Part I (KNN Analysis):

The baseline accuracy with KNN was determined to be **65.3% -** via optimization of the parameters and factors documented below, this accuracy was improved to nearly **80%** with an AUC value of **85.2.** The following factors provided the best accuracy: FromGrade, IsNonAnnual, ToGrade, Tuition, TotalSchoolEnrollment, FPP, SchoolSponsor, TotalPax, and SingleGradeTripFlag. While the following parameters provided the best accuracy: 20 neighbors, 2 dist_power, and a rectangular weight function. At first, I tried tuning each of the parameters and noticed minor gains in accuracy. Then I pivoted to optimizing each of the parameters by first testing whether the original factors resulted in a gain or loss in accuracy when taken out, thereafter testing whether new quantitative factors resulted in a gain or loss in accuracy - one at a time. After optimizing the factors, I re-tuned the parameters by testing similar adjustments to the first iteration of parameter tuning (e.g. testing each of the weight functions).

See Appendix for Final Confusion Matrix and ROC Area Under the curve & Curve Visual.

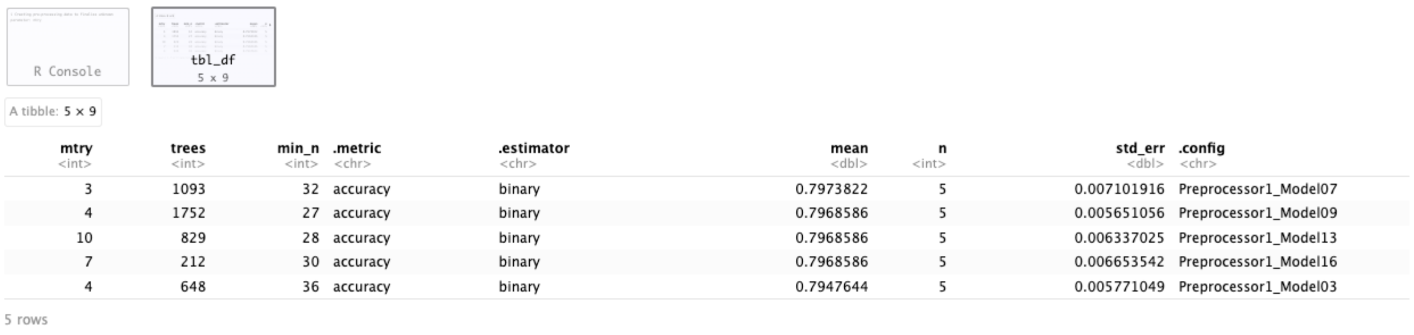
Run Number	Factors	Neighbors	Dist_Power	Weight_Func	Accuracy	Comments
Original	**ToGrade, Tuition, Days, TotalSchoolEnrollment, FPP**	**20**	**2**	**Rectangular**	**65.3%**	**N/A**
1	ToGrade, Tuition, Days, TotalSchoolEnrollment, FPP	20	1	Rectangular	65.3%	Decreasing Dist_Power has no effect on the accuracy of the model.
10	ToGrade, Tuition, Days, TotalSchoolEnrollment, FPP	20	4	Gaussian	65.3%	Switching to a Gaussian weight function results in lower accuracy, all else equal.
16	ToGrade, Tuition, Days, TotalSchoolEnrollment, FPP	20	4	Inv	65.3%	When comparing all other weight functions (all else equal), the rectangular weight function results in the highest accuracy.
18	FromGrade, IsNonAnnual, ToGrade, Tuition, Days, TotalSchoolEnrollment, FPP	20	4	Rectangular	77.0%	IsNonAnnual is found to be a key variable; when added to the model (all else equal) we observe a >7% increase in accuracy.
23	FromGrade, IsNonAnnual, ToGrade, Tuition, TotalSchoolEnrollment, FPP	20	4	Rectangular	77.5%	Relative to run 18 (all else equal), when removing Days we observe a 0.5% increase in accuracy.

