DSC 478 Final Report

In-Vehicle Coupon Recommendation

Adil Shedbalkar, Ronaldlee Ejalu, Tejas Desai

Executive Summary

Link to Dataset: https://www.kaggle.com/mathurinache/invehicle-coupon-recommendation

Link to Presentation Video: https://www.youtube.com/watch?v=X9pnuZpbwJU

This dataset comprised of variables containing weather/time, social/economic values of an individual such as age, gender, occupation, income, etc., as well as other factors such as their proximity to places such as coffee houses, bars, and restaurants.

The purpose of our analysis was to create classification algorithms which can effectively predict whether a driver will accept a coupon or not. Our goal was to find which features play the largest role in determining the target variable, whether the driver accepts the coupon, while also ensuring that the model is interpretable, not too complex, and can be applied in the future to unseen instances. Another thing we wanted to discover was if there was any formation of interesting groups/clusters that may arise in this dataset.

Presenting coupons to a customer can be a very efficient and cheap way of attracting them to a business or service. Coupons are not only less intrusive, and cheaper than other methods of advertising such as commercials or billboards, but they also contain further and more personal incentives for consumers to draw them in. The acceptance of a coupon is likely to increase traffic to a business and is bound to drive sales.

Both random forests and PCA were used as a supervised and unsupervised feature selection techniques, respectively. Using the reduced datasets, models including random forests, logistic regression, decision trees, and the stochastic gradient descent (SGD) classifier were built and compared. Overall, PCA led to much lower accuracy scores and did not lead to interpretable results as compared to the random forests feature selection.

Model, KPrototype and KMedoid were used as an unsupervised algorithm to extract some structure from the data set. We used these algorithms because our data set comprised of only categorical variables.

We concluded that there are a few main factors that businesses should consider when determining if coupons are a worthwhile expenditure to attract customers. We found that certain businesses, specifically Coffee Houses, Bars, Takeaway, and places that cost under $20. These are all places that people may frequently visit often and may not be very costly compared to others. Time was also very important in determining if a driver would accept a coupon, and we found that people would be more accepting of coupons when they are not rushing to be somewhere, whether that be work in the morning or back home late at night. The clustering results were inconclusive as to specific groups of customers being formed. We suspect that there might be some bias in how the data was collected

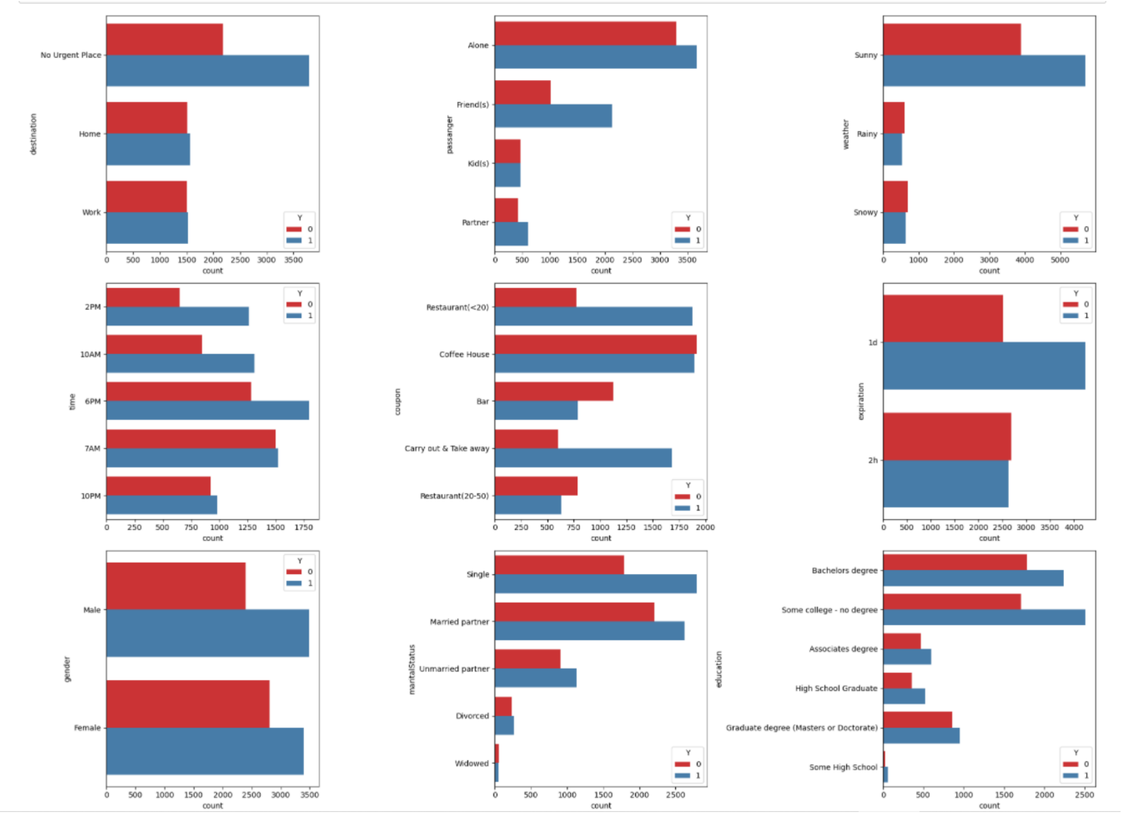
Contributions:

Adil: Data Preprocessing, Random Forest Feature Selection, Classification via Random Forest, Decision Tree, Logistic Regression, Stochastic Gradient Classifier, model performance/comparison, PCA

Tejas: Exploratory Data Analysis (EDA), Classification Random Forest, Decision Tree, Random Forest and Decision Tree Max Depth limit validation charts, Feature Selection charts, PCA and ROC Curves for RF Reduced Dataset Models.

Ronaldlee: Data PreProcessing and exploratory data analysis, KMode, KPrototype and KMedoid Clustering.

