Machine Learning for Analysis and Prediction of Customer Data

Graham Cooper

1706447

Macewan University

CMPT 496

April 22, 2019

Dr. Calin Anton

Abstract

The goal of the project is to analyze a dataset of bike sales and discover what customer features best explains whether or not they bought a bike. The project will use machine learning to discover what features are important. The machine learning algorithms used are Decision Trees, Neural Networks, Naïve Bayes, Support Vector Machines, and Random Forest. This project will incorporate both cross validation and nested cross validation.

1. Introduction

More features in a dataset may not mean increased predictive power. This is because irrelevant and redundant features will not add to the predictive performance of the learning algorithm and they may cause problems for the learner's predictive capabilities as the learner is trying to classify based off useless feature. Also, more features increase computational resources needed to train the data; therefore, having the fewest number features possible without sacrificing predictive performance much is goal. Over-fit training data is another problem. With regards to over-fit data, the learning algorithm learns the data too well that causes test data to be less accurate in its predictive performance; the learning algorithm learns a very detailed definition of the data and becomes over-fit that prevents the learning algorithm from generalizing unseen data during the testing phase. This is because the learner learned irrelevant information and tries to classify according to that irrelevant information, which will reduce the accuracy. The goal of feature selection is to find the minimum subset of features that can predict the labels. This will result in a reduction of computational resources, reduce redundant and useless features, increase learning efficiency, and an increase in predictive accuracy of the labels.

2. Pre-processing

The dataset that was obtained was from edX.org from a course they offered “Principles of Machine Learning” by Microsoft. The data set contains information on whether or not the customer purchased a bicycle.

2.1 Feature Reduction

The dataset contained 24 features, as shown in figure 1. These 24 features were reduced to 13 features in the pre-processing stage, as shown in figure 2. The data contained 16 749 items, but after removing duplicates the data was reduced to 16 404 items. If an item was a duplicate then the first item was kept and the later ones were dropped from the dataset. Feature engineering was done to transform the data into more usable forms of data. The dataset was also normalized because some of the machine learning algorithms required this process, such as Neural Networks.

2.2 Feature Engineering

The BirthDate attribute was transformed into the customer’s age. This was done because age is an integer and can be more easily used than a person’s birthdate that contains month, day, and year. The new attribute, Age was broken up into four numeric categories with values between 0 and 3: under 29, between 29 and 43, between 44 and 58, and over 58. These values represent specific age groups in society where consumer behaviour should be changing. For example, people under 29 don't usually have careers yet and their wealth is generally lower than older customers who are more likely to have careers. People 29 to 43 will more often have smaller children at home, while people 44 to 58 will usually have children that have left home or about to leave home. People over 58 are usually planning to retire in the near future or have already retired and their children have usually moved out of their house.

The attribute PhoneNumber was transformed into AreaCode that contained the area code within the phone number. AreaCode contained two values 500 and 555, where the rest of the numbers contained irrelevant information. The Gender attribute was renamed into Sex and transformed into binary values. Sex refers to a person's primary sexual characteristics and gender refers to a person's secondary characteristics. Put another way, sex refers to male and female, while gender refers to masculinity and femininity. Sex is a more accurate label than gender because gender refers to how people act, not their biology.

Occupation was transformed into Jobs with five numeric categorical values instead of string categorical values. Education was transformed into EducationType with five numeric categorical values instead of string categorical values. Also, CountryRegionName was transformed into Country with six numeric categorical values instead of string categorical values. YearlyIncome was broken up into Income with four values: less than or equal to $50,000, greater than $50,000 and less than or equal to $75,000, greater than $75,000 and less than or equal to $100,000, and over $100,000. Income was broken up this way because this is a standard practice in research when dealing with people’s income as it represents how much spending power a customer has and it breaks them up into a small number of usable groups.

Lastly, TotalChildren was changed into ChildrenFlag with binary values representing no children and children. The reason for this was because if people have children then they should be more likely to purchase a bike than people who don’t have any children. There is an attribute already, NumberChildrenAtHome that keeps track of more detailed and relevant information than TotalChildren. As TotalChildren would keep track of information that is not as important, such as a 30-year old child that doesn’t live at home.

3.0 Machine Learning

The machine learning algorithms selected for this project were: Decision Trees, Support Vector Machines, Neural Networks, Naive Bayes, and Decision Forests. There are 12 features excluding the label. With 12 features the total number of subsets were 4095. The methods used to find the best subset of features was cross validation. Nested cross validation was performed after to tune the learning algorithms. The random states were fixed so that each algorithm would train and tested on the same data. The machine learning algorithms performance was measured by their accuracy rating.

3.1 Cross Validation

Cross validation was performed on Decision Trees, Support Vector Machines, and Naive Bayes. Cross validation was not performed on Random Forest and Neural Networks. Random Forest is very similar to Decision Trees, but instead Random Forest uses many trees instead of one. With feature size varying between 1 and 12 it becomes unclear what to set the number of trees to be built as, for example, having a higher number of feature size can be more beneficial with more trees, but for a low number of feature size less trees can be beneficial. For these reasons, cross validation was not performed on Random Forest. Neural Networks has a similar problem, it's difficult to find the appropriate network size. The problem is "how many nodes should be in a layer and how many hidden layers should be implemented". A smaller network size will benefit from a fewer number of features, while a larger network will benefit from a larger number of features. Also, some features are binary values while others have multiple values that will affect the size of the Neural Network. For these reasons, cross validation was not performed on Neural Networks.

Naive Bayes used a Gaussian algorithm for classification. The Gaussian Naïve Bayes assumes the mean and standard deviation are the maximum likelihood. Support Vector machines used the RBF kernel, a C value of 1 as the default parameters, and a default gamma value of one divided by the number of features. The sklearn website for their Support Vector Machines describes why these parameters are important:

> When training an SVM with the *Radial Basis Function* (RBF) kernel, two parameters must be considered: C and gamma. The parameter C, common to all SVM kernels, trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. Gamma defines how much influence a single training example has. The larger gamma is, the closer

other examples must be to be affected. (Section 1.4.6.1.3. Parameters of the RBF Kernel)

The Decision Trees used the CART algorithm. CART supports regression and does not compute a rule sets. Also, CART builds binary trees with the feature and threshold that provide the largest information gain at each node. Max depth is the maximum depth of the tree. The default parameters used for Decision Trees max depth was set to where the nodes are expanded until all leaves are pure or until all leaves contain less than the minimum samples split of value one. Max features are the number of features looked at for performing the best split. The max features default setting was to use all the possible features, which used the number of features of each size per item. For example, if the size was six then it used all six features.

3.2 Nested Cross Validation

To reduce the time constraints, the top ten features where chosen by best accuracy along with the top four items of each size between one through twelve for each machine learning algorithm. This resulted in a total of 55 items for Decision Trees, Support Vector Machines and Naïve Bayes.

For Decision Trees, there were only two parameters to be tuned: maximum depth and max features. The maximum depth values were from one through fifteen. The max features values were from one through twelve. No other parameters where adjusted for Decision Trees.

For Random Forests, the top 55 items selected by Decision Trees with cross validation were used as the set of features for nested cross validation. There were only two parameters to be tuned: number of trees built and max features. The max features values were from one through twelve. The number of trees built for each run was 10, 25, 50, 75, and 100. No other parameters where adjusted for Random Forest.

For Support Vector Machines, the tuning parameters were gamma and C. The gamma values were 0.1, 0.01, and 0.001. The C values were 1, 10 and 100. The Gaussian Naïve Bayes has no tuning parameters; therefore, nested cross validation was not performed.

The time need to run Neural Networks was too great to run all 4095 features on with nested cross validation. It was estimated to take at least 28 days with the sklearn package. Instead the union of the top 55 items selected from Decision Trees, Naïve Bayes, and Support Vector Machines were chosen. A total of 114 subsets were the result of the join after the removal of duplicates. The Neural Network was set to two layers with the size of the layers between 1x to 1.5x for the first layer and 0.75x and 1.25x for the second layer. The second layer would never be larger than the first layer. For example, for the size of one the parameters that would be used for tuning would be [(1, 1), (2, 1), (2,2)]. Where the first number in the brackets is the number of nodes in the first layer in the network and the second number in the brackets is the number of nodes in the second layer of the network. Another example is for the size of five where the parameters that would be used for tuning would be [(6, 4), (5, 4), (7, 3), (8, 3), (5, 5), (6, 6), (7, 6), (7, 7), (6, 3), (7, 4), (7, 5), (8, 7), (8, 4), (8, 6), (8, 5), (6, 5), (5, 3)].

4.0 Results

There are many interesting things that were discovered in this project. First, the most important features within the dataset. Second, the size of features that provides the greatest accuracy with the least amount of number of features. Third, the machine learning algorithms that provided the best performance with regard to accuracy. Fourth, the most important features selected for each of the twelve number of feature sizes.

4.1 Feature Importance

Figure 3 shows a bar chart of the most important features by count how many times they occurred across all of the learning algorithms for a total of 274 items. The bar chart in figure 3 can be broken up into three main categories: category one is above 76% with NumberChildrenAtHome at 94.5%, Married at 87.2%, AgeBracket at 78.5%, and Sex at 76.6%; category two is between 45% and 58% with ChildrenFlag at 57.3%, JobType at 47.8%, and EducationType at 45.3%; category three is the remaining features with all below 35% with AreaCode at 34.3%, Income at 33.6%, NumberCarsOwned at 31.7%, Country at 27.4%, and HomeOwnerFlag at 27.0%.

4.2 Feature Size

Figure 4 shows the features grouped by their size and sorted by their mean accuracy. Size seven is the best size for three reasons. First, the accuracy drop with size seven compared to the highest accuracy in that table is very small. Second, the accuracy drop size six from size seven is quite significant compared to the first point above. This means that having a seventh feature is providing a decent boost in accuracy. Third, category one and two from the section 4.1 combine for a total of seven features, which matches size seven as the best number of features for this dataset.

4.3 Learning Algorithm's Performance

Figure 5 shows the sorted mean accuracy grouped by learning algorithm. Naïve Bayes performed the best out of all the learning algorithms then Support Vector Machines comes in second. The next three in order are Neural Networks, Random Forest, and Decision Trees, which perform very similarly with each other.