34	FromGrade, IsNonAnnual, ToGrade, Tuition, TotalSchoolEnrollment, FPP, SchoolSponsor, TotalPax	20	4	Rectangular	77.9%	Relative to run 23 (all else equal), when we add SchoolSponsor and TotalPax we observe a 0.4% increase in accuracy.
38	FromGrade, IsNonAnnual, ToGrade, Tuition, TotalSchoolEnrollment, FPP, SchoolSponsor, TotalPax, SingleGradeTripFlag	20	4	Rectangular	78.2%	Relative to run 34 (all else equal), when we add SingleGradeTripFlag we obvserve a 0.3% increase in accuracy.
40	FromGrade, IsNonAnnual, ToGrade, Tuition, TotalSchoolEnrollment, FPP, SchoolSponsor, TotalPax, SingleGradeTripFlag	20	2	Rectangular	79.1%	Relative to run 39 (all else equal) we observe a 0.7% increase in accuracy when dist_power is decreased from 3 to 2.
43	FromGrade, IsNonAnnual, ToGrade, Tuition, TotalSchoolEnrollment, FPP, SchoolSponsor, TotalPax, SingleGradeTripFlag	25	2	Rectangular	76.2%	All else equal, it is found that accuracy is not improved by increasing or decreasing the number of neighbors in increments of 5.
49	**FromGrade, IsNonAnnual, ToGrade, Tuition, TotalSchoolEnrollment, FPP, SchoolSponsor, TotalPax, SingleGradeTripFlag**	**20**	**2**	**Cos**	**~80%**	**When doing a final round of optimization of the weight function, we find the highest accuracy using the Cos weight function.**

Part II (Random Forest Analysis):

Through adding/subtracting predictors and tuning the mtry, min_n and trees parameters, the accuracy has been improved to 82.25%. Random forests create an ensemble of diverse trees by randomly selecting subsets of the data and features to build multiple trees, and then average their predictions. Each tree is created from a random subset, ensure the individual trees are diverse, each captures different trends in data; the random forest takes a vote from each of the trees it has built, and choose the most popular outcomes as the final prediction.the Adjusting parameters can significantly impact the model’s performance, and they are often tuned using cross-validation methods. For example, here I use tune() to automatically select the best hyperparameter value based on some criterion, such as cross-validation accuracy.

```
rf_recipe <-
  recipe(RetainedLabel ~ SPRNewExisting + FromGrade + InitialSystemDate + FRPActive +
LatestRPL + FromGrade + FPPtoSchoolenrollment + SPRGroupRevenue + FRPTakeuppercent
+ DepartureDate + FPPtoPAX + DifferenceTraveltoFirstMeeting + DepositDate + ReturnDate +
IsNonAnnual + Tuition + ToGrade + TotalSchoolEnrollment + FPP + SchoolSponsor + TotalPax
+ SingleGradeTripFlag, data = train)  # <-- the dot here indicates use all columns for prediction


rf_model <-
  rand_forest(mtry = tune(), min_n = tune(), trees = tune()) |>
```



