a test set that is 25% of the final dataset size. Sampling methods will be applied to the resulting training set and model performance will be evaluated after it is fit on this post-sampling training set.

Before experimenting with the other sampling methods there will be one run with no sampling which will serve as a baseline comparison to the other methods. This run should show us how well a model is able to categorize observations when trained on the naturally heavily imbalanced training set with an overall Positive Outcome Ratio that will be around the data set's original 5%.

The next sampling technique that will be tested is Random Over Sampling**[4].** This technique will over-sample the minority class from the dataset by picking samples at random with replacement. Specifically the minority of price estimates that did result in a positive outcome will be randomly sampled with replacement so that the overall post-sample Positive Outcome Ratio of the training set will be higher than the pre-sample Positive Outcome Ratio. This is achieved by increasing the overall number of records in the training set. This strategy is fundamentally asking the question of whether the model will perform better when the class imbalance in the dataset is reduced by repeating examples from the minority class. Some general concerns about this type of strategy is whether the resulting training set which will end up having duplicate records is able to generalize to the test set and beyond. When we are working with such a heavy imbalance in the data set where the minority class makes up only 5% of the overall records in the data set then to over-sample to reach a balanced dataset will need

to re-sample from the minority class an extremely high number of times to reach the balance and introduces the possibility of the model over-fitting on the heavily over-sampled minority class records in the post-sample training set.

The next sampling technique is Synthetic Minority Over-Sampling Technique (SMOTE)**[5].** SMOTE is an oversampling technique that over-samples the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen[**6**]. The idea with the SMOTE strategy is to try and create a training set that is slightly more varied than the direct over-sampling approach. The post-sample training set will have more records than the pre-sample training set and it will have a higher Positive Outcome Ratio. With SMOTE there will not be records that are direct duplicates in the training set and instead a synthetic collection of records will supplement the minority class instead. The goal here is to be able to carry the model prediction performance outside of training better by having the model train on a dataset that has more variation than the direct over-sampling approach.

The next sampling method being tested is Random Under Sampling**[7].** This technique will under-sample the majority class by randomly picking samples without replacement. The result of this method is a smaller dataset than the pre-sample training set size and it will also have a higher Positive Outcome Ratio. The main difference here compared to the previous sampling strategies is that while we are still reducing the overall class imbalance with all of our sampling methods, this under-sampling approach will do so by reducing the number of records in the training set as opposed to the previous methods which increased the size of the training set by sampling. Something to keep in mind here with a data set that is so heavily imbalanced where the minority class makes up only 5% of the total records, then if you were to under-sample down to a 1:1 ratio and have a resulting training set that is balanced you will by losing a lot of your majority class records. Whether or not this will have a negative effect on model performance is situational. Depending on how homogeneous the majority class observations are determines how much variability may be lost when observations get left out of the post-sample

train set based on random selection. Acknowledging the possibility of missing out on information about the majority class that could be helpful in prediction, the resulting Positive Outcome Ratio will be modified using the sampling strategy parameter where the ratio of majority : minority examples in the post-sample training set is varied.

The parameter that will be adjusted in the testing runs to try and counteract the above phenomenon is called the Sampling Strategy. This will help not have to leave out so many records when over-sampling as well as not have to re-sample the minority class as heavily when under-sampling. When a float value is passed to this parameter it corresponds to the desired ratio of the number of samples in the minority class over the number of samples in the majority class after resampling. Therefore, the ratio expressed as:

Sample Ratio = (Number of samples in the minority class)/(Number of samples in the majority class)

Another parameter that will be situationally adjusted will be the random state. The random state is an integer that is used as a seed by the random number generator component of any of the methods. Modifying this parameter gives the option for a different set of records to be randomly selected at the various steps in the process.

The modeling technique that will be used to evaluate performance for each run of the experiment is a Random Forest model. The random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. A parameter of the random forest that will be evaluated is the number of estimators also known as the number of trees in the forest. Model performance will be evaluated using it's Out-of-Bag Error which is the average error for each training observation calculated using predictions from the trees that do not contain that training observation in their respective bootstrap sample[**8**]. The goal here is to select the model that is performing the best for the training iterations of that sampling run and then take that version of the model to be used in test set evaluation.