Figure 6 shows the sorted maximum accuracy grouped by learning algorithm. Support Vector Machines performed the best then followed by Naïve Bayes, Neural Networks, Random Forest, and Decision Trees in that order.

Figure 7 shows the mean accuracy grouped by learning algorithm and size, while figure 8 shows the maximum accuracy grouped by learning algorithm and size. There are a few interesting things with these figures. First, the trend in both figures show that as the size increases from one to seven the accuracy consistently increases with each increase in size. The accuracy essentially tops out at size seven and above size seven the accuracy stays at very similar level and doesn't deviate much in either direction. Second, Naïve Bayes and Support Vector Machines are always in the top two spots for every size for both mean and maximum accuracy. Third, for figure 7, Naive Bayes performs best from size one to five and Support Vector Machines perform best from size six to twelve. For figure 8, Naïve Bayes performs best from size one to four and Support Vector Machines perform best from size five to twelve.

Figure 9 shows Decision Trees mean accuracy grouped by size, while figure 10 shows Decision Trees maximum accuracy grouped by size. The results are similar as before, but with size seven being in the number one spot on both figures.

Figure 11 shows Naïve Bayes mean accuracy grouped by size, while figure 12 shows Naïve Bayes maximum accuracy grouped by size. The results are similar in both figures as before, but with size six performing best with using the least amount of feature and sacrificing little accuracy. Size seven still has performs quite well.

Figure 13 shows Neural Networks mean accuracy grouped by size, while figure 14 shows Neural Networks maximum accuracy grouped by size. The results are similar as before with size seven performing quite well.

Figure 15 shows Support Vector Machines mean accuracy grouped by size, while figure 16 shows Support Vector Machines maximum accuracy grouped by size. The results are similar as before, but with size seven being in the number one spot on both figures.

Figure 17 shows Random Forests mean accuracy grouped by size, while figure 18 shows Random Forests maximum accuracy grouped by size. These results are a little bit different for Random Forests compared to the other learning algorithms. First, size eleven is the best accuracy in both figures; however, reducing the feature size from size eleven to size seven doesn’t sacrifice a huge amount of accuracy. Second, the mean accuracy generally improves every time the size is increased, but it appears that size eight gives the best mean accuracy with the fewest number of features. Third, the max accuracy is best at size six, but not sacrificing too much accuracy to get the smallest size of features.

4.4 The Important Features

This section examines which features the machine learning algorithms chose as the best for each size. The machine learning algorithms are aggregated together in this section. The main trend that is visible with the figures from 19 to 30 is that the best features are usually size minus one and the last feature has a lot of variance. For example, size five will have four features with close to 100% usage and the last feature will vary between a large number of remaining features with very low occurrence. This trend picks the best features up to size five, but after size five a new feature is usually not selected 100% of the time.

Figure 19 shows size one, where NumberChildrenAtHome and NumberCarsOwned are the two most frequent and they occur at 25% each, followed by Income with 20%. Starting from size two a trend starts to build up until size five. In figure 20, size two has NumberChildrenAtHome as the best feature and all sizes above size two also include it. In figure 21, size three has Married as the next best feature and all sizes above size three also include it. In figure 22, size four has AgeBracket as the next best feature and all sizes above size four also include it. In figure 23, size five has Sex as the next best feature and all sizes above size five also include it. It's important to note that these four features are the same as category one in section 4.1 and the order that they occur in is also the same.

In figure 24 and 25, size six and seven has ChildrenFlag as the next best feature, but JobType is closely behind it. In figure 26, with size eight it selects the ChildrenFlag as the next best feature with close to 100% occurrence. After size eight there is not anything of significance to report. These features after size eight don't usually select many features 100% of the time and if they do its usually at a very large size. The remaining feature and size charts are: figure 27 with size nine, figure 28 with size ten, figure 29 with size eleven, and figure 30 with size twelve.

These figures show that the best five features in order of importance are NumberChildrenAtHome, Married, AgeBracket, Sex, and ChildrenFlag. With importance being decided by how soon the learning algorithms selected them with lower number of sizes.

5.0 Limitations

There are a few limitations with this project. First, Random Forest and Support Vector Machines did not sweep the whole grid; therefore, it's possible to improve upon their accuracy. Second, the Neural Network only created a network of two layers and it's possible that adding more layers would increase the accuracy. Third, the Neural Network didn't process all of the combinations of items, but instead only 114 of the 4095 items. It's possible that other items that were not looked at would have provided better accuracy. Fourth, the time limit of this project, being only three and a half months, made it impossible to re-examine the pre-possessing phase and create a new dataset with adjusted features and compare the datasets against each other.

The sklearn package created some limitations for this project. First, their Decision Trees used CART instead of an algorithm such as C4.5, which would have performed better because it doesn't look at the same feature twice like CART may potentially do. Also, with C4.5 a tree can be visualized and understood easier than with CART because CART can potentially look at the same node multiple times. Second, the learning algorithms with sklearn did not utilize the GPU that would have significantly increased processing time and could have potentially made it possible to run Neural Networks on the whole 4095 set of features with nested cross validation and add more than two layers in the network. Using the GPU also could have helped increase the performance of Support Vector Machines that would have made it potentially possible to sweep the whole grid or at least a larger percent of the grid.

Third, Naïve Bayes assumes independence and it out performed Neural Networks, Decision Trees, and Random Forest, but the other learning algorithms should be performing better than Naïve Bayes because it assumes independence, whereas the other learning algorithms don't. One reason why this could have occurred is because of the pre-processing stage, perhaps, there is an issue where some feature or set of features could be modified to increase the predictive accuracy of the learning algorithms.

Fourth, if the sklearn package supported the GPU and/or multi-threading for all the learning algorithms then it might have been possible to run nested cross validation on the entire 4095 items and sweep the whole grid, but this may be too computational intensive to be possible within the time frame of this project, but at the very least it could have expand the nested cross validation to more than 55 items for each learning algorithm.

Fifth, the sklearn package supported a very limited number of ensemble methods. They supported Bagging, Random Forest, AdaBoost, Gradient Tree Boosting, and Voting. It would have been interesting to use other of ensemble methods such as Error-Correcting Output Codes, Mixture of Experts, Stacked Generalization, and Cascading. However, with the computational recourses required for this dataset, the pre-processing stage, the goal of the project, and the limitations of sklearn caused issues with many of these ensemble methods.

6.0 Conclusion

This project has identified with regard to the bikes dataset a number of interesting findings. Naïve Bayes provided the best mean accuracy while Support Vector Machines provided the best maximum accuracy. The twelve features were broken up into three distinct categories of importance based on how frequent they occurred in the learning algorithms. The best size for most of the learning algorithms for mean and maximum accuracy was size seven. The best five features in order of importance were NumberChildrenAtHome, Married, AgeBracket, Sex, and ChildrenFlag.

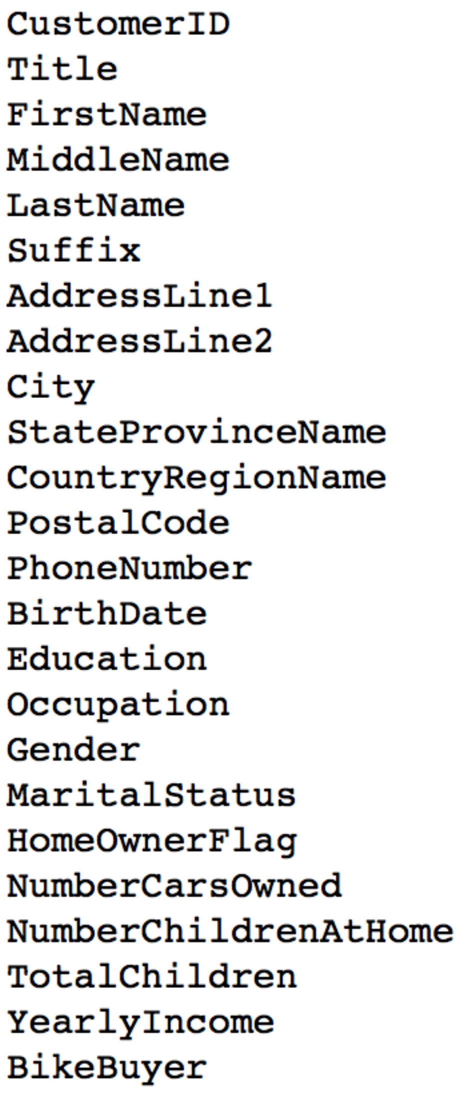
References

SVM. (n.d.). Retrieved From https://scikit-learn.org/stable/modules/svm.html#svm-kernels

Appendix

Figure 1

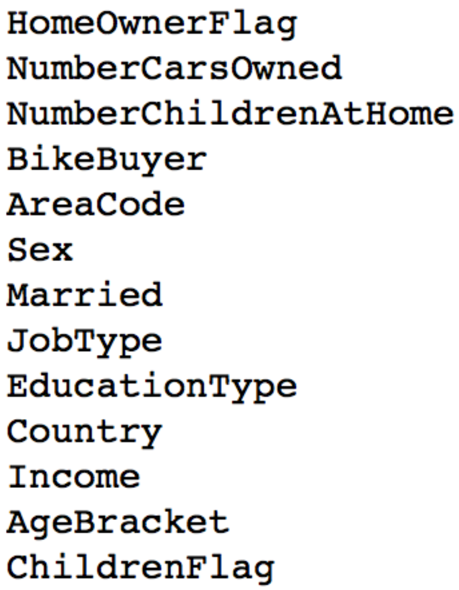
Original Features



```
CustomerID
Title
FirstName
MiddleName
LastName
Suffix
AddressLine1
AddressLine2
City
StateProvinceName
CountryRegionName
PostalCode
PhoneNumber
BirthDate
Education
Occupation
Gender
MaritalStatus
HomeOwnerFlag
NumberCarsOwned
NumberChildrenAtHome
TotalChildren
YearlyIncome
BikeBuyer
```

Figure 2

List of Features after Feature Engineering



```
HomeOwnerFlag
NumberCarsOwned
NumberChildrenAtHome
BikeBuyer
AreaCode
Sex
Married
JobType
EducationType
Country
Income
AgeBracket
ChildrenFlag
```