Data Exploration



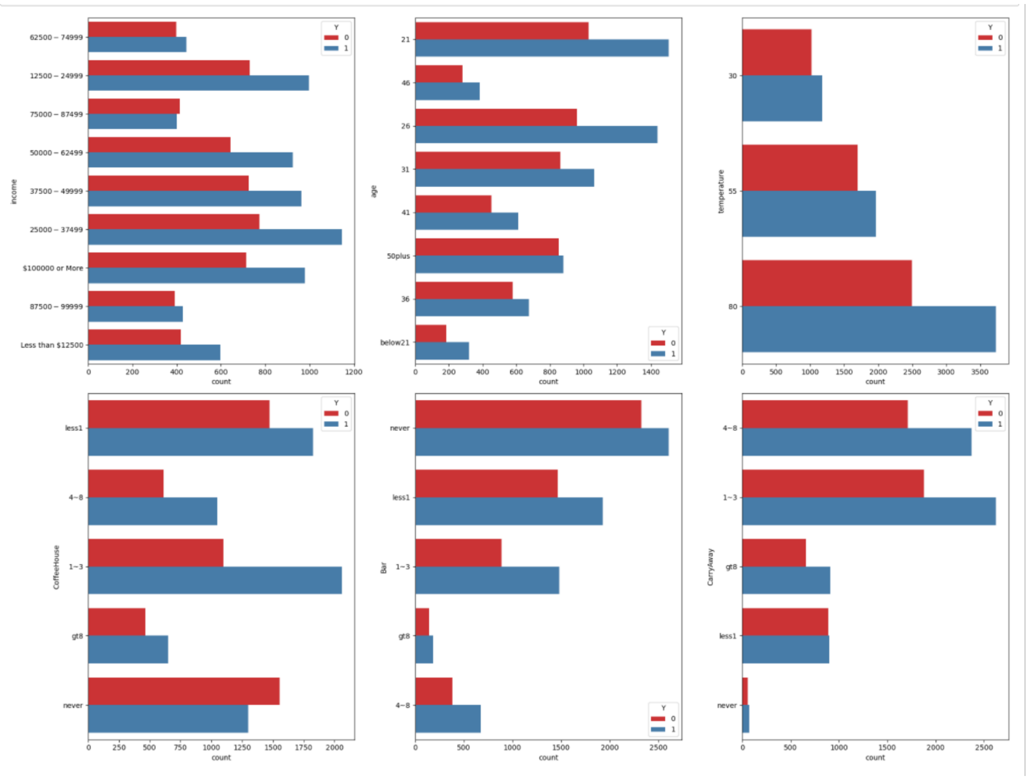


Figure A: Histograms of the distribution of values for most of the variables in the dataset

From Figure A and the percentage value found in the notebook, we can say that when the weather is "Sunny" coupon has more chance of being accepted compared to "Rainy" and "snowy" weather. We can also see that in 80-degree temperature coupon has more chance of being accepted compared to the lower 55- or 30-degree temperatures. As for timing, early in the day or late at night times have lower chances for coupons to be accepted. At these times, people may be rushing in the morning to go to work or at night going back home. When people don't have to be somewhere at a specific time or aren't rushing, they tend to accept the coupon more often. Expiry time has a great influence on whether coupons will be accepted or not, as we can see from the chart an expiry date of 1 day has a lot higher chance of being accepted than in 2 hours. The younger age demographic will generally accept the coupon more often. Single people will accept the coupon more often than other marital statuses. Generally, people in a lower income bracket seem to have a higher chance of accepting the coupon, compared to middle income. People in the highest income bracket are also more likely to accept coupons at a higher rate. People also seem to accept coupons more often for places that are less expensive and are therefore more likely to go to them often. Based on the exploratory data analysis, we can say that the following are some of the variables that will be of interest in evaluating whether coupons will get accepted: weather, destination, temperature, time, type of coupon, and income.

Data Pre-processing and Preparation

The dataset initially contained 12,684 records, with 26 columns, including the target, Y (whether the driver accepts a coupon or not), however, one variable, car, was dropped, as it was NA in 12,576 rows.

Furthermore, variables such as Bar, CoffeeHouse, CarryAway, RestaurantLessThan20, and Restaurant20to50 had missing values in 100-200 rows. We chose to drop these rows because of the negligible impact on number of rows, and the fact that most often occurring values for these variables were not clear cut (multiple values had nearly the same count). We also discovered that the variable toCoupon_GEQmin had the same value for every single row, so it was unnecessary to include in further analysis, and dropped.

Next, there were a few variables that were categorical in nature, but had some form of order, ie. they were ordinal variables. Ordinal variables in the dataset included time, age, income, Restaurant20to50, RestaurantLessThan20, CarryAway, CoffeeHouse, Bar, education, and temperature. These were transformed into numeric values based on their order. For example, 7AM was the earliest time so it was assigned 0, while 10AM was the next chronologic time stamp so it was assigned 1.

Marital Status was condensed to 3 values, rather than 6. Unmarried Partner, Divorced, and Widowed were now replaced with just as Unmarried, as this grouping made sense.

Doing both operations was necessary not only in terms of logical sense, but it also meant that not as many dummy variables would be needed for analysis. For example, instead of needing 5 different dummy variables for time, only one column was needed after the transformation. The remaining categorical variables that were not ordinal were transformed into dummy variables.

After these preprocessing steps, we were left with a dataset of 12,079 rows and 63 columns. An 80/20 train/test split was performed for the following classification tasks.

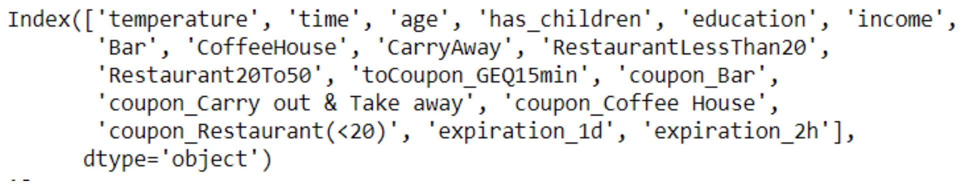
Feature Selection Methods

For classification, the following classifiers were used: Random Forests, Logistic Regression, Decision Trees, and Stochastic Gradient Descent. However, since the dataset was very large in terms of dimensionality, two methods of supervised and unsupervised feature selection were attempted to reduce dimensionality.

The first method was supervised feature selection, through random forests. Random forests feature selection through the SelectFromModel function from sklearn.feature_selection library with 10-fold cross validation was used to determine the most important features in determining our target variable, whether the person accepts the coupon. The way this feature selection method works is it essentially assigns each feature a value of importance in predicting the target variable. Features that score above the mean importance are preserved, while the others are not deemed valuable in predicting the target. This threshold can be tuned, but for our purposes this parameter will be left as is.

The method summary can be found here: https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f

This feature selection method was able to deem 18 features as greater than mean importance, and included variables such as temperature, time, age, Bar, RestaurantLessThan20 and Restaurant20to50 (times person went to restaurant with given average expenses), etc. The full list of variables can be found below (Figure B).



```
Index(['temperature', 'time', 'age', 'has_children', 'education', 'income',
       'Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20',
       'Restaurant20To50', 'toCoupon_GEQ15min', 'coupon_Bar',
       'coupon_Carry out & Take away', 'coupon_Coffee House',
       'coupon_Restaurant(<20)', 'expiration_1d', 'expiration_2h'],
      dtype='object')
```

Figure B: List of selected features that scored above mean importance.

On the other hand, unsupervised feature selection was done via PCA. Since our goal for PCA was dimensionality reduction, we decided that the best method for determining the number of components would be choosing the number of components that give at approximately an 85% cumulative explained variance. 95% of variance would require far more, at 22 components. Two versions of the scree plot are provided below as reference, but choosing the number based on the knee (Figure D) would not be appropriate for our purposes, as it would have a much smaller explained variance ratio that would have a drastic negative effect on model building accuracy, especially on the test set. 12 components were used in the following models, as they gave us ~85% explained variance (Figure C).