CHAPTER 5

Model Fitting and Evaluation

A set of 14 different runs, each with a unique set of parameters will be established and the performance of each run will be compared with the rest of the cohort to understand the strengths and weaknesses of each variation. This collection obviously does not contain every possible set of parameters that it is possible to evaluate but the hope is that the set is enough to help us gain some insight into strategies that are effective for our imbalanced classification problem.

Sample Run	Sampling Technique	Random State	Sampling Strategy	Pre-Sample Training Size	Post-Sample Positive outcome Count	Post-Sample Not Positive outcome Count	Post Sample Training Size	Post Sample Training Positive Outcome Ratio
A	**NONE**	**18**	**N/A**	**34,701**	**1,776**	**32,925**	**34,701**	**5.12%**
B	**OVER**	**18**	**'auto'**	**34,701**	**32,925**	**32,925**	**65,850**	**50.00%**
C	**OVER**	**7**	**'auto'**	**34,701**	**32,886**	**32,886**	**65,772**	**50.00%**
D	**OVER**	**18**	**0.25**	**34,701**	**8,231**	**32,925**	**41,156**	**20.00%**
E	**OVER**	**18**	**0.50**	**34,701**	**16,462**	**32,925**	**49,387**	**33.33%**
F	**OVER**	**18**	**0.75**	**34,701**	**24,693**	**32,925**	**57,618**	**42.86%**
G	**SMOTE**	**18**	**'auto'**	**34,701**	**32,925**	**32,925**	**65,850**	**50.00%**
H	**UNDER**	**18**	**'auto'**	**34,701**	**1,776**	**1,776**	**3,552**	**50.00%**
I	**UNDER**	**25**	**'auto'**	**34,701**	**1,815**	**1,815**	**3,630**	**50.00%**
J	**UNDER**	**1995**	**'auto'**	**34,701**	**1,806**	**1,806**	**3,612**	**50.00%**
K	**UNDER**	**7**	**'auto'**	**34,701**	**1,815**	**1,815**	**3,630**	**50.00%**
L	**UNDER**	**18**	**0.25**	**34,701**	**1,776**	**7,104**	**8,880**	**20.00%**
M	**UNDER**	**18**	**0.50**	**34,701**	**1,776**	**3,552**	**5,328**	**33.33%**
N	**UNDER**	**18**	**0.75**	**34,701**	**1,776**	**2,368**	**4,144**	**42.86%**

[Table 5.1: Sampling Strategy Overview]

Run A has no sampling and the data set remains imbalanced with a 5.12% Positive Outcome Ratio. This will serve as a baseline due to the fact that the data set is left in its natural state and will be useful to compare all other runs against.

Rows B-F are using the over-sampling strategy with modifications to the parameters between them. One of the parameters that varies between the over-sampling runs is that a different random seed is tested to see how

having a different shuffle of the data affects the model performance. The other is the sampling strategy which is tested with values of 0.25, 0.50 and 0.75. The effect of this can be seen in the counts for the Post-Sample Positive outcome Count where the value grows as the sampling strategy parameter value grows. This plays out in the Positive Outcome Ratio value for those runs which also has a positive relationship with the sampling strategy parameter value.

Rows G is where SMOTE was applied. This run uses the SMOTE procedures to generate synthetic samples to represent the minority class in the training set. The result is that there are a balanced number of Positive outcome/No-Positive outcome observations in the training set, where the difference between the Pre-Sample Positive outcome Counts and Post-Sample Positive outcome Counts represents the number of synthetic observations generated by the SMOTE procedure.

The remaining rows, H-N, are various iterations through the under-sampling approach exploring several different random seeds as well as three different sampling strategy parameters. Specifically, Runs H-K are identical other than the random seed which is changed for each run to help us understand the effect of different shuffles of the data and how they influence model performance. Runs K, L, M, N all use the same random seed but experiment with different sampling strategy values. The effect of this can be seen by looking at the Post-Sample Positive outcome Count and Positive Outcome Ratio values for these runs, both of which have a positive ratio with the sampling strategy technique

Random Forest models were applied to each of the 14 sample runs and tree numbers of 100, 200, 300, 500, 750, 1000, 2000 were tested with their out of bag error values compared to the other tree numbers from that sample run. The lowest error for that run was selected for the subsequent steps of the process for that sample run.