Figure 3

Total Feature Count

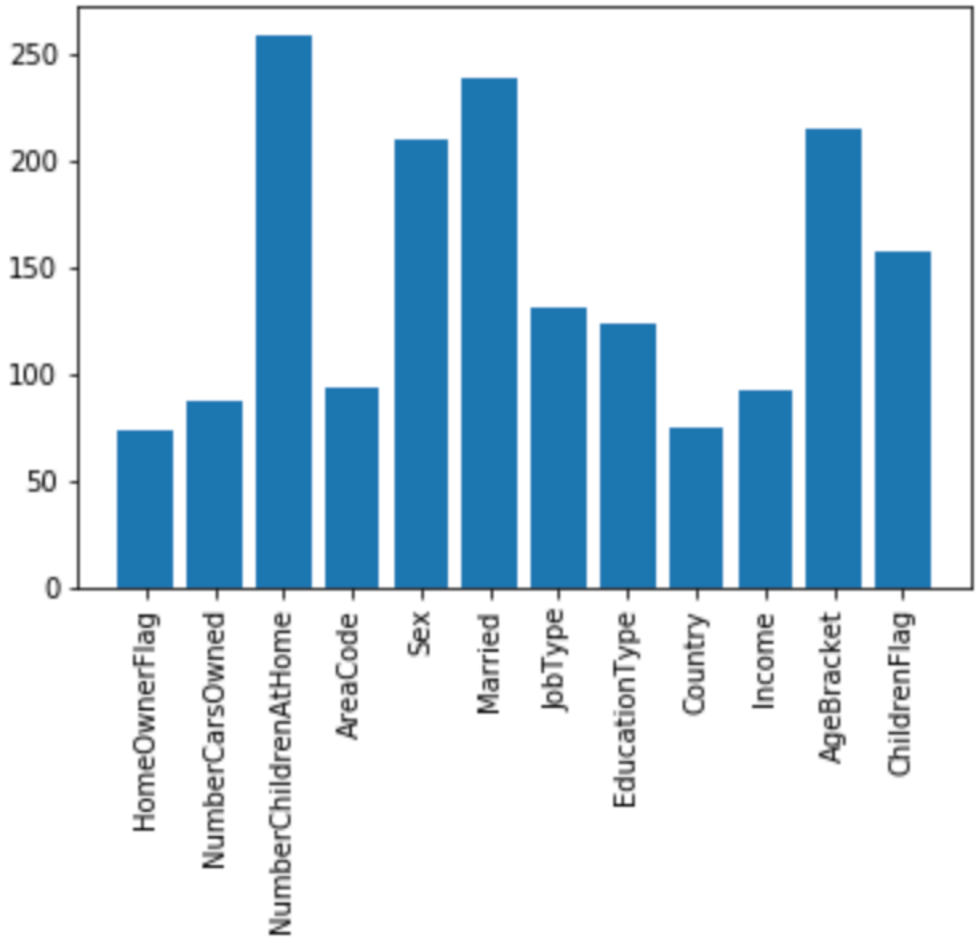
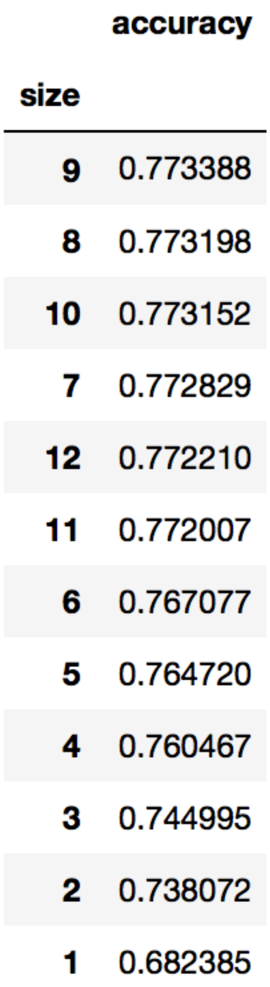


Figure 4

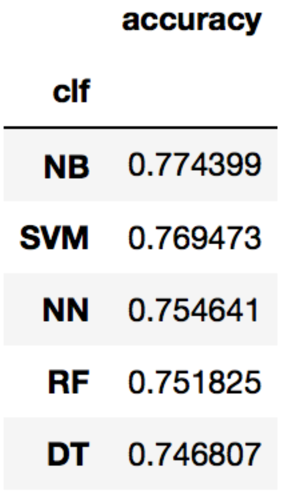
Mean Accuracy Grouped by Size



	accuracy
size	
9	0.773388
8	0.773198
10	0.773152
7	0.772829
12	0.772210
11	0.772007
6	0.767077
5	0.764720
4	0.760467
3	0.744995
2	0.738072
1	0.682385

Figure 5

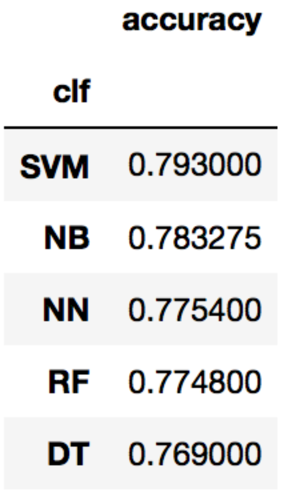
Mean Accuracy Grouped by Learning Algorithm



	accuracy
clf	
NB	0.774399
SVM	0.769473
NN	0.754641
RF	0.751825
DT	0.746807

Figure 6

Max Accuracy Grouped by Learning Algorithm



	accuracy
clf	
SVM	0.793000
NB	0.783275
NN	0.775400
RF	0.774800
DT	0.769000

Figure 7

Mean Accuracy Grouped by Size and Learning Algorithm

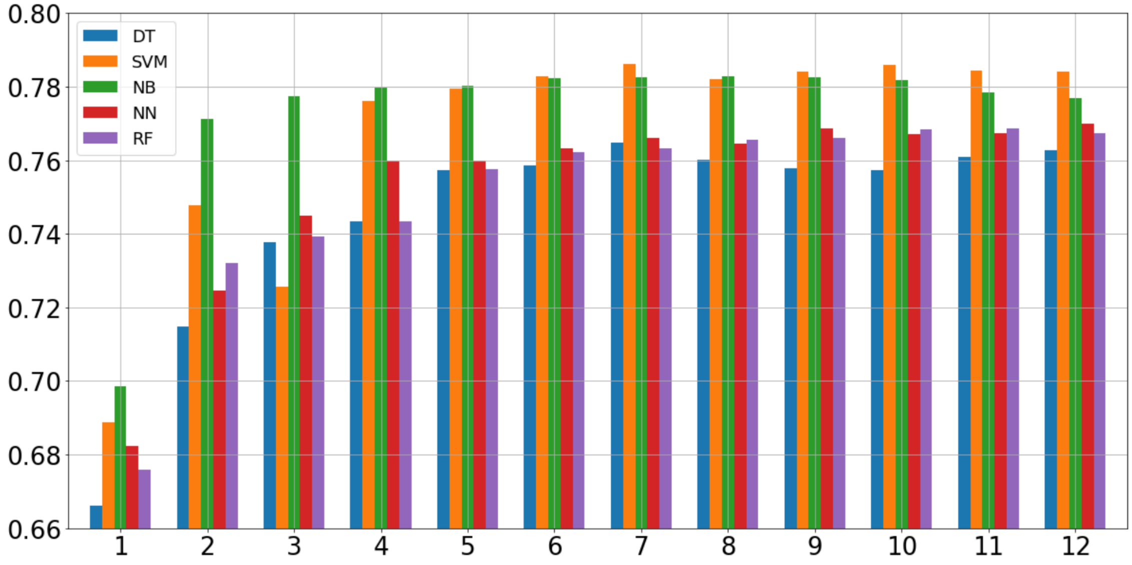


Figure 8

Max Accuracy Grouped by Size and Learning Algorithm

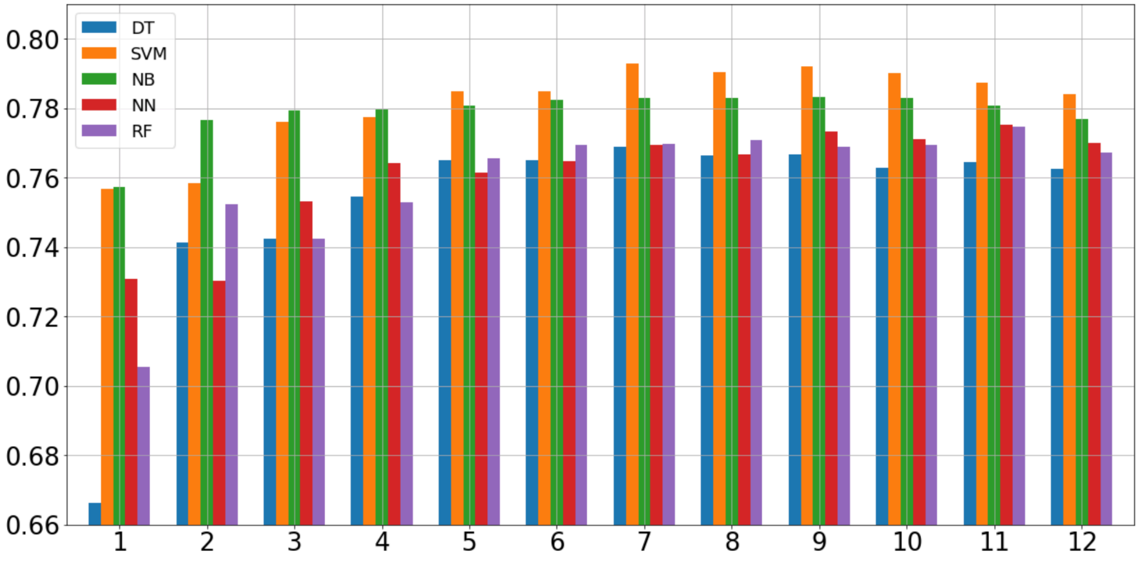
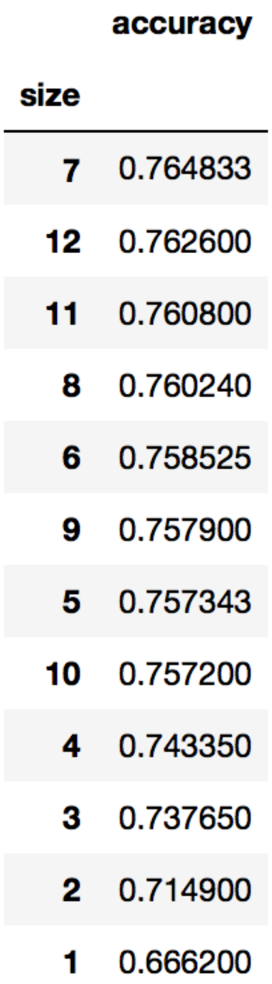