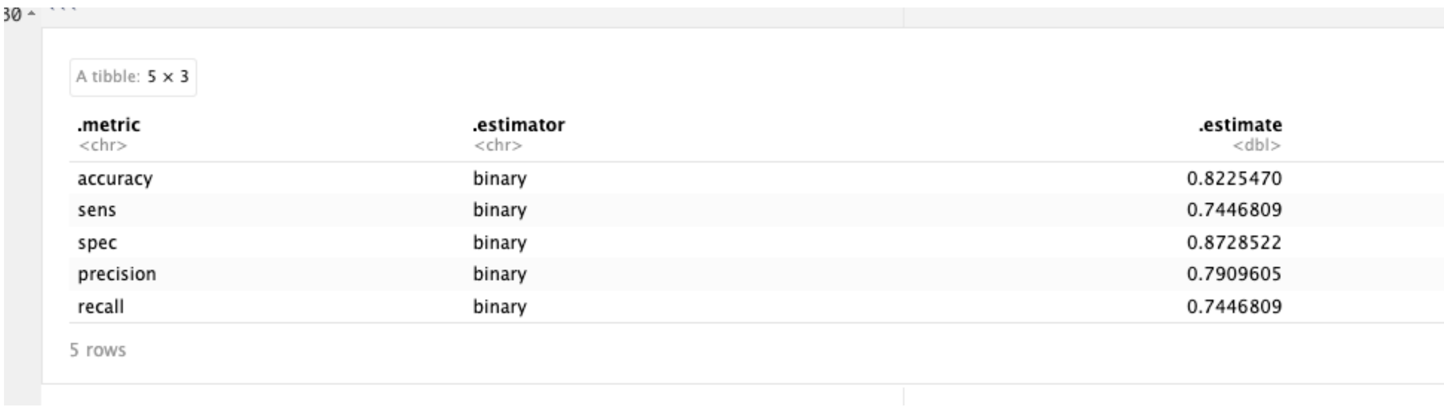
A tibble: 5 x 9

mtry <int>	trees <int>	min_n <int>	.metric <chr>	.estimator <chr>	mean <dbl>	n <int>	std_err <dbl>	.config <chr>
3	1093	32	accuracy	binary	0.7973822	5	0.007101916	Preprocessor1_Model07
4	1752	27	accuracy	binary	0.7968586	5	0.005651056	Preprocessor1_Model09
10	829	28	accuracy	binary	0.7968586	5	0.006337025	Preprocessor1_Model13
7	212	30	accuracy	binary	0.7968586	5	0.006653542	Preprocessor1_Model16
4	648	36	accuracy	binary	0.7947644	5	0.005771049	Preprocessor1_Model03

5 rows

```
rf_model <-
  rand_forest(mtry = 3, min_n = 32, trees = 1093) |> # <-- Best parameters from tuning above
```

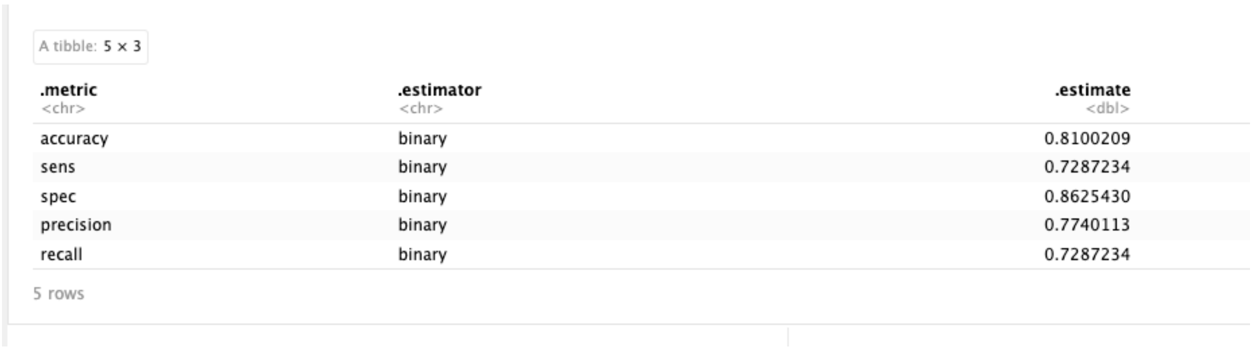
Mtry=3 means at each split, the algorithm randomly will select 3 features from total available features. min_n=32 means the minimum number of data points required to perform a split at a node is 32, sets a limit on the tree's depth and prevents overfitting. trees=1093 means the random forest will consist of 1093 decision trees.



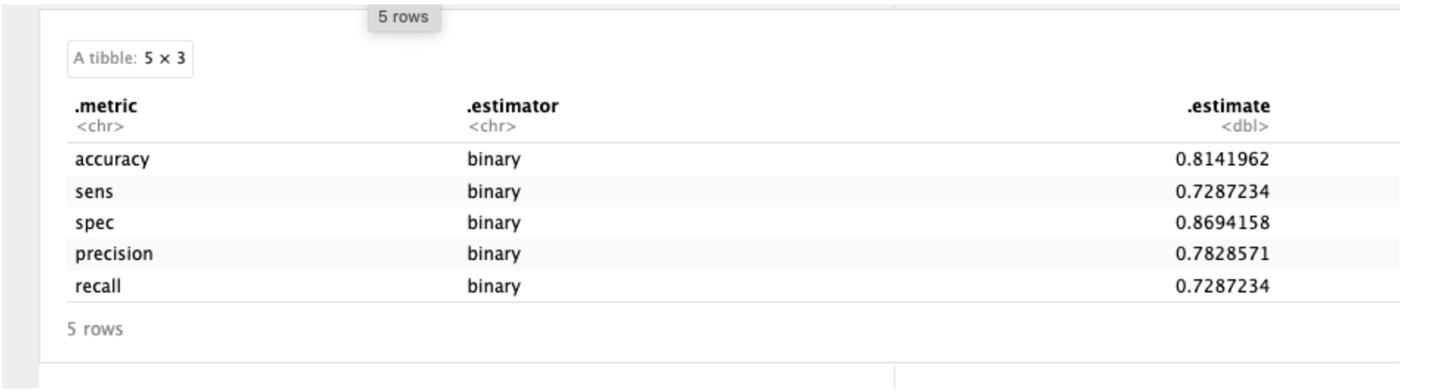
A tibble: 5 x 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.8225470
sens	binary	0.7446809
spec	binary	0.8728522
precision	binary	0.7909605
recall	binary	0.7446809