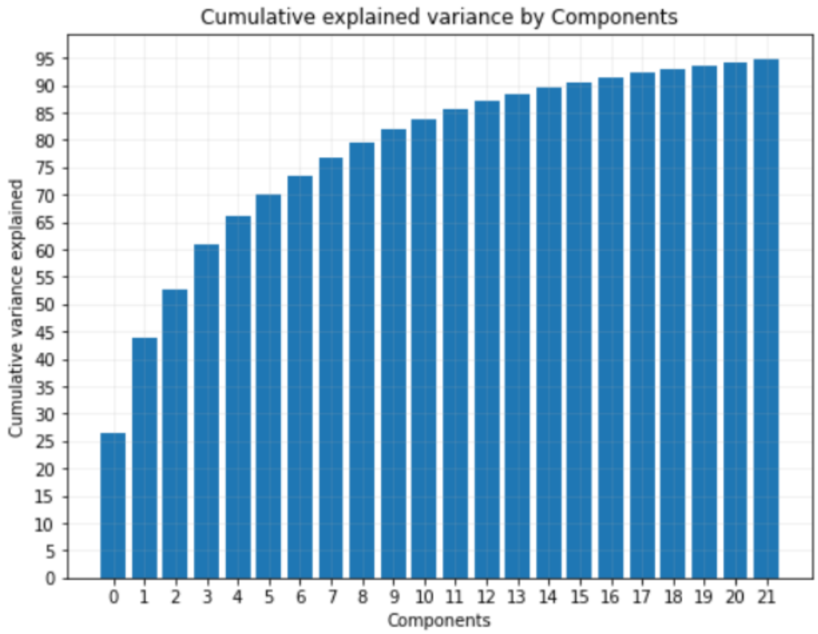


Figure C: Cumulative explained variance per number of components. Approximately 85% of the variance is explained by 12 components. This chart is a condensed version of the chart found in the notebook.

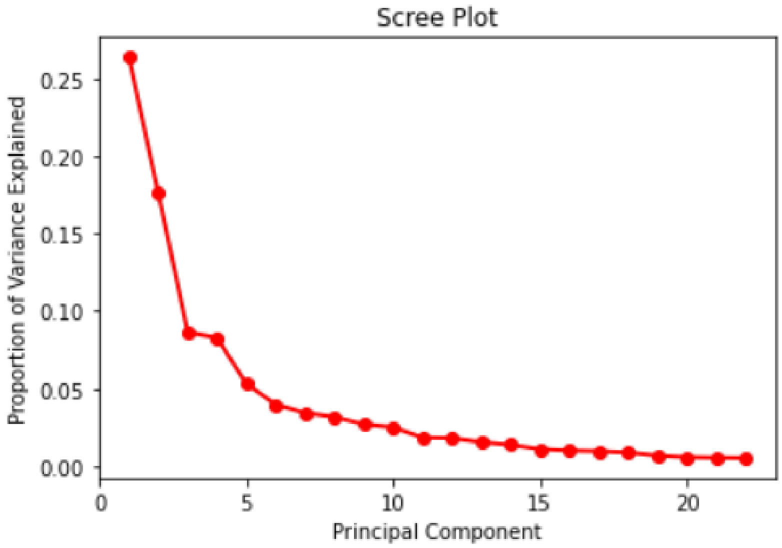


Figure D: Another look at the scree-plot. The knee will not be useful to determine the number of components for our purposes.

From here, random forests, logistic regression, decision trees, and SGD classifiers were built for both versions of the reduced data to see how the results of the two methods vary and which can give us the best possible model. 10-fold cross-validated grid search was used on all models to determine the most optimal parameters for each. In the case of random forests and decision trees, a model was first built without a cap on max depth. Next, another was made with a cap on max depth while preserving the values of the other parameters found through grid search. This was done in order to see if building a smaller tree would create similar results and be easier to explain and visualize.

The grid search parameters for each classifier are as follows:

Random Forests: min_samples_split, max_depth, n_estimators

Logistic Regression: penalty, C, class_weight

Decision Tree: min_samples_split, max_depth, criterion

SGD Classifier: penalty, alpha, l1_ratio

Results- Random Forest Reduced Dataset

The first classifier that was used was random forests. Using grid search, the most optimal values found were min_samples_split: 5, max_depth: None, and n_estimators: 90. This produced a model with very high training accuracy score at 95.4%, but the testing accuracy was much lower, at 73.9%. As a result, the model was likely overfitting. Not specifying the maximum depth of the tree means that it can grow as large as possible to try and fit the data as closely as possible. Another model will be built with a limit placed on maximum depth to try and alleviate this. The model had the most trouble with recall of the 0 case. Furthermore, the balanced testing accuracy was commendable, at 72.6%.

The next random forests model kept the optimal values for min_samples_split: 5, and n_estimators: 90, but used a training vs. testing accuracy chart to determine the optimal max_depth without including 'None' as a possible value to ensure that the tree is not growing extremely large. Figure E below shows the training and testing accuracies for each value of max_depth. We can see from the chart that we reach optimum point in bias-variance tradeoff at about a max_depth of 9. After this we do not see much improvement in test accuracy but the training accuracy continues to rise, indicating we are starting to overfit the training dataset. For this second decision tree, the parameters will be max_depth: 9, min_samples_split: 5, n_estimators: 90.

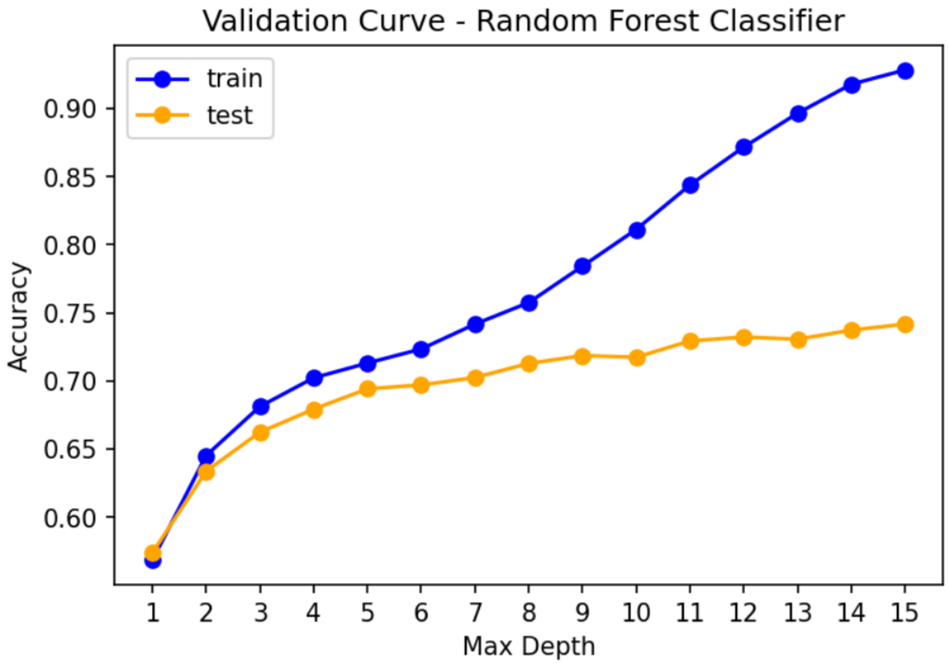


Figure E: Training and testing accuracies for each value of max_depth. Note the increase in both as the model gets more and more complex (large).