Figure 9

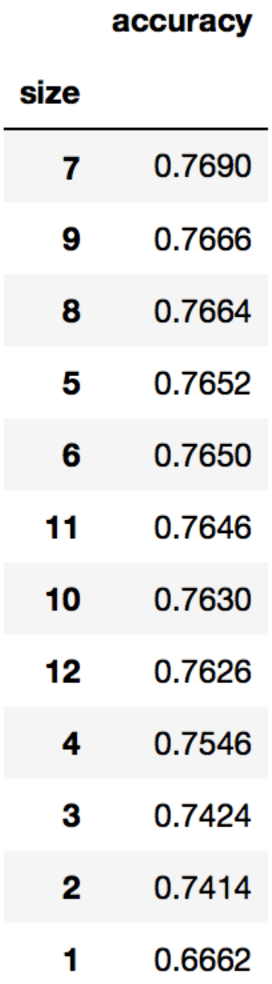
Decision Trees Mean Accuracy Grouped by Size



	accuracy
size	
7	0.764833
12	0.762600
11	0.760800
8	0.760240
6	0.758525
9	0.757900
5	0.757343
10	0.757200
4	0.743350
3	0.737650
2	0.714900
1	0.666200

Figure 10

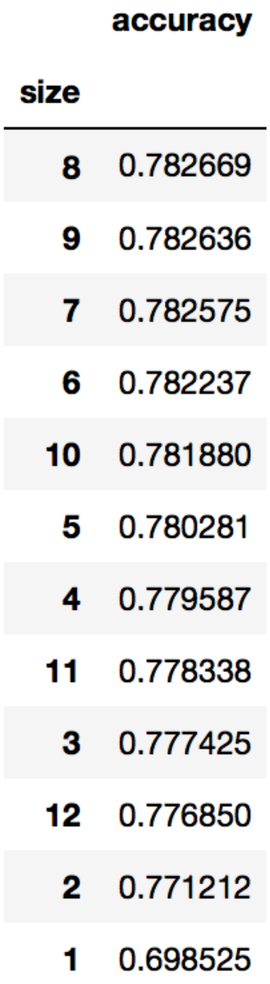
Decision Trees Max Accuracy Grouped by Size



	accuracy
size	
7	0.7690
9	0.7666
8	0.7664
5	0.7652
6	0.7650
11	0.7646
10	0.7630
12	0.7626
4	0.7546
3	0.7424
2	0.7414
1	0.6662

Figure 11

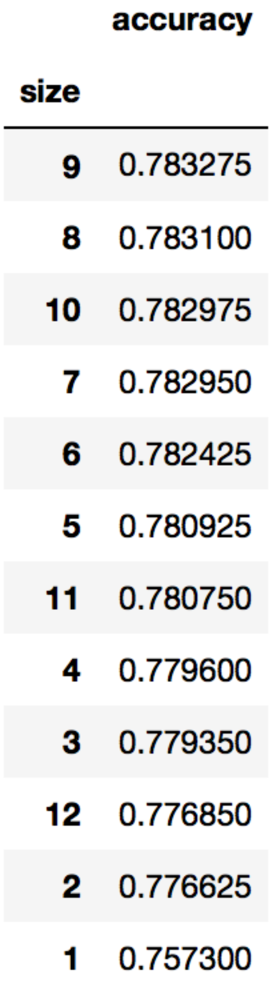
Naïve Bayes Mean Accuracy Grouped by Size



	accuracy
size	
8	0.782669
9	0.782636
7	0.782575
6	0.782237
10	0.781880
5	0.780281
4	0.779587
11	0.778338
3	0.777425
12	0.776850
2	0.771212
1	0.698525

Figure 12

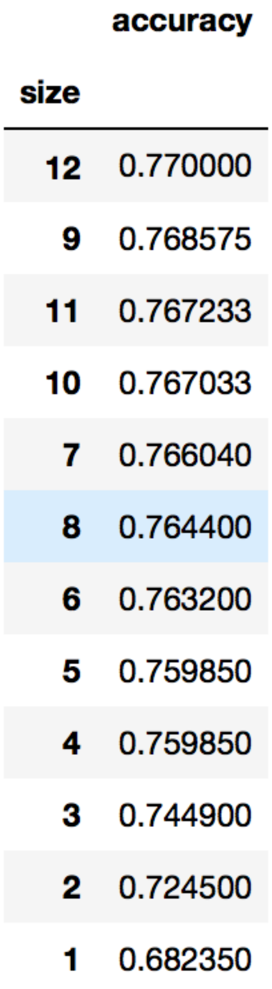
Naïve Bayes Max Accuracy Grouped by Size



	accuracy
size	
9	0.783275
8	0.783100
10	0.782975
7	0.782950
6	0.782425
5	0.780925
11	0.780750
4	0.779600
3	0.779350
12	0.776850
2	0.776625
1	0.757300

Figure 13

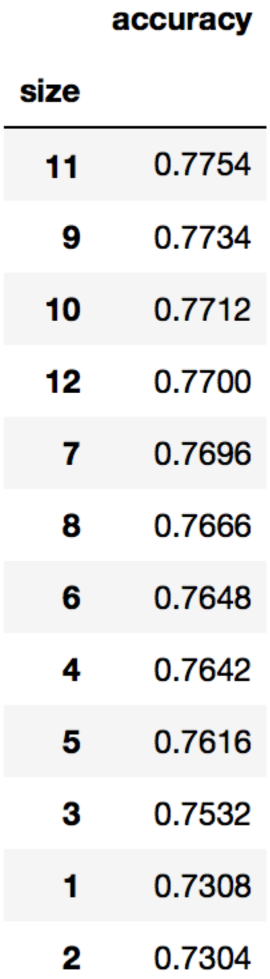
Neural Networks Mean Accuracy Grouped by Size



	accuracy
size	
12	0.770000
9	0.768575
11	0.767233
10	0.767033
7	0.766040
8	0.764400
6	0.763200
5	0.759850
4	0.759850
3	0.744900
2	0.724500
1	0.682350

Figure 14

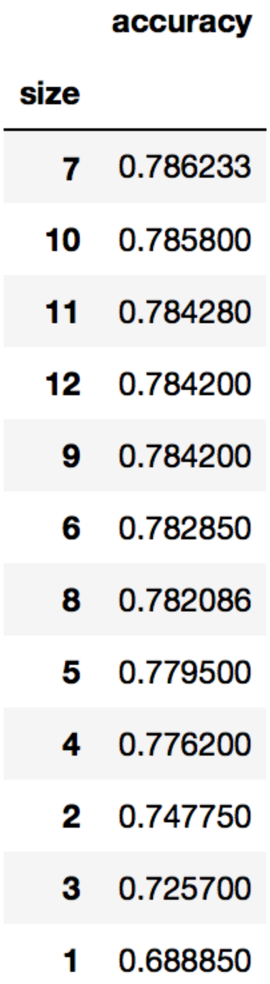
Neural Networks Max Accuracy Grouped by Size



	accuracy
size	
11	0.7754
9	0.7734
10	0.7712
12	0.7700
7	0.7696
8	0.7666
6	0.7648
4	0.7642
5	0.7616
3	0.7532
1	0.7308
2	0.7304

Figure 15

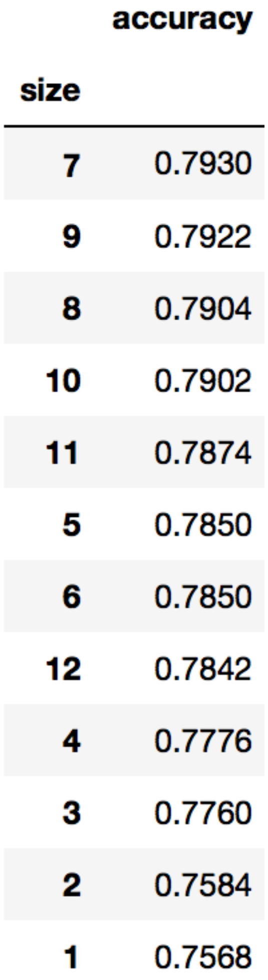
Support Vector Machines Mean Accuracy Grouped by Size



	accuracy
size	
7	0.786233
10	0.785800
11	0.784280
12	0.784200
9	0.784200
6	0.782850
8	0.782086
5	0.779500
4	0.776200
2	0.747750
3	0.725700
1	0.688850

Figure 16

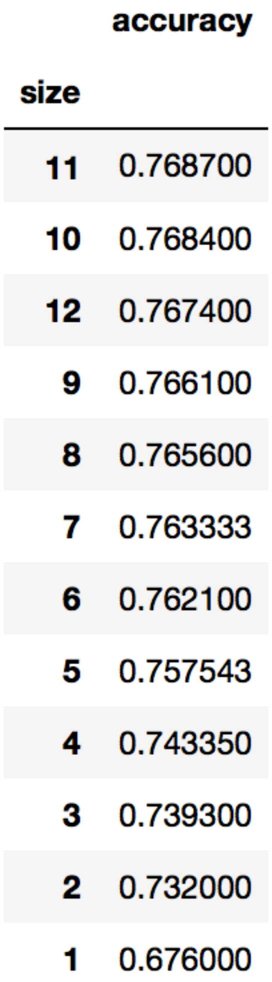
Support Vector Machines Max Accuracy Grouped by Size



	accuracy
size	
7	0.7930
9	0.7922
8	0.7904
10	0.7902
11	0.7874
5	0.7850
6	0.7850
12	0.7842
4	0.7776
3	0.7760
2	0.7584
1	0.7568