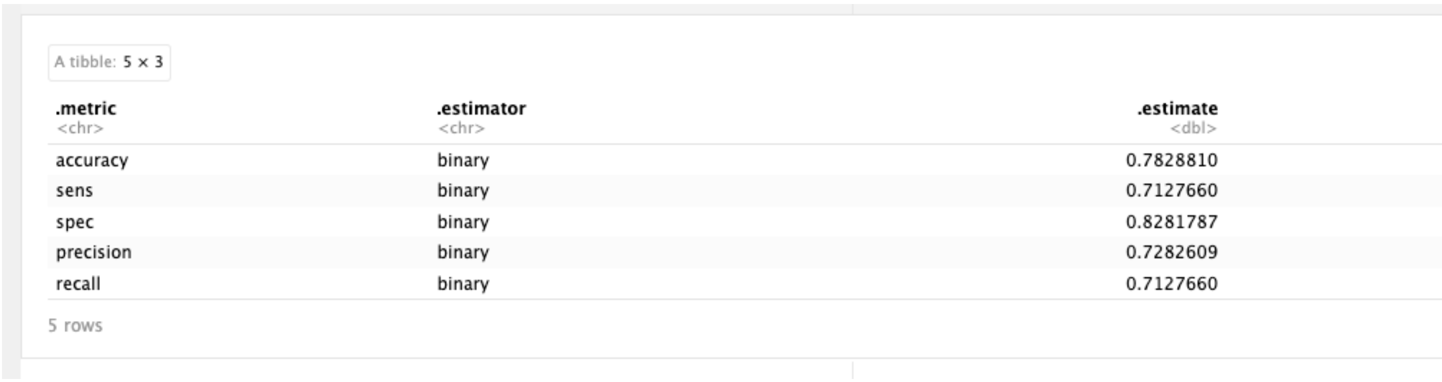
5 rows



```
A tibble: 5 × 3

.metric      .estimator      .estimate
<chr>        <chr>           <dbl>
accuracy     binary          0.8100209
sens         binary          0.7287234
spec         binary          0.8625430
precision    binary          0.7740113
recall       binary          0.7287234

5 rows
```



```
5 rows

A tibble: 5 × 3

.metric      .estimator      .estimate
<chr>        <chr>           <dbl>
accuracy     binary          0.8141962
sens         binary          0.7287234
spec         binary          0.8694158
precision    binary          0.7828571
recall       binary          0.7287234

5 rows
```