Sample Run	100 Tree Error	200 Tree Error	300 Tree Error	500 Tree Error	750 Tree Error	1000 Tree Error	2000 Tree Error
A	0.051785	0.051468	**0.051267***	0.051324	0.051324	0.051267	0.051324
B	0.001716	0.000987	0.001093	0.001109	0.000835	0.000850	**0.000774***
C	0.002098	0.001262	0.001186	0.001003	0.000927	0.000912	**0.000775***
D	0.006026	0.005613	0.005467	0.005175	0.005175	**0.004495***	0.004835
E	0.001863	0.001296	0.001174	0.001114	0.001134	0.001012	**0.000992***
F	0.001822	0.001458	0.001198	0.001059	0.000989	0.001007	**0.000955***
G	0.024935	0.024784	0.024875	**0.024571***	0.024647	0.024571	0.024586
H	0.413007	0.405124	0.397804	0.396678	**0.391329***	0.394144	0.391892
I	0.407163	0.389807	0.398623	0.394215	0.387052	0.388154	**0.388124***
J	0.416113	0.411130	**0.406977***	0.410853	0.408638	0.407530	0.409468
K	0.395041	0.387603	0.381267	0.381818	0.381267	0.380441	**0.378788***
L	0.224775	0.219482	0.219707	**0.216667***	0.220721	0.220045	0.218243
M	0.355856	0.355668	0.347410	**0.344032***	0.349287	0.349287	0.348724
N	0.404681	0.385859	0.388996	0.388996	0.389720	0.386100	**0.383205***

[Table 5.2: Random Forest Error Rates]

*** = best fitting random forest tree count that run**

The above table lists the out of bag error values for each of the sample runs. 2,000 trees was most commonly selected as the optimal number of trees and 500 trees was the next most common. 100 and 200 trees were both never selected as the optimal number of trees in the random forest. For each given run, the version of the random forest with the best fitting tree count according to its out-of-bag error value will be used for the following evaluation of that run.

It is worth calling attention to the differences in the out-of-bag error values themselves between the runs. Before we have even had a chance to evaluate our model on the test set of observations these differences provide us something interesting to analyze. Runs B-F corresponding to the over-sampling runs have the lowest out-of-bag error values for their runs regardless of the number of trees being evaluated. The baseline, Run A, with no sampling generally has a slightly higher out-of-bag error value. Run G out-of-bag errors are slightly higher than Run A, with Run G representing the SMOTE approach. Then the under-sampling runs, H-K, have the highest out-of-bag error values for the tree number evaluation step. What this means is that when the Random Forest Model is attempting to validate it's performance while iterating through the training set, the over-sampling runs are having

the highest number of correct predictions. Something to keep in mind is whether or not this is due to over-fitting on the training data, how we will be able to tell is by evaluating the test set with various metrics to see how performance translates from training to testing for the various sampling strategies.

The resulting random forest models were fit on the test set and resulting accuracy metrics are contained in the table below:

Sample Run	Actual No Positive outcome & Predicted No Positive outcome	Actual No Positive outcome & Predicted Positive outcome	Actual Positive outcome & Predicted No Positive outcome	Actual Positive outcome & Predicted Positive outcome	F1 Score
A	**10,934**	**1**	**632**	**0**	**0.000**
B	**10,921**	**14**	**630**	**2**	**0.006**
C	**10,954**	**20**	**588**	**5**	**0.016**
D	**10,926**	**9**	**631**	**1**	**0.003**
E	**10,923**	**12**	**631**	**1**	**0.003**
F	**10,923**	**12**	**630**	**2**	**0.006**
G	**10,922**	**13**	**632**	**0**	**0.000**
H	**6,290**	**4,645**	**233**	**399**	**0.141**
I	**6,315**	**4,659**	**230**	**363**	**0.129**
J	**6,270**	**4,695**	**212**	**390**	**0.137**
K	**6,057**	**4,917**	**213**	**380**	**0.129**
L	**10,445**	**490**	**570**	**62**	**0.105**
M	**8,856**	**2,079**	**398**	**234**	**0.159**
N	**7,383**	**3,552**	**294**	**338**	**0.149**

[Table 5.3: Confusion Matrix and F1 Score]

This table gives us insight into the classification accuracy of each run when evaluated against the testing set which the model had no previous exposure to. A pattern that is immediately noticeable is that runs A-G showed poor performance in terms of producing true positives and the corresponding F1 score values suffered as a result. This sheds some light on the differences in the out-of-bag error value differences we noticed in the training runs and indicates that these runs may suffer from an over-fitting problem.