Figure 17

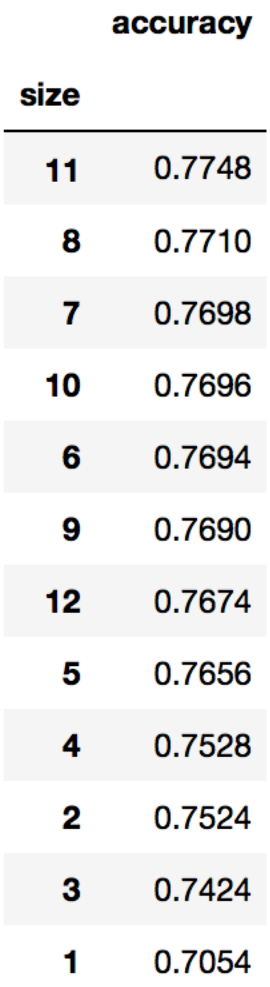
Random Forest Mean Accuracy Grouped by Size



	accuracy
size	
11	0.768700
10	0.768400
12	0.767400
9	0.766100
8	0.765600
7	0.763333
6	0.762100
5	0.757543
4	0.743350
3	0.739300
2	0.732000
1	0.676000

Figure 18

Random Forest Max Accuracy Grouped by Size



	accuracy
size	
11	0.7748
8	0.7710
7	0.7698
10	0.7696
6	0.7694
9	0.7690
12	0.7674
5	0.7656
4	0.7528
2	0.7524
3	0.7424
1	0.7054