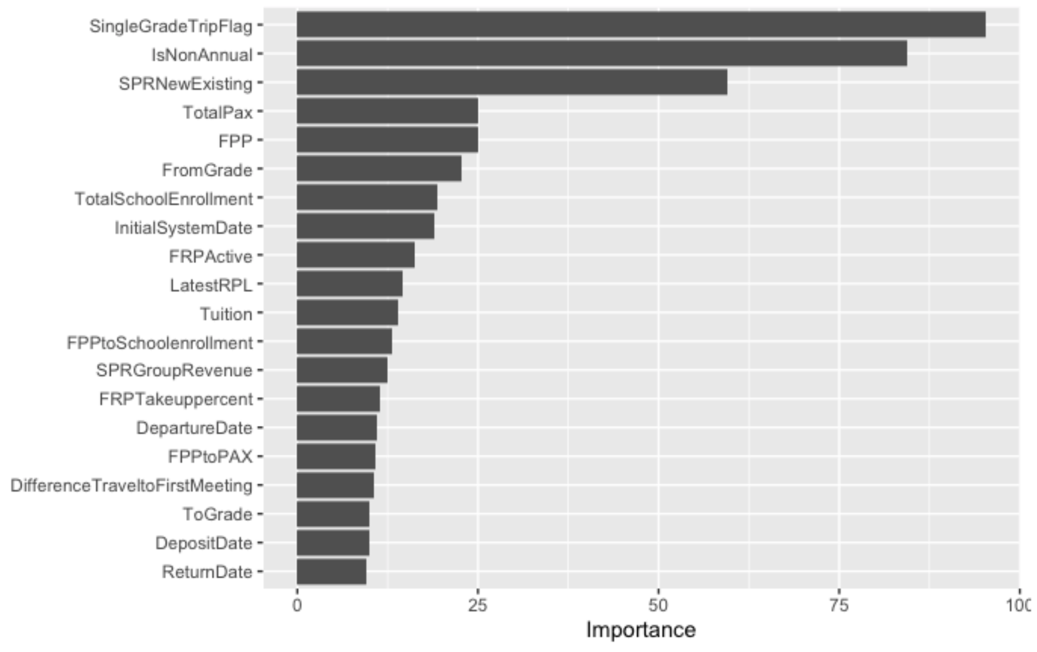


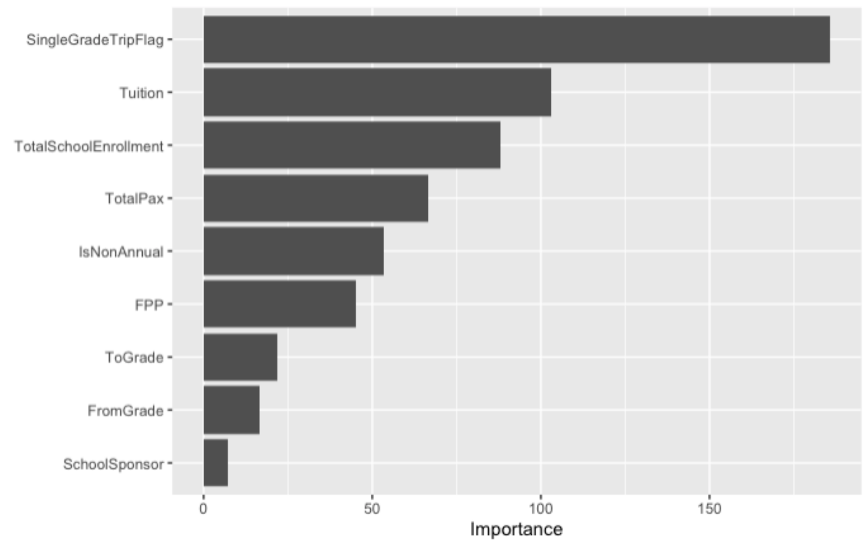
```
A tibble: 5 × 3

.metric      .estimator      .estimate
<chr>        <chr>           <dbl>
accuracy     binary          0.7828810
sens         binary          0.7127660
spec         binary          0.8281787
precision    binary          0.7282609
recall       binary          0.7127660

5 rows
```

From the above comparisons, we can see when the accuracy is improved, all the other metrics such as sensitivity, specificity, precision and recall are improved also, the model has a better performance. Higher sensitivity indicates the model is good at identifying positive cases when they are present. High specificity indicates that the model is good at identifying negative cases when they are present. The higher precision tells us there are more true positive cases out of all positive predictions. Recall is synonymous with sensitivity, more actual positive cases were identified by the model; the model has greater ability to capture all positive cases. For example, higher recall in medical diagnosis means a higher proportion of sick people are correctly identified as sick out of all the sick people.

Parameter Values	**Accuracy**
trees=100 (in the given file), all parameters	0.810

trees=tune(), all parameters	0.814
trees=tune(), 9 parameters	0.783
trees=tune(), all the top 20 most important parameters	0.8225





The given Rmd file has accuracy 0.810. First, I replaced trees=100 with trees =tune() in step 2, to determine number of trees, tuning hyperparameters in a decision tree model; I ran all lines, and got the accuracy 0.814 and the ranking of the top 20 important columns, after I put in the best parameters from tuning above and run again. Second, I added the columns we chose to use in the KNN recipe, FromGrade + IsNonAnnual + Tuition + ToGrade + TotalSchoolEnrollment + FPP + SchoolSponsor + TotalPax + SingleGradeTripFlag, to my step 1 recipe instead of all columns, with the already updated step 2. When I ran again, I got an accuracy of 0.783, so I know it is better to include more parameters for this model. I deleted the columns and replaced them with the 20 columns from the ranking of the first run, and now the accuracy increased to 0.8225. Random forests aim to include all available features in each decision tree to promote diversity, to reduce correlation, capture different aspects of the data and handle high-dimensional data.

See Appendix for Final ROC Area Under the curve & Curve Visual.