Since this model is not overfitting, it is also beneficial to see the feature importance of each of the variables. This can be seen below in Figure F.

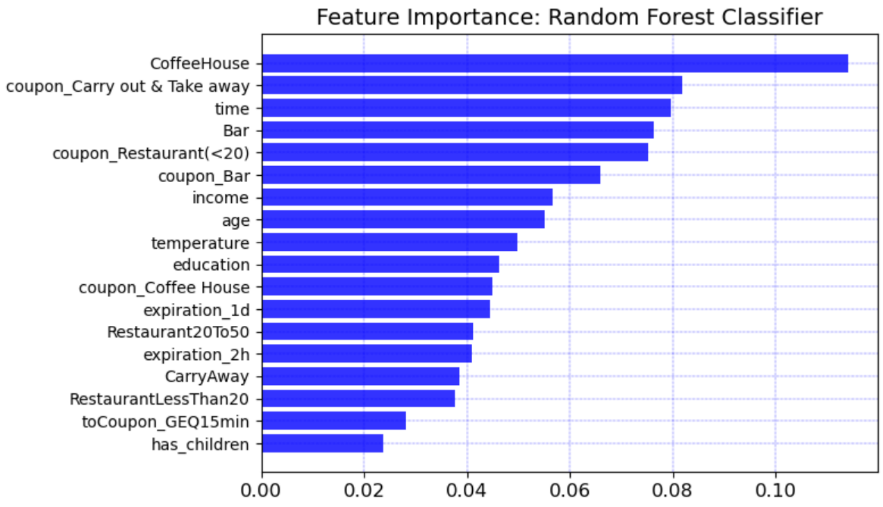
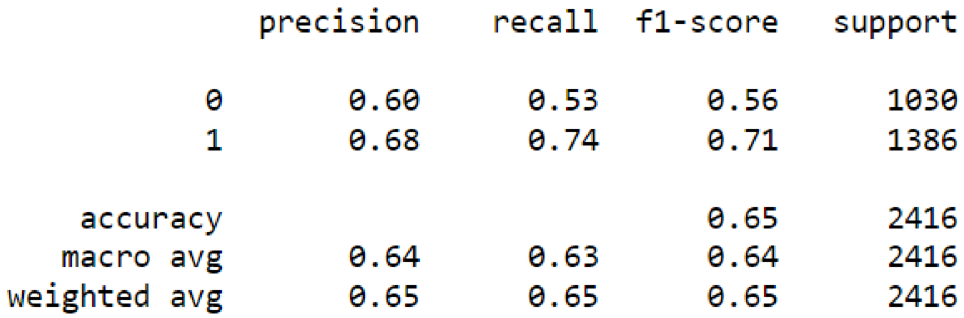


Figure F: We see that the most important features seem to be CoffeeHouse, coupon_Carrry out & Take away and time of the day.

The next classifier that was used was logistic regression. Optimal parameters via grid search were C: 0.1, class_weight: None, and penalty: L2. Although logistic regression did not score very highly in terms of accuracy scores, unlike random forests, it did not overfit the data. This can be seen by the small difference in training (67.1%), and testing (65.0%) accuracies. It also had a balanced accuracy score of 63.4%, which can be seen when numerically when viewing the confusion matrix. The classification report (Figure G) gives us a more in-depth view and reveals that the worst aspect of the model was recall and precision of class 0 (person denied offer of coupon).



```
              precision    recall  f1-score   support

           0       0.60      0.53      0.56      1030
           1       0.68      0.74      0.71      1386

    accuracy                           0.65      2416
   macro avg       0.64      0.63      0.64      2416
weighted avg       0.65      0.65      0.65      2416
```

Figure G: Classification report reveals the model has the most trouble with recall of the 0 case.

The third classifier that was built was decision trees. This is like random forests in the idea that random forest is an ensemble method that considers many decision trees but is worthwhile to test either way. The best parameters based on grid search were found to be criterion: entropy, max_depth: None, and min_samples_split: 100. The decision tree classifier performed very well in terms of not overfitting the training data, while also retaining a good accuracy score for both the training and test sets. The training accuracy was 74.6%, the testing was 70.7%, and the balanced testing was 69.2%. Just as before, the model had most trouble predicting when the actual label was 0, as it guessed the label was 1 instead 41% of the time.

In order to make visualization easier, another decision tree was built with a cap on maximum depth while preserving the other parameter values. A similar curve to Figure D was produced and we concluded that after around a max_depth of 6, we don't see much increase in test accuracy. For the understanding/visualizing tree better, we will prune the tree with a max_depth of 6. This tree can be seen in the notebook. This model had similar training, and testing accuracies (70.6%, and 68.4% respectively). The feature importance for this model is shown below, in Figure H.

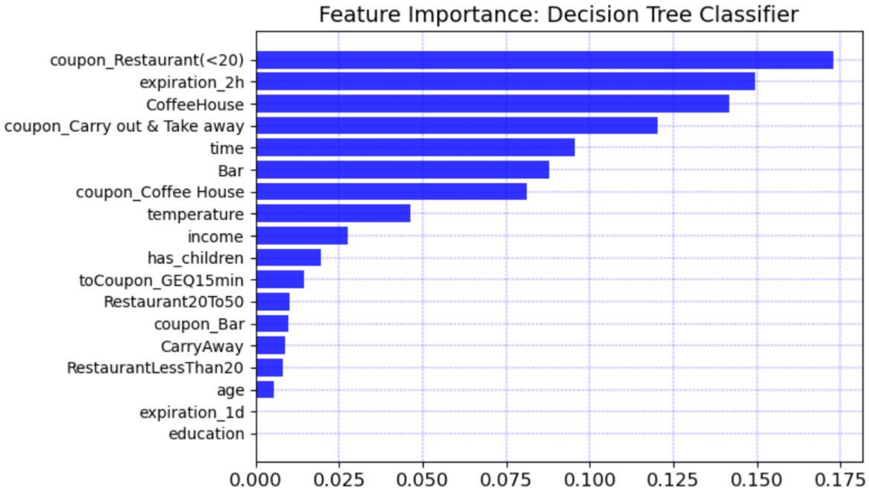


Figure H: coupon_Restaurant(<20) is the most significant feature, followed by expiration_2h and CoffeeHouse

Finally, the last classifier to be tested was stochastic gradient descent (SGD). Grid search found the optimal parameters to be alpha: 0.01, l1-ratio: 0.3, and penalty: elasticnet. Overall, the classifier performed similar to logistic regression in terms of accuracy scores. Similarly, based on the small difference in training (66.6%), and testing (64.4%) accuracies, the model is not overfitting the training data. It also has trouble predicting the 0 class (denial of coupon), just as the models before, as seen by the lower recall/precision scores of this class. It had a balanced accuracy of 63.7% that reflects this.