Runs H-N which correspond to the under-sampling runs all out-perform runs A-G in the F1 score and show a much higher number of true positives in the testing data. It is important to note that there are also many more false positives for these runs as well, but given the specific business context and intended use case this is less important than maximizing true positives and is an acceptable byproduct of doing so. From the cohort of under-sampling runs M, N and H showed the most promise. It is interesting to note that between runs H, I, J, K the only thing that changed was the random seed for that run. This shows that under-sampling with a 1:1 parity of outcome class examples in the training data under this context of originally imbalanced training data may be susceptible to the sampling shuffle of that particular run. It is possible that the sampling strategy parameter usage in runs M & N provided more robustness and insulation to this variability as less majority-class records are removed from the training set in these runs than in H, I, J, K.

Another accuracy metric that we can use that more closely align with the business use case are the strategy is to rank the test records based on predicted likelihood of a positive outcome in descending order. Then, the records are divided into five even buckets and the actual positive outcome is calculated for each bucket in each run. Performance for this metric can be judged in several ways, important factors include the Positive Outcome Ratio in the first bucket for that run as well as whether the Positive Outcome Ratio decreases in each consecutive bucket from the predictions on the test set. The metrics from this approach are displayed in the table below:

Sample Run	Test Set Positive Outcome Ratio	Bucket 1	Bucket 2	Bucket 3	Bucket 4	Bucket 5
A	**5.46%**	**8.86%**	**6.87%**	**4.97%**	**3.93%**	**2.68%**
B	**5.46%**	**8.82%**	**6.31%**	**5.36%**	**4.24%**	**2.59%**
C	**5.13%**	**7.61%**	**6.48%**	**5.27%**	**3.89%**	**2.38%**
D	**5.46%**	**8.34%**	**7.22%**	**4.93%**	**4.15%**	**2.68%**
E	**5.46%**	**8.82%**	**6.40%**	**5.23%**	**4.15%**	**2.72%**
F	**5.46%**	**8.64%**	**6.74%**	**4.80%**	**4.32%**	**2.81%**
G	**5.46%**	**8.82%**	**6.91%**	**4.50%**	**4.54%**	**2.55%**
H	**5.46%**	**9.68%**	**6.09%**	**5.79%**	**3.67%**	**2.08%**
I	**5.13%**	**8.43%**	**6.09%**	**5.58%**	**3.98%**	**1.56%**
J	**5.20%**	**9.33%**	**6.61%**	**4.48%**	**3.11%**	**2.12%**

K	**5.13%**	**8.69%**	**6.09%**	**5.36%**	**3.72%**	**1.77%**
L	**5.46%**	**9.51%**	**7.00%**	**4.54%**	**3.93%**	**2.33%**
M	**5.46%**	**10.11%**	**6.48%**	**5.06%**	**3.63%**	**2.03%**
N	**5.46%**	**9.64%**	**6.96%**	**4.80%**	**3.76%**	**2.16%**

[Table 5.4: Model Evaluation Bucketing Approach]

Some observations from the table are that the highest Bucket 1 Positive Outcome Ratio is for run M which was the under-sampling run that also had the highest F1 score. The lowest Bucket 1 Positive Outcome Ratio is for run C which was the over-sampling run that also did very poorly in the confusion matrix and F1 score. The Positive Outcome Ratio for each consecutive bucket is descending for all runs so that shows that they are all able to rank the test set records in a way that is directionally correct. It is also interesting to note that even when using this unconventional model performance evaluation metric runs M, H, N still perform the best which is consistent with the more conventional evaluation strategy of the confusion matrix and F1 score. Given the context of the business application, these Bucket 1 Positive Outcome Ratios mean that the model would be able to prioritize opportunities for agents to work much more efficiently than when just following up on opportunities at random.

It is worth noting that the differences between the runs when evaluated by the Bucket 1 Positive Outcome Ratio from this strategy are less drastic than when evaluating with the confusion matrix and resulting F1 score where there were wider proportional differences. This suggests that the variation in business impact of implementing the various strategies is not as large as the variation of more precise conventional statistical model evaluation metrics.