Figure 19

Feature Frequency at Size One

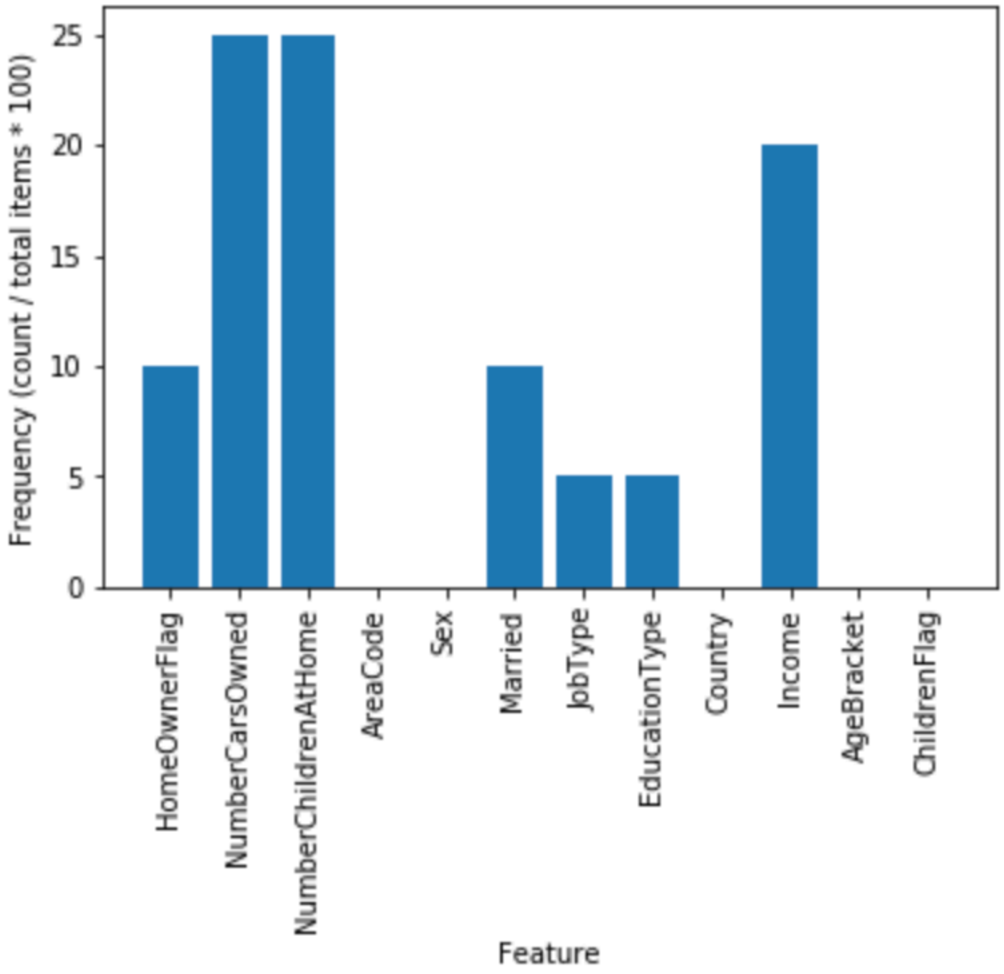


Figure 20

Feature Frequency at Size Two

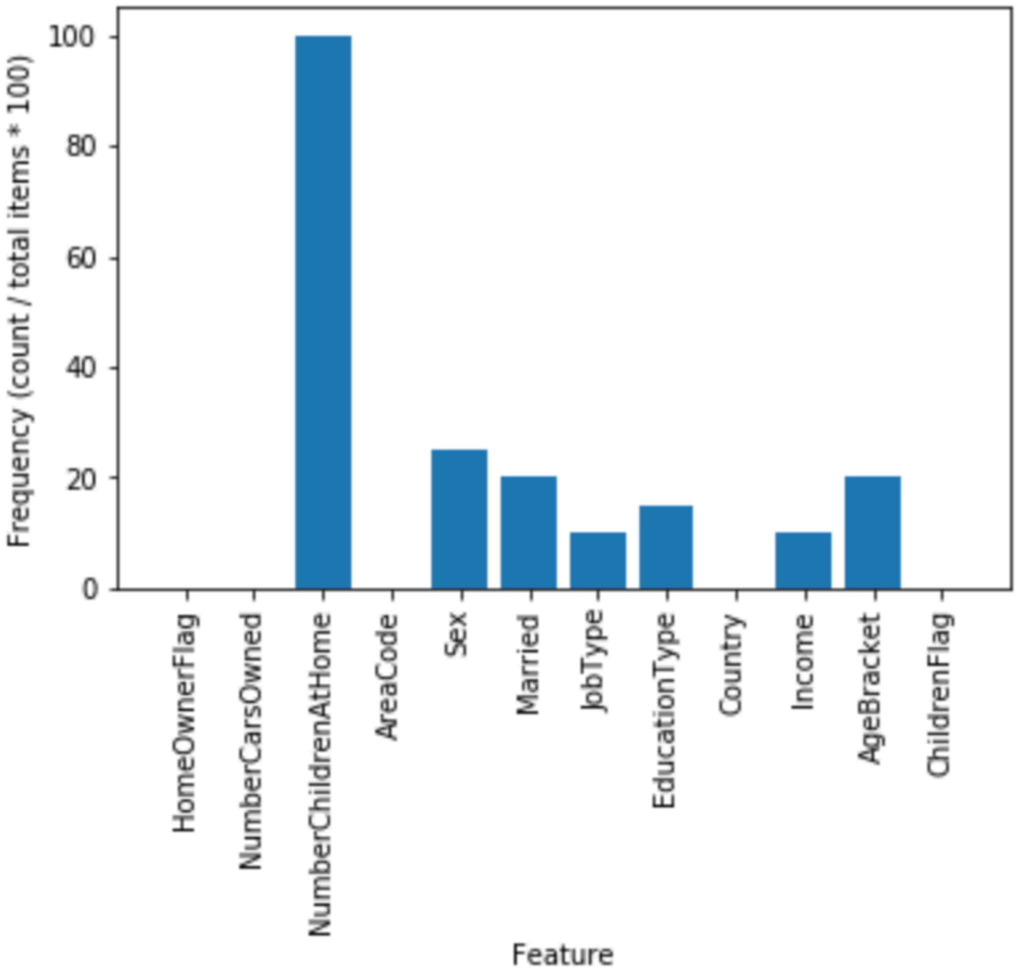


Figure 21

Feature Frequency at Size Three

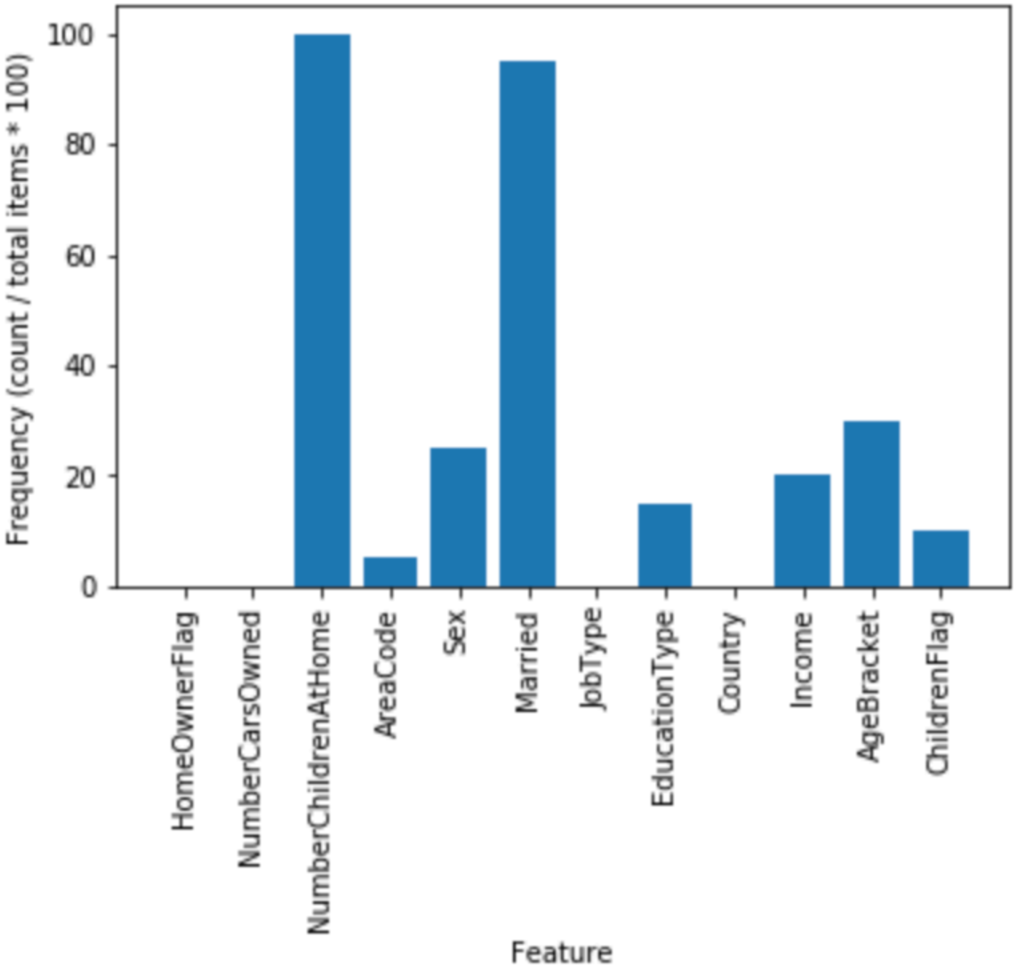


Figure 22

Feature Frequency at Size Four

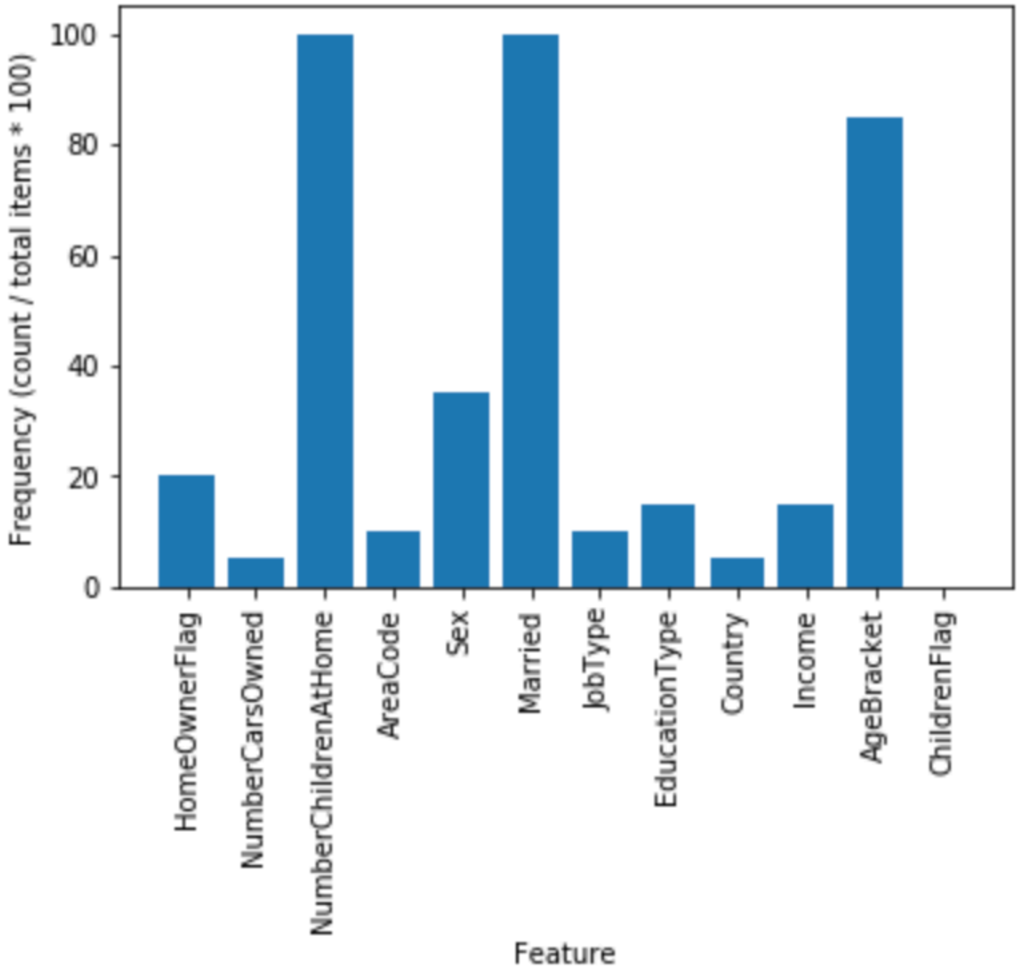


Figure 23

Feature Frequency at Size Five