Below, in Figure I, the training, testing, and balanced testing accuracies for each of the models for the random forests reduced dataset are plotted. We can see that the first random forests model, without a cap on maximum depth yields a model that is extremely overfit, as the training accuracy is much higher than the testing accuracies. The second random forests model takes care of this, as a cap was placed on maximum depth of the tree. This yields a much more parsimonious model that can be explained and visualized easily, while at the same time retaining almost the same exact testing and balanced testing accuracies. For Decision Trees, the first model did not seem to be overfitting, so that was used here. It performs very similarly to random forests. Logistic regression and SGD Classifier are the worst of the models, with slightly lower accuracy scores. However, they are by no means bad, just slightly worse than the others.

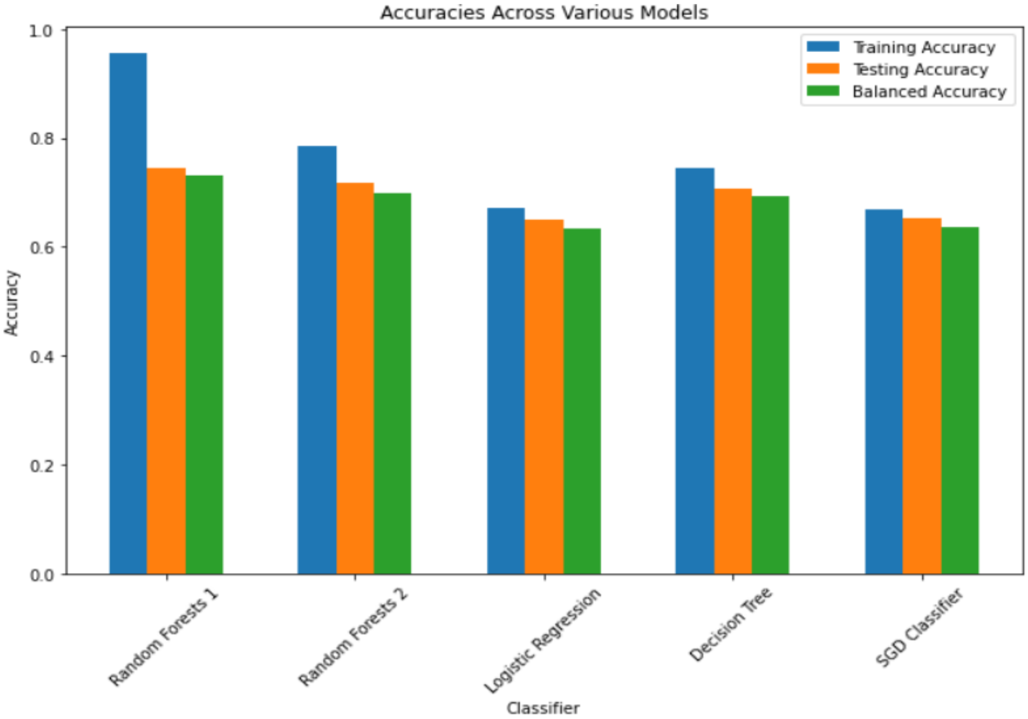
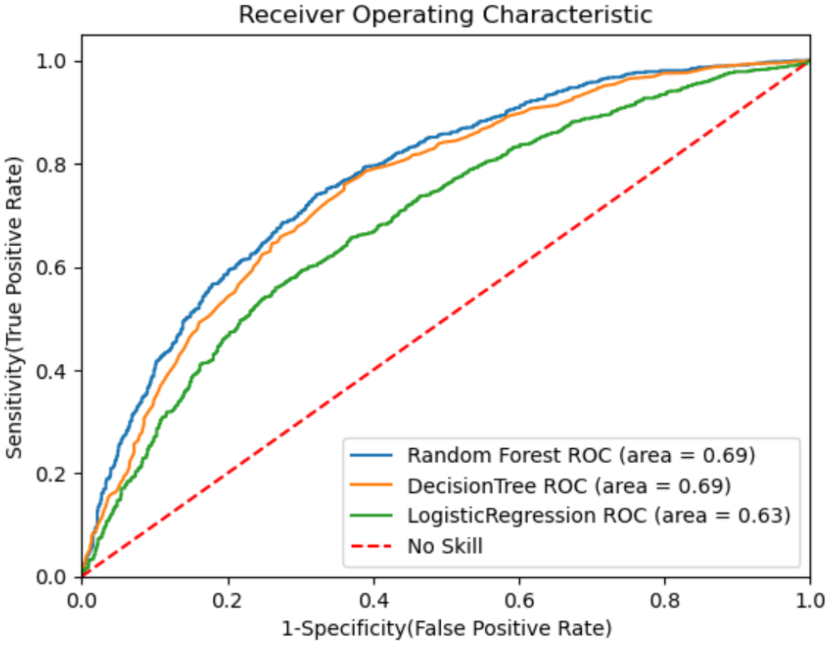


Figure I: Accuracies across all models using the random forests reduced dataset



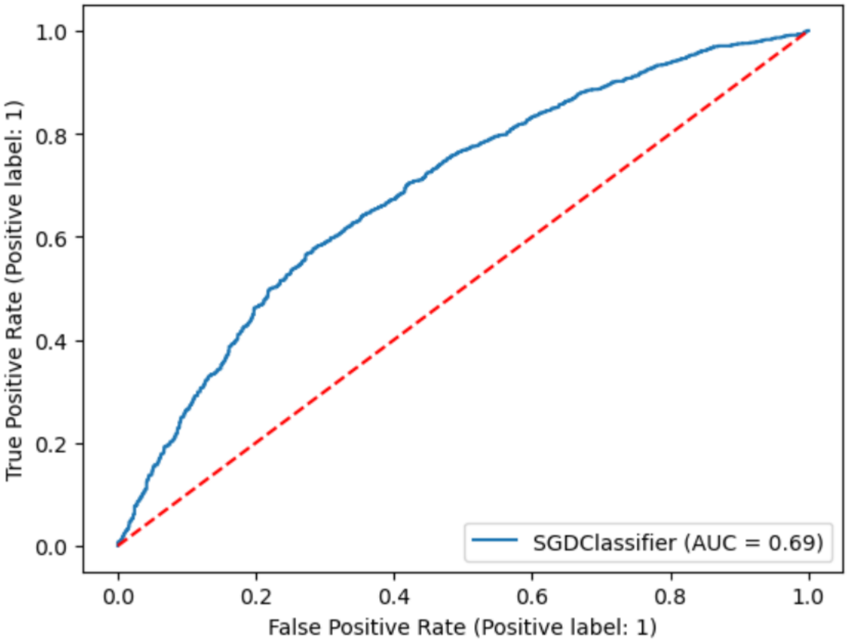


Figure J: ROC Curves and AUC values for the classifiers on the Random Forests Reduced Dataset

Above, in figure J, we can see the ROC and AUC values for each of the models. AUC values of 1 are indicate a perfect fit, with 100% true positive rate and 0% false positive rate. However, these are only typically possible in theory, and values that are between or above 70-80% are still impressive and indicate that the model is at least doing better than randomly guessing and able to classify most instances accurately. In Figure J, we see similar values for area under the curve for Random Forest, Decision Tree, and SGD Classifier at 0.69, while we get lower values with Logistic Regression at 0.63.