Part III (Comparison of Models):

Metric	**KNN**	**Random Forest**	**Analysis**
Accuracy	Approximately 80%	82.25%	Random forest has slightly higher accuracy
Sensitivity	63.8%	74.4%	Random forest has the better ability to detect true positives
Specificity	90%	87.2%	KNN is very effective at identifying negatives
Precision	80%	79%	KNN is slightly higher, when it predicts a positive, it is correct 80% of the time.
Recall	63.8%	74.4%	Aligns with sensitivity

With these metrics, the Random Forest model shows a superior performance in accuracy and recall, making it more reliable for predicting true positives—customers who will return. However, KNN shows a higher specificity, indicating it is better at minimizing false positives, a valuable trait if the cost of a false positive is high.

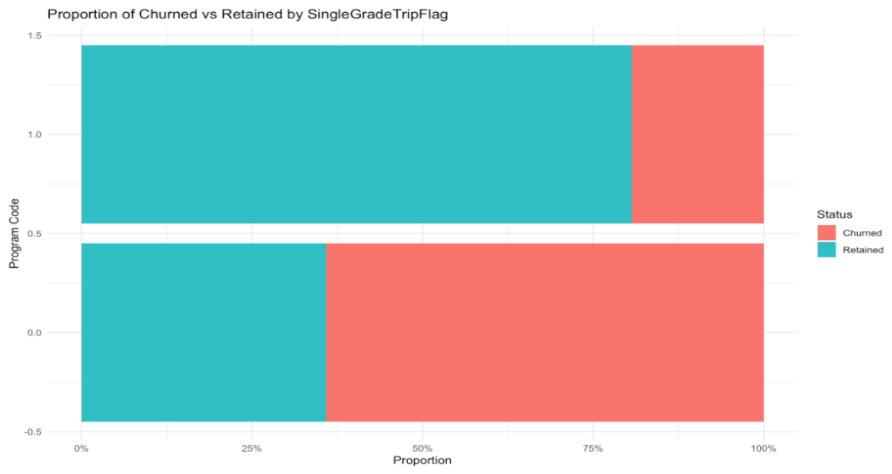
The lower sensitivity of the KNN model (63.8%) compared to the Random Forest (74.4%) suggests it may miss a significant number of potential return customers. This could be a drawback if the goal is to capture as many potential returnees as possible, even at the risk of some false positives.

On the other hand, the higher specificity of the KNN model (90%) means it performs well in avoiding false positives—identifying customers who are not likely to return. This could be particularly beneficial in a scenario where the business strategy prioritizes precision in targeting to conserve resources or in high-stakes environments where the cost of targeting wrong individuals is substantial.

Part IV (Recommendations):

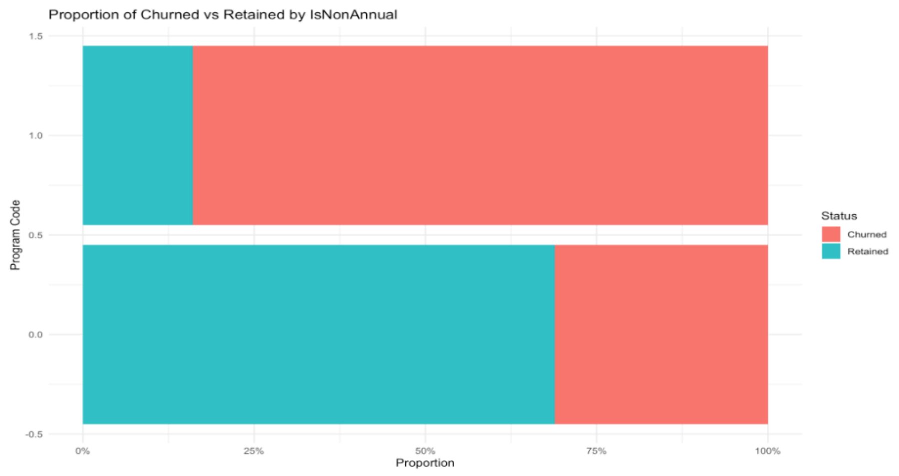
Based on the variable importance plot from the Random Forest model, the below predictors are confirmed to be important in customer retention prediction:

1. <u>Single Grade Trips</u>: The 'SingleGradeTripFlag' appears to be the most important predictor of retention. This suggests that trips taken by a group comprising students from the same grade have a higher retention rate.