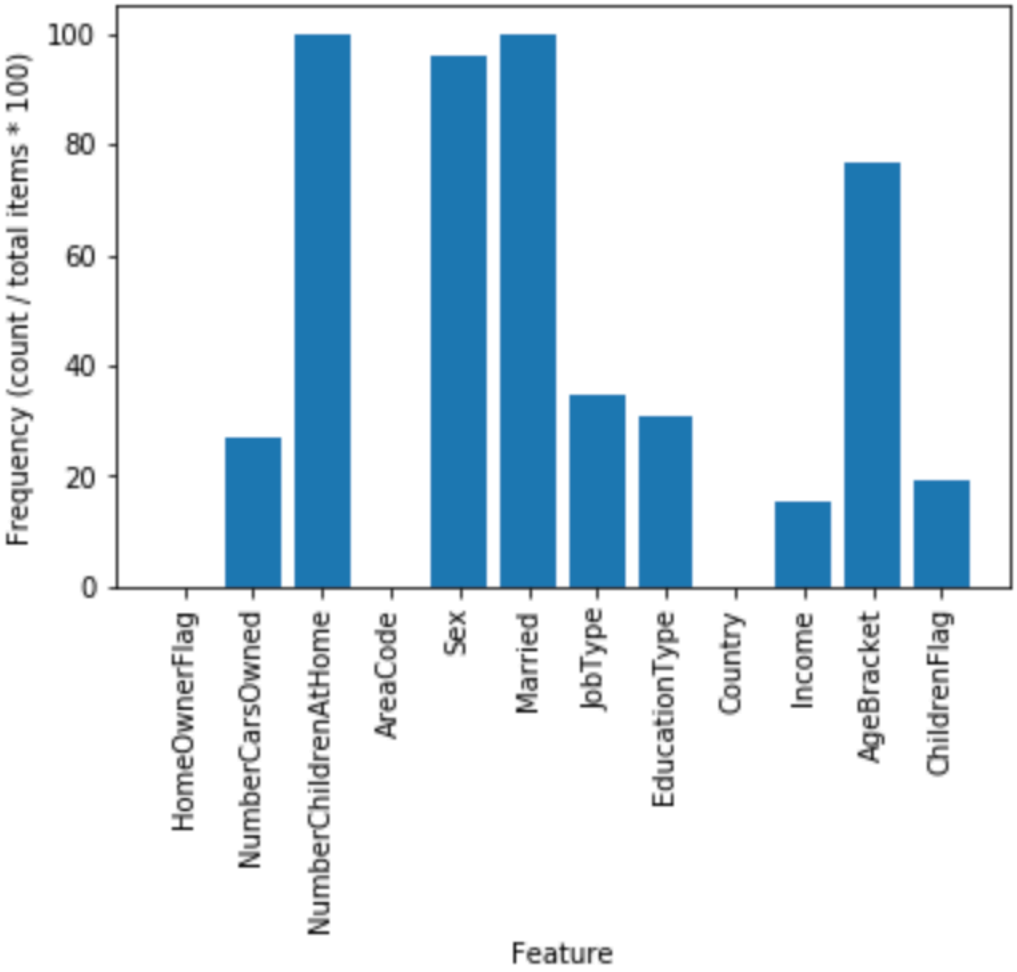


Figure 24

Feature Frequency at Size Six

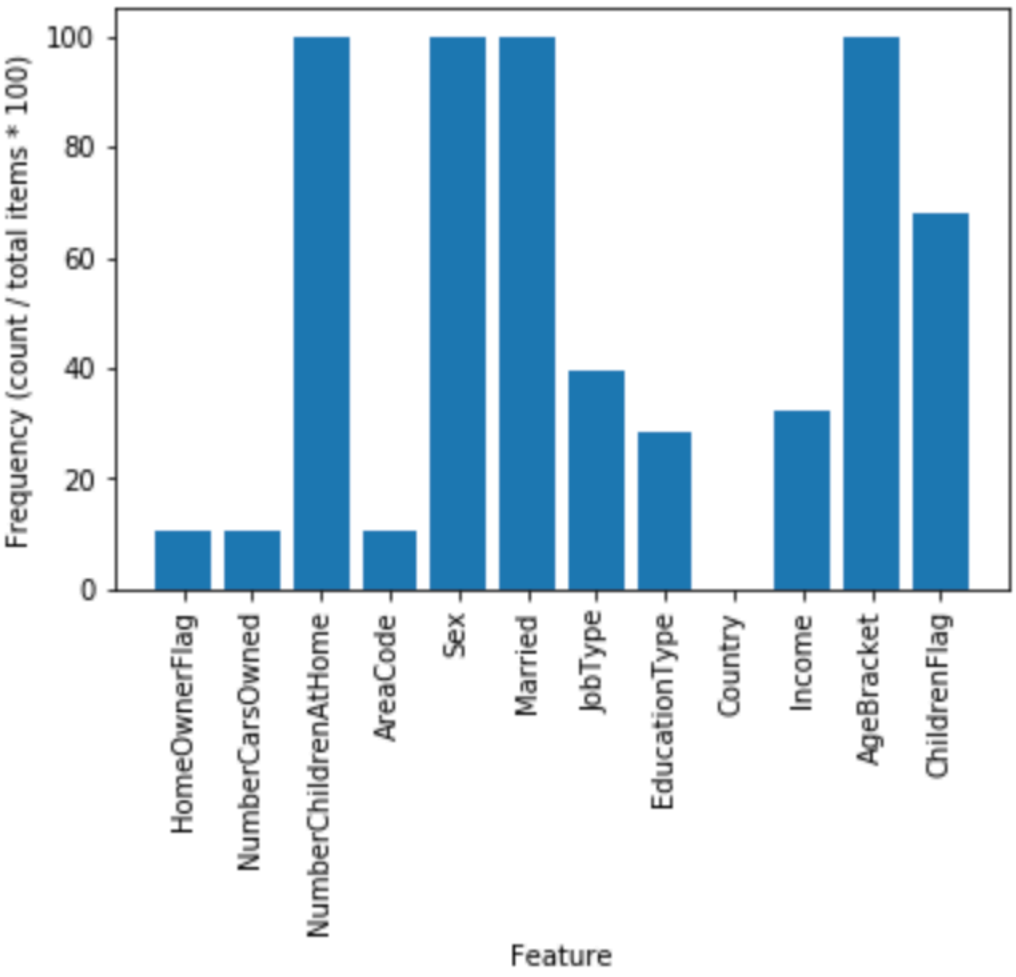


Figure 25

Feature Frequency at Size Seven

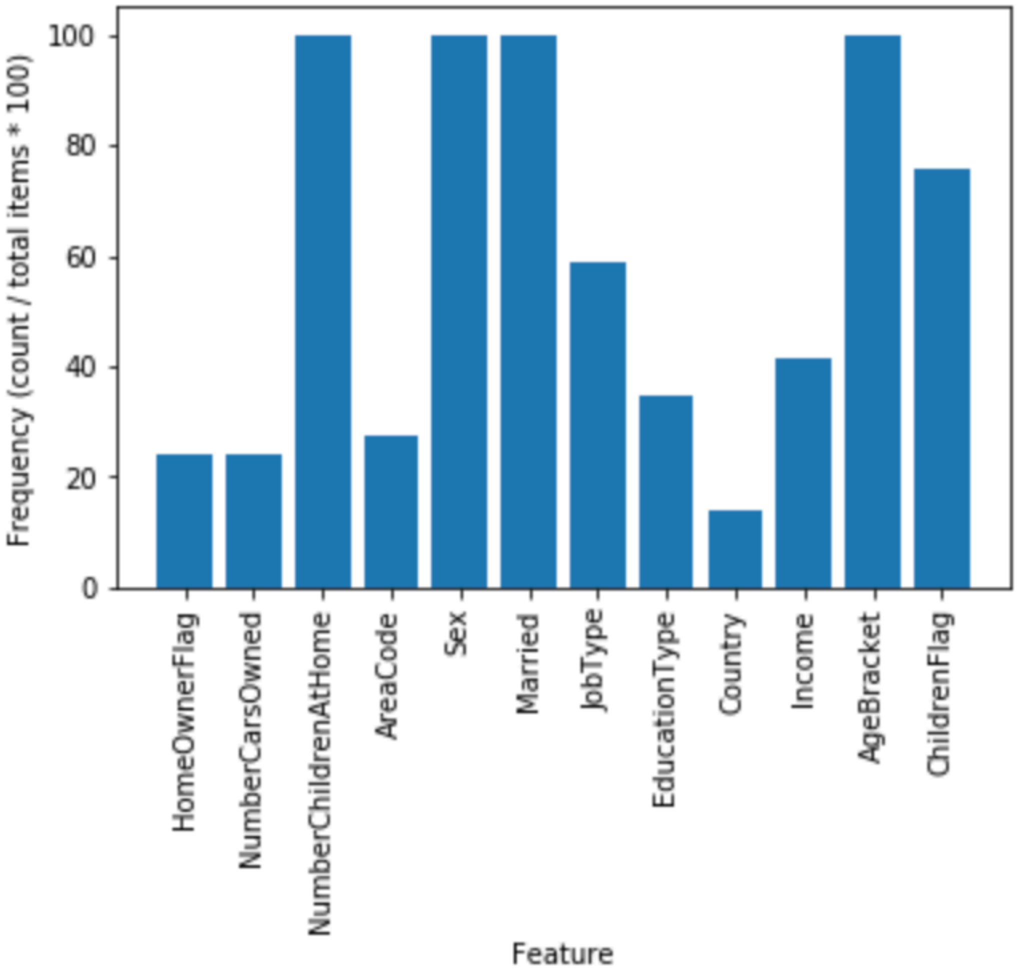


Figure 26

Feature Frequency at Size Eight

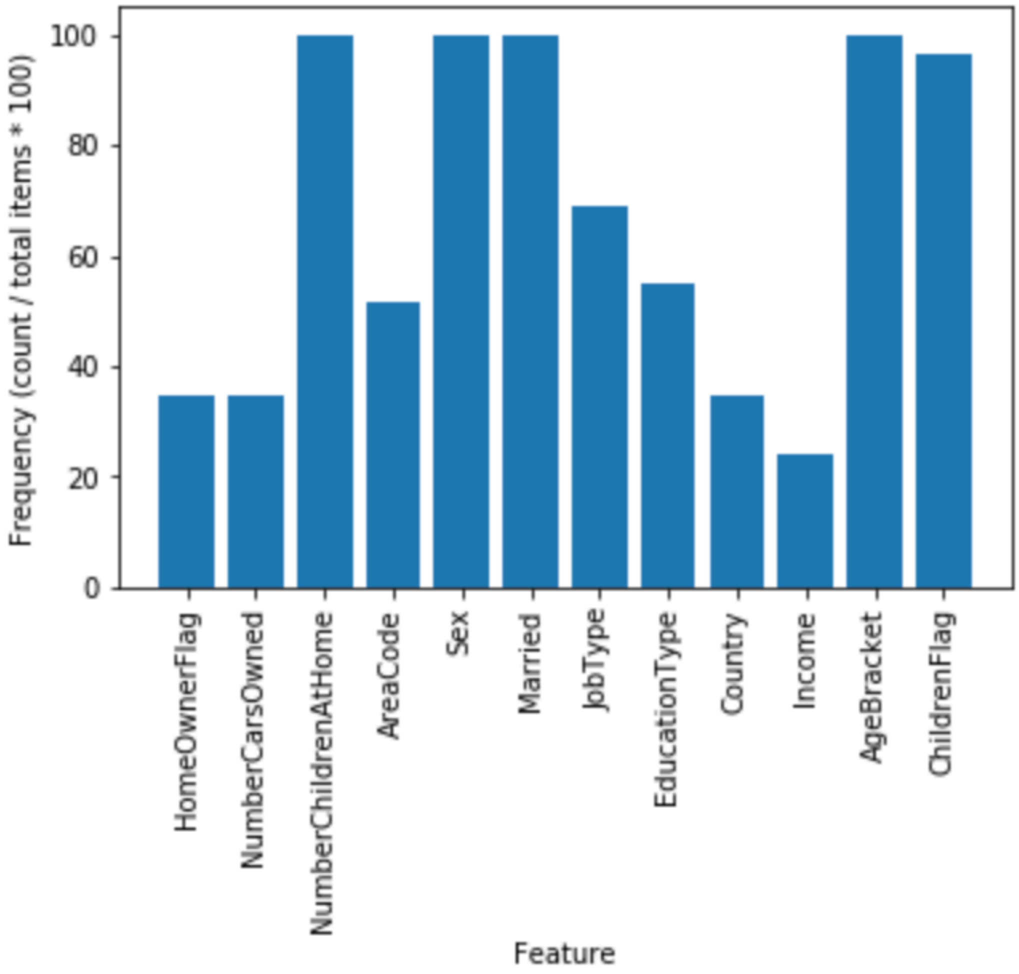


Figure 27

Feature Frequency at Size Nine

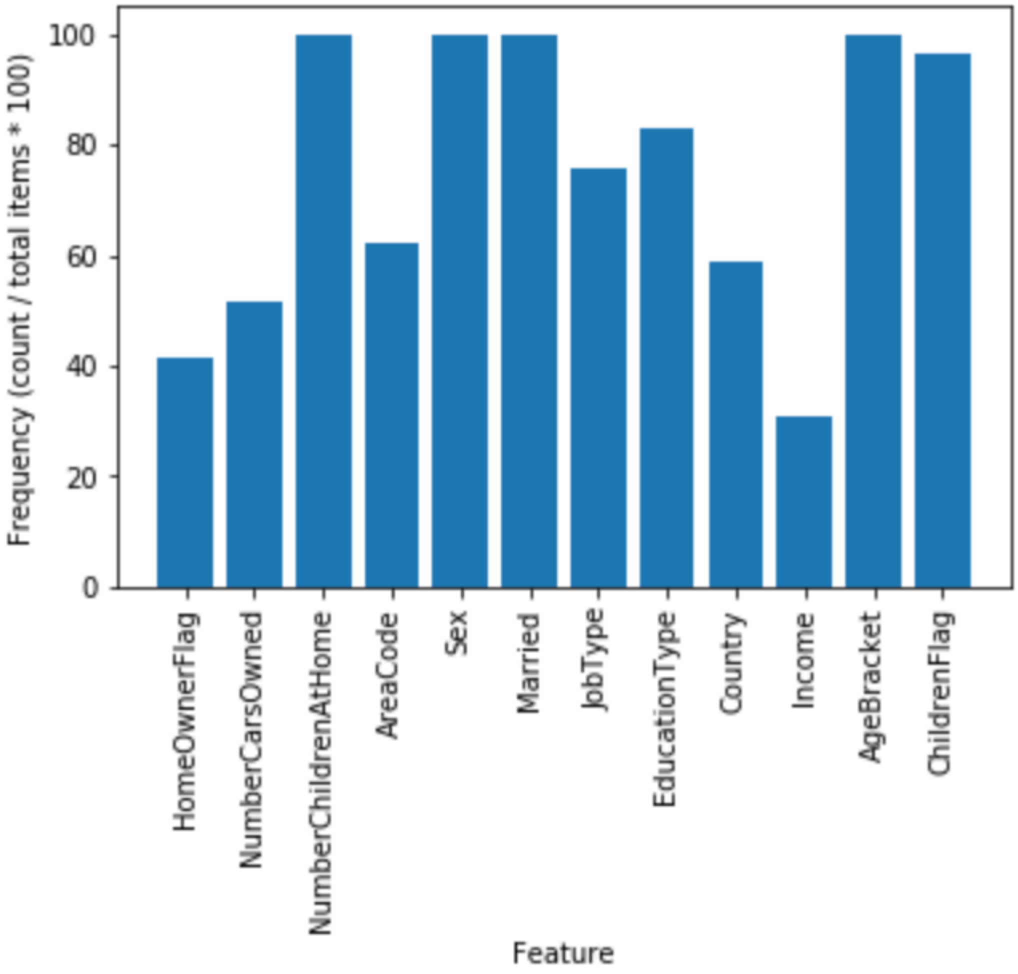


Figure 28

Feature Frequency at Size Ten