Results- PCA Reduced Dataset

Random forests classification was once again done with and without a cap on max_depth, using the same method above. The grid search optimal parameters for the unrestricted max_depth parameter iteration were min_samples_split: 10, n_estimators: 90, max_depth: None. Just as seen with the supervised feature selection random forests model from before, random forests on the PCA reduced dataset also greatly overfit the training data (98.4% accuracy) while obtaining much lower accuracy on the test data (65.5%). The testing balanced accuracy was also low, at 64.7%. Since the model is overfitting likely due to the uncapped maximum depth, another random forest was built with a cap on this value.

For the max_depth capped random forest chose a max_depth of 12, based on the generated training and testing accuracy scores in Figure K below. We can see from this figure that we reach an optimum point in bias-variance tradeoff at about max_depth of 5, as after this we do not see much improvement in testing accuracy, but training accuracy keeps going up, so we start moving towards the overfitting zone. Accuracy scores are significantly lower than the random forest reduced dataset, even after taking

12 components, which means PCA is not helping much with this dataset. Using a max_depth of 5, the training, testing, and balanced accuracy scores were 66.3%, 62.3%, and 58.3%, respectively. These scores are unimpressive and quite low, further supporting the idea that PCA might not be the best method feature selection here, and our initial method of random forests feature selection is likely more feasible.

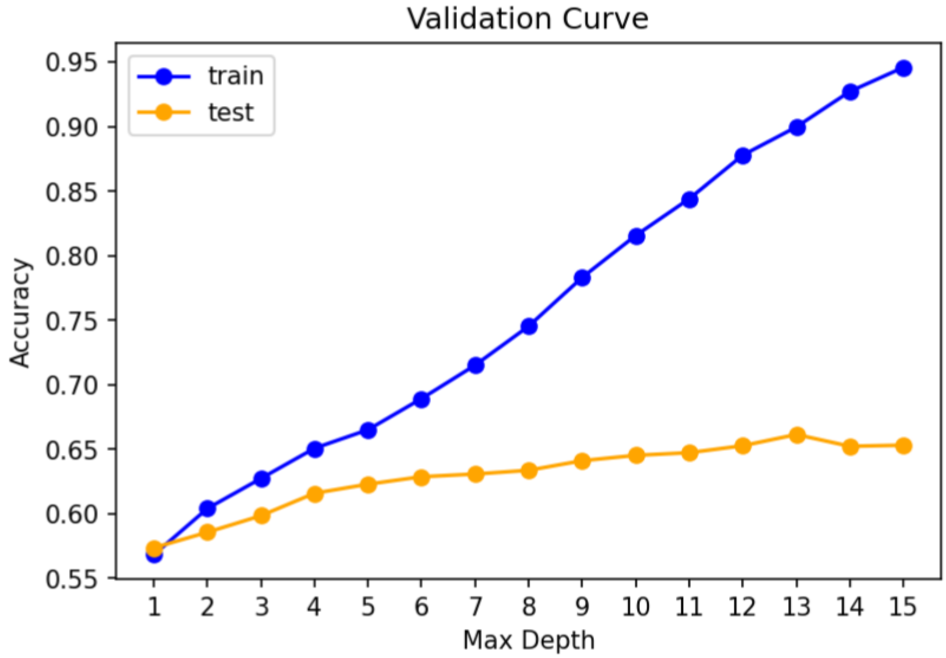


Figure K: Training and testing accuracies at different values of max_depth

Logistic Regression parameters via grid search were C: 1, class_weight: None, and penalty: l1. Although logistic regression did not overfit the training data, the results were not very impressive. The training, testing and balanced testing accuracies (62.9%, 60.9%, 58.0% respectively) were all low. As before with all models, recall for the 0 class (0.38) was where the model ran into the most trouble, but it was by far the worst with logistic regression so far.

The optimal parameters for decision trees were criterion: gini, max_depth: 7, and min_samples_split: 7. Decision Trees on the PCA reduced dataset performed very similar (albeit marginally better) than logistic regression. The training, testing, and balanced accuracy scores (67.1%, 61.3%, and 58.6% respectively) were also underwhelming results. The recall was barely better than logistic regression, at 0.38.

The SGD classifier performed the worst of all other classifiers on the PCA reduced dataset. Scores were slightly lower than that of logistic regression. Its optimal parameters were alpha: 0.01, l1_ratio: 0.1, and penalty: l2. It had the lowest training, testing, and balanced accuracy scores (63.9%, 59.7%, and 56.6% respectively), as well as the worst recall of the 0 class, at 0.35.

An overview of each of the models' training, testing, and balanced testing accuracies is shown below in Figure L. As expected, the first random forests model was overfitting the training data. All other models, especially logistic regression and SGD had low training and testing accuracies. Based on these results, the PCA transformed data is not as good as the random forests reduced dataset due to lower training and testing accuracy scores across the board.

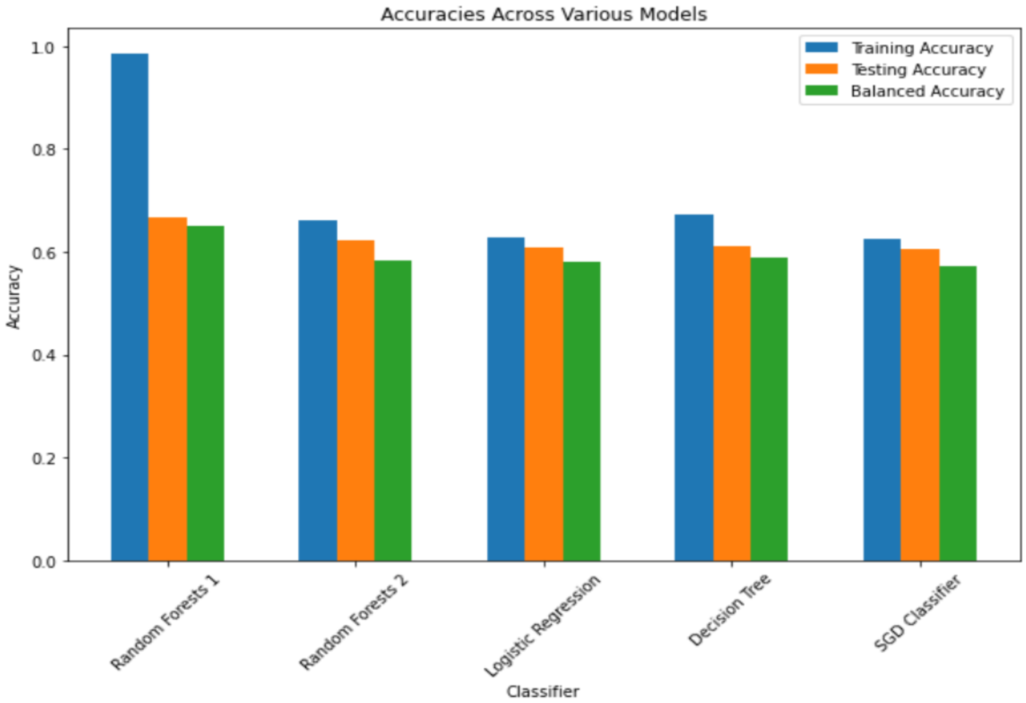


Figure L: Accuracy scores of models on PCA reduced dataset. All models have less than 70% accuracy.