The Random Forest model is also able to give us some insight into which of the variables is the most valuable in terms of informing accurate prediction and here is that feature importance breakdown for the top 7 features of each sample run:

Sample Run	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7
A	**Variable 5 (0.185)**	**Variable 8 (0.112)**	**Variable 9 (0.107)**	**Variable 4 (0.097)**	**Variable 16 (0.046)**	**Variable 7 (0.029)**	**Variable 11 (0.025)**
B	**Variable 5 (0.187)**	**Variable 8 (0.111)**	**Variable 9 (0.106)**	**Variable 4 (0.097)**	**Variable 16 (0.46)**	**Variable 7 (0.030)**	**Variable 11 (0.025)**
C	**Variable 5 (0.187)**	**Variable 8 (0.111)**	**Variable 9 (0.107)**	**Variable 4 (0.98)**	**Variable 16 (0.46)**	**Variable 7 (0.030)**	**Variable 11 (0.026)**
D	**Variable 5 (0.187)**	**Variable 8 (0.111)**	**Variable 9 (0.106)**	**Variable 4 (0.097)**	**Variable 16 (0.046)**	**Variable 7 (0.031)**	**Variable 11 (0.025)**
E	**Variable 5 (0.188)**	**Variable 8 (0.111)**	**Variable 9 (0.107)**	**Variable 4 (0.98)**	**Variable 16 (0.047)**	**Variable 7 (0.031)**	**Variable 11 (0.026)**
F	**Variable 5 (0.187)**	**Variable 8 (0.111)**	**Variable 9 (0.107)**	**Variable 4 (0.97)**	**Variable 16 (0.046)**	**Variable 7 (0.031)**	**Variable 11 (0.025)**
G	**Variable 16 (0.235)**	**Variable 5 (0.080)**	**Variable 9 (0.072)**	**Variable 8 (0.062)**	**Variable 4 (0.058)**	**Variable 17 (0.031)**	**Variable 2 (0.024)**
H	**Variable 5 (0.193)**	**Variable 9 (0.111)**	**Variable 8 (0.107)**	**Variable 4 (0.096)**	**Variable 16 (0.048)**	**Variable 7 (0.027)**	**Variable 11 (0.026)**
I	**Variable 5 (0.194)**	**Variable 8 (0.109)**	**Variable 9 (0.105)**	**Variable 4 (0.095)**	**Variable 16 (0.050)**	**Variable 11 (0.027)**	**Variable 7 (0.027)**
J	**Variable 5 (0.189)**	**Variable 8 (0.110)**	**Variable 9 (0.104)**	**Variable 4 (0.098)**	**Variable 16 (0.048)**	**Variable 7 (0.028)**	**Variable 11 (0.026)**
K	**Variable 5 (0.199)**	**Variable 9 (0.109)**	**Variable 8 (0.106)**	**Variable 4 (0.096)**	**Variable 16 (0.049)**	**Variable 11 (0.028)**	**Variable 7 (0.026)**
L	**Variable 5 (0.193)**	**Variable 8 (0.111)**	**Variable 9 (0.107)**	**Variable 4 (0.098)**	**Variable 16 (0.049)**	**Variable 7 (0.027)**	**Variable 11 (0.026)**
M	**Variable 5 (0.197)**	**Variable 8 (0.111)**	**Variable 9 (0.107)**	**Variable 4 (0.098)**	**Variable 16 (0.050)**	**Variable 11 (0.027)**	**Variable 7 (0.026)**
N	**Variable 5 (0.193)**	**Variable 9 (0.106)**	**Variable 8 (0.107)**	**Variable 4 (0.098)**	**Variable 16 (0.049)**	**Variable 11 (0.026)**	**Variable 7 (0.026)**

[Table 5.5: Feature Importance]

The important features are fairly consistent for each run. Variable 5 consistently in the top two variables by importance, and is first in all runs except for run G. Other than that, the Variable 9 and it's Variable 8 value are consistently among the most important features.

CHAPTER 6

Conclusion

The objective of this research paper is to experiment with strategies that influence model performance when performing classification in an imbalanced dataset. The previous sections have covered topics such as exploratory data analysis where a fundamental understanding of the dataset was established, followed by the steps that were taken to prepare the dataset for classification. Then a set of procedures were defined and executed giving insight into how different sampling strategies affect model performance.