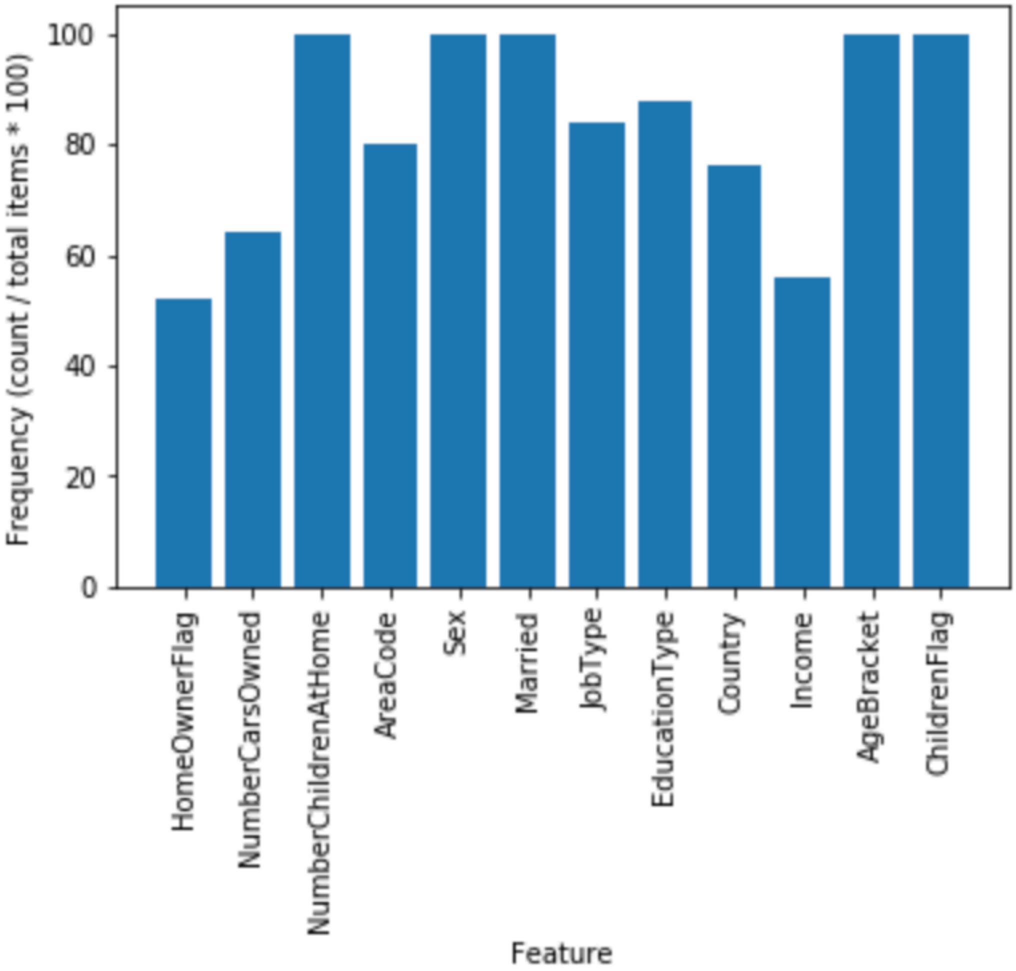


Figure 29

Feature Frequency at Size Eleven

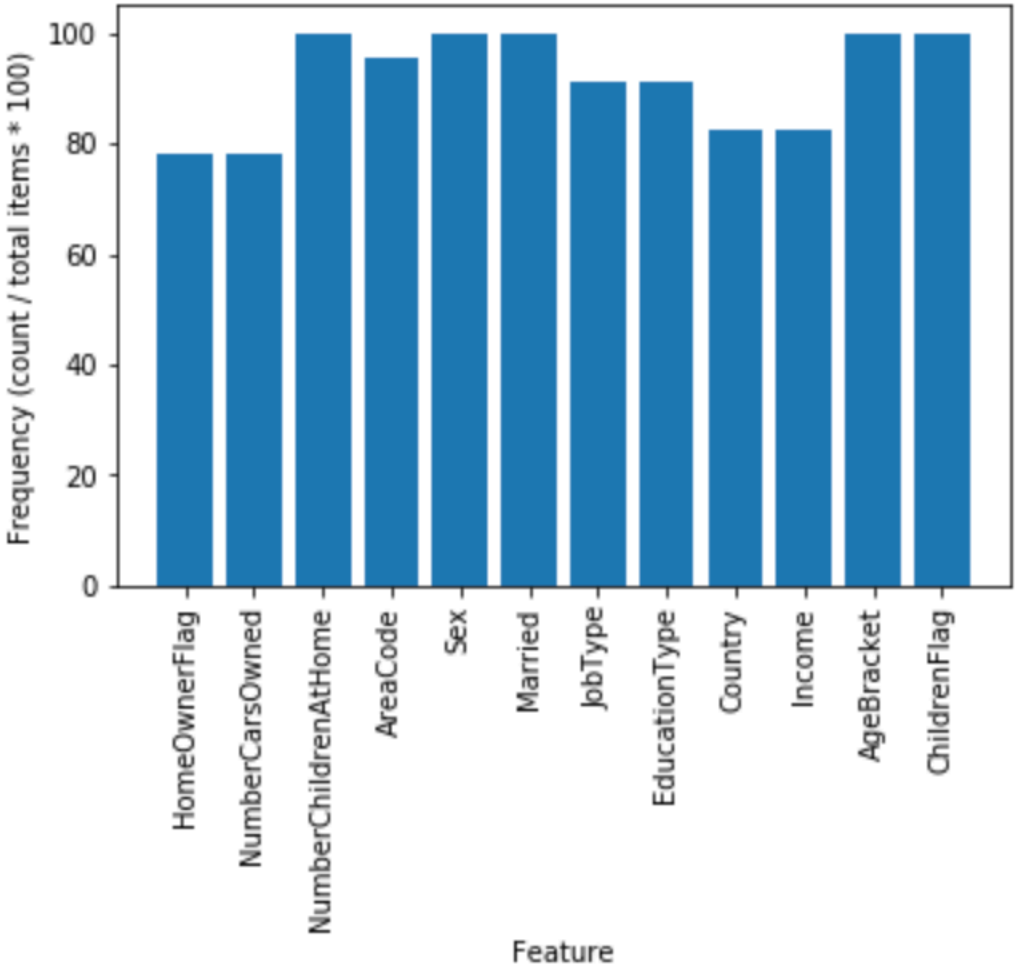
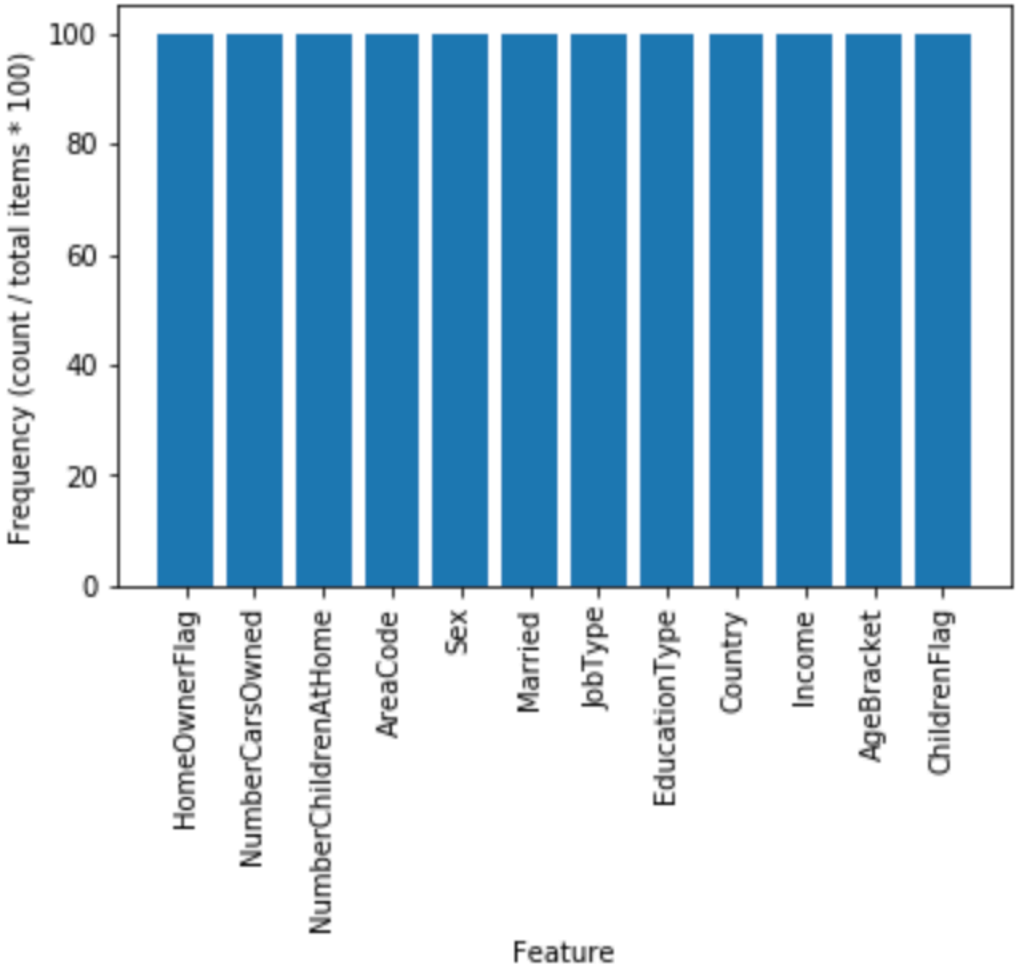


Figure 30

Feature Frequency at Size Twelve


Frequency (count / total items * 100)
HomeOwnerFlag
NumberCarsOwned
NumberChildrenAtHome
AreaCode
Sex
Married
JobType
EducationType
Country
Income
AgeBracket
ChildrenFlag
Feature