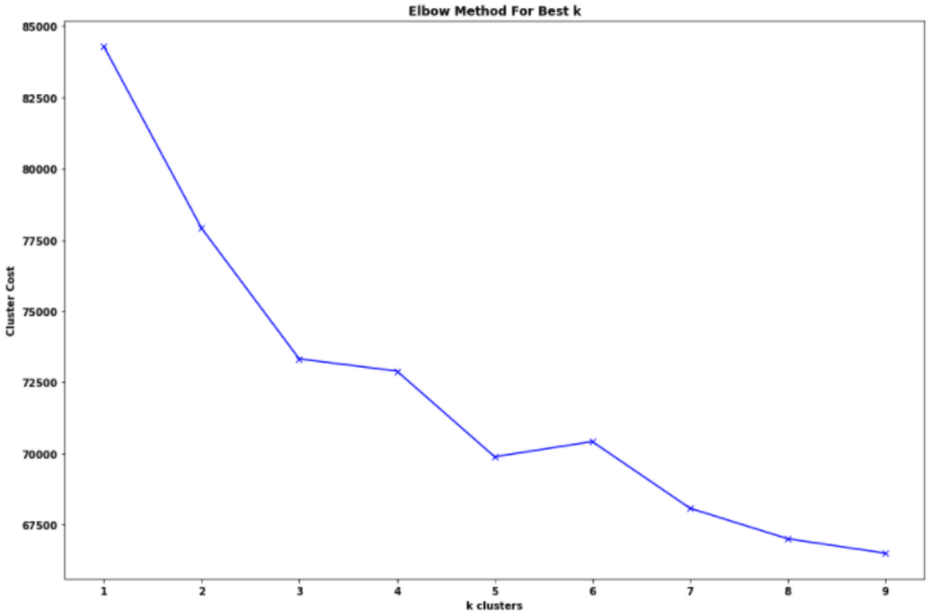
PCA did not seem to be as useful of a feature selection/dimensionality reduction technique as Random Forests feature selection for our dataset. For 85% of variance captured, we needed 12 components, but for 95 we would need around 22. 12 components are less than the number of features in the random forests reduced dataset, but the results of random forests feature selection were far more interpretable and higher in overall training and testing accuracy across the board for all models. 22 components would be more features than random forests and would have been even more difficult to interpret. At this point, it does not seem that PCA is helping us efficiently perform dimensionality reduction.

Results- Clustering

Before clustering, we thought of normalizing our data set but since we are dealing with categorical variables with transformed ordinal values, we decided not to standardize the data set because the values are not that big. We used several clustering algorithms, including KMode, KPrototype and KMedoids.

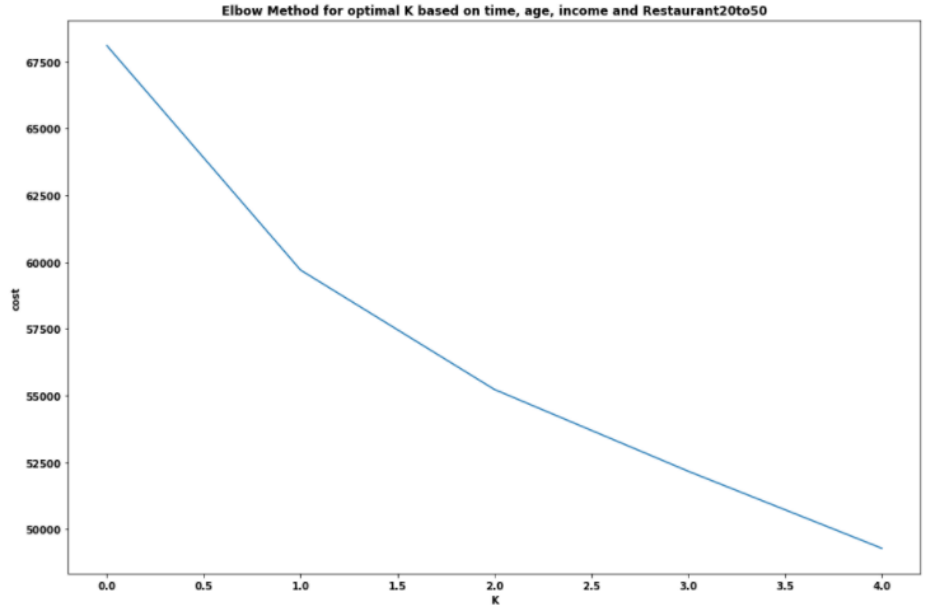
First, using Kmodes with Cao we used the elbow method to determine the optimal k where 3 and 4 looked the best ks so we derived clusters based on both k = 3 and 4.

When k = 3, the cluster sizes were 4807, 2321 and 2535 for cluster 0, 1, and 2. When k = 4, the cluster sizes were 3362, 2198, 2140 and 1720 for clusters 0, 1, 2, and 3. Both k = 3 and 4 had well defined clusters but with exceptionally low homogeneity and completeness scores of 0.0026 and 0.0017 respectively. Furthermore, looking at the values of optimal k below:



3 appeared to be the optimal value, however its completeness and homogeneity scores were low to consider any of the clusters derived.

Second, after noticing the kind of scores we were getting for both homogeneity and completeness, we decided to use the KPrototype algorithm-based time, age, income and Restaurant20To50. using the KPrototype algorithm we derived different clusters based on both k = 2, 3 and 6. When k = 2, the cluster sizes were 6064 and 3609 for clusters 0 and 1 respectively. When k = 3, the cluster sizes were 3504, 2648, and 2511 for clusters 0, 1, and 2. When k = 6, the cluster sizes were 1670, 1708, 1278, 2213, 1193 and 1601 for clusters 0, 1, 2, 3, 4, and 5. 0.005 was both the homogeneity and completeness scores for k = 2 and 6. 0.0005 and 0.0008 were the completeness and homogeneity scores respectively for k = 3. These scores did not make any difference from what we saw earlier on. The elbow method below:

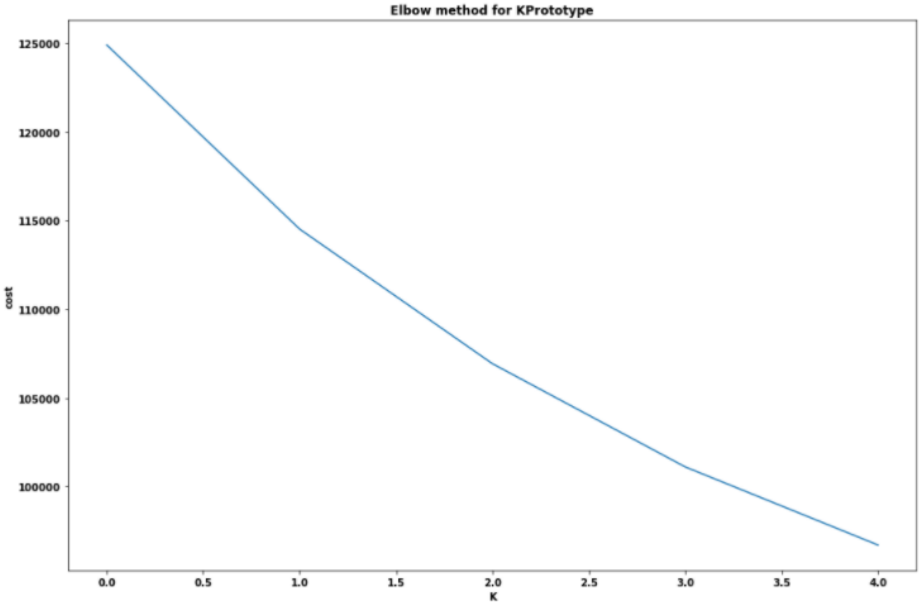


Showed that the best optimal k would be between 1 and 2, however k = 2 reported lower completeness and homogeneity scores.

Third, since the scores we were getting did not make too much of a difference, we decided to take the features of importance attributes, which we derived from random forests and created a subset of the data set for these features. Also, ran a correlation matrix to rule out any multicollinearity amongst the different features, then using KModes again we derived different clusters based on both k = 4 and 5. 3281, 2178, 2045 and 2159 were cluster sizes for clusters 0, 1, 2, and 3 respectively when k = 4 and 2004, 2610, 1565, 2238, 1246 were cluster sizes for cluster 0, 1, 2, 3, and 4 respectively. 0.005 and 0.01 were the completeness and homogeneity scores respectively when k = 4. 0.003 and 0.006 were the completeness and homogeneity scores respectively for k = 5. As previously seen, these scores are still low.

Fourth, using the KPrototype algorithm, which based on the education attribute where we derived different clusters with k = 3, and 2474, 3032, and 4157 were the cluster sizes for cluster 0, 1, and 2.

0.002 was the score for both completeness and homogeneity which is low. Furthermore, we derived a another KPrototype algorithm based on age and using k = 2, cluster sizes of 6311 and 3352 for respective clusters 0 and 1 were determined. Also, using the same attribute, age, we used k = 3 to derive cluster sizes of 3591, 3282, 2790 for clusters 0, 1, and 2 respectively. 0.001 was both the homogeneity and completeness score which is too low to accept the clusters derived when k = 2. Additionally, k = 3 had a completeness score of 0.001 and homogeneity score of 0.002 and both scores were low to consider the clusters derived. The values of k = 2 and 3 were determined using the elbow method as described in the chat below:



Finally, using KMedoids algorithm, we derived different clusters based on k = 2 and 3 where the completeness and homogeneity scores didn't make much difference from the values that we have seen before.

Conclusion

We found the best models to predict whether a consumer will accept a coupon to be either the second random forest (capped max depth) or the decision tree. Both models are fitted on the supervised random forests feature selection method. These models not only had the best training, testing, and balanced accuracy scores, but also had small differences in their training and testing accuracies to indicate a low probability of overfitting. They are also inherently models that are easy to explain and interpret rather than what may be considered more black box models such as SGD and logistic regression, which may not be interpretable to audiences or business executives that may need visualizations in order to understand. With decision trees and random forests, the results can be easily shown and explained to all audiences. They also reveal feature importances that essentially highlight which variables are having the largest impact, and in a business sense, what the business should focus their resources on.

Using KModes, KPrototype, and KMedoids on the categorical variables, we were unable to segment our data set into the groups of interest because the completeness and homogeneity scores were low for all the different Ks we tried. Since there are so many features even after deriving the attributes of interest, we still need to reduce our features and then run the different clustering algorithms previously used and this is something that we can investigate in the future, however this is likely to result in even smaller completeness and homogeneity scores. Since the results of this clustering method were inconclusive as to specific groups of consumers being formed, we suspect that there might be some bias in our data set and this is something that needs to be investigated. A bias may arise with the approach used for data collection.

When looking at the feature importances of both pruned random forest and decision tree classifiers, we can see quite a few similarities. The way feature importance works is that it calculates a value based on the decrease in node impurity weighted by the probability of reaching that node. The more a given feature decreases impurity, the higher its importance. The criterion for impurity in this scenario was entropy.

In both the models, we saw CoffeeHouse and Bar as being one of the most important features. This variable is defined by how many times the driver visits a coffeehouse or bar every month. It makes logical sense that those who visit these places more often may be more inclined to accept coupons since they have more opportunities and incentive to constantly use them. We saw this in our initial exploratory plot comparing the type of coupon against and the number of drivers who accepted the coupon. At 0 (never), more people rejected the coupon than accepting it, but as we see the number of times they visit a coffeehouse or bar increasing (1-4), they start accepting the coupons more often than rejecting them.

Another two variables that were one of the most important in both models were coupon_carry out & take away, and coupon_Restaurant(<20). The meaning of the first feature is essentially the type of coupon that was offered was for a restaurant that was carryout/takeaway, and the latter is the average price of the restaurant the coupon pertains to. This also lines up with our initial thoughts and exploratory data analysis, as we saw people much more likely to accept coupons for these places rather than not. Restaurants that are of this type are typically cheaper, have more visits from people in a given time, and therefore may be more applicable/appealing to more people.

Time was the last variable that seemed to generally score in the top 5 feature importances for both models. This was also highlighted in our initial histograms. Earlier in the morning, around 7AM, people may be in a rush to get to work or school and are less likely to be accepting coupons. Afterwards, later in the morning and afternoon, as well as after work (6PM) they are less likely to be in a rush and more accepting of coupons offered to them. However, late at night (10PM), people may be rushing to get home once again to sleep before having to wake up for work/school early in the morning again and be less likely to accept coupons at this time.

Combining all these variables together, from a business standpoint, we can highlight a few noteworthy assumptions. First, places that people may visit multiple times a month may include coffeehouses and bars. Customers may be more accepting of coupons from Carryout/Take away restaurants and restaurants whose average meal costs less than $20. A business that falls under these categories may find more success with coupons and find the tradeoff between the cost of advertising (coupons) to be worth, while restaurants that are more expensive or see less visits from the same people in a month may want to reconsider or do more research in terms of how they want to advertise/ attract customers to their business. Lastly, the time of day in which the coupon may be very important. Passing out coupons early before work or too late at night will not be efficient in terms of cost and resource allocation. Instead, passing them out midday or in the early evening may be more effective. Businesses that are considering coupons to attract customers should consider all these things (and likely many others) to determine how beneficial it will be to them.