Some insights that this provided were that the runs that utilized Random Under Sampling seemed to provide better model fitting outcomes. This was visible using conventional metrics such as the F1 Score as well as the "bucketing" strategy employed to understand the model performance in scenarios that aligned with the business use case. In particular, runs labeled H, M, and N showed the most promise during testing. It is worth noting that Run H's parameters may be susceptible to the shuffle of the data in the particular random seed that it was run with. We can see that other runs with the same parameters but a different random seed such as runs I, J, K had lower F1 Scores as well as lower Positive Outcome Ratios in the first bucket of the ranked test set. The other highest performing runs M & N however, showed that improvements to the Random Under Sampling approach can be observed when modifying the sampling strategy parameter. The result of this modification in both runs is that the training dataset remains imbalanced and preserves the majority-minority class dynamic but reduces the imbalance of the outcomes for data points in the training set. It is possible that in addition to increasing the model's performance that this sort of sampling strategy could also make the approach more robust to changing the random shuffle of the data set.

It is worth noting that another effect of the Random Under Sampling strategies that showed the most performance was that there were more false positives in the test set evaluation. This could be an issue in some

contexts but given the particular business problem that is aiming to be solved over the course of this research paper false positives were acceptable and less important than maximizing the true positives in the model predictions.

Another interesting observation was that whether we were evaluating model performance using conventional statistical metrics or our less conventional bucketing strategy that better aligned with the business use case, we were able to see that the same runs were performing the best across the different model evaluation strategies. This was an encouraging finding and gives reason to be optimistic that conventional statistical metrics will be directly useful when trying to build a solution to a real-world problem that will be implemented in a business context.

Therefore, we arrived at the end of the analysis and were able to demonstrate the ability to understand a business problem and the available data points, gain insight and perform operations on that dataset, and then iterate through model fitting with different strategies to arrive at an optimized solution to the original business problem. Given the correct context, it may be worth experimenting with Random Under Sampling strategies to manipulate the imbalance in a dataset for the goal of classification.

Future work to expand upon this research could be to run the analysis on a larger data set collected over a longer time period. Especially if there was a way to influence the data collection to get a more well-rounded set of observations and ideally balance out some of the skewed distributions that we saw in some of the variables. It would also be interesting to see the same overall set of sampling experimentation applied to other data sets with various levels of imbalance between their majority and minority classes. It would be interesting to see if the findings from this particular data set still provide useful results when the imbalance in the classes is not quite as heavy. There are of course more combinations of random seeds and sampling strategies that could be evaluated as well as investigating whether other types of models than the random forest see similar effects of these sampling methods.

The final item that is important to mention is that even though some runs showed more promise in addressing the task at hand, their generalizability remains in question. Specifically with runs H-K there is a lot of potential for variability between samples which cast doubt on whether the approach would generalize to other datasets as effectively or not. Future work should be done on the topic to fully understand the uncertainty caused by random under sampling. It would be worthwhile to generate a large number of training datasets and measure the average performance of each method across those collection of training runs, understanding the mean and standard deviation of the model's accuracy under those conditions across many training runs would provide much more insight into the reliability of the under sampling methods.

REFERENCES

[1] “Sklearn.Preprocessing.OneHotEncoder — Scikit-Learn 1.0.2 Documentation.” *Scikit-Learn*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html. Accessed 23 Mar. 2022.

[2] “Sklearn.Preprocessing.StandardScaler — Scikit-Learn 1.0.2 Documentation.” *Scikit-Learn*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. Accessed 23 Mar. 2022.

[3] “Sklearn.Model_selection.Train_test_split — Scikit-Learn 1.0.2 Documentation.” *Scikit-Learn*, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html?highlight=train%20test%20split#sklearn.model_selection.train_test_split. Accessed 23 Mar. 2022.

[4] “RandomOverSampler — Version 0.9.0.” *Welcome to Imbalanced-Learn Documentation!*, https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html. Accessed 23 Mar. 2022.

[5] “SMOTE — Version 0.9.0.” *Welcome to Imbalanced-Learn Documentation!*, https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html?highlight=smote#imblearn.over_sampling.SMOTE. Accessed 23 Mar. 2022.

[6] Chawla, N. V., et al. “SMOTE: Synthetic Minority Over-Sampling Technique.” *Journal of Artificial Intelligence Research*, AI Access Foundation, June 2002, pp. 321–57. *Crossref*, doi:10.1613/jair.953.

[7] “RandomUnderSampler — Version 0.9.0.” *Welcome to Imbalanced-Learn Documentation!*, https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html? Accessed 23 Mar. 2022.

[8] “OOB Errors for Random Forests — Scikit-Learn 1.0.2 Documentation.” *Scikit-Learn*, https://scikit-learn.org/stable/auto_examples/ensemble/plot_ensemble_oob.html. Accessed 23 Mar. 2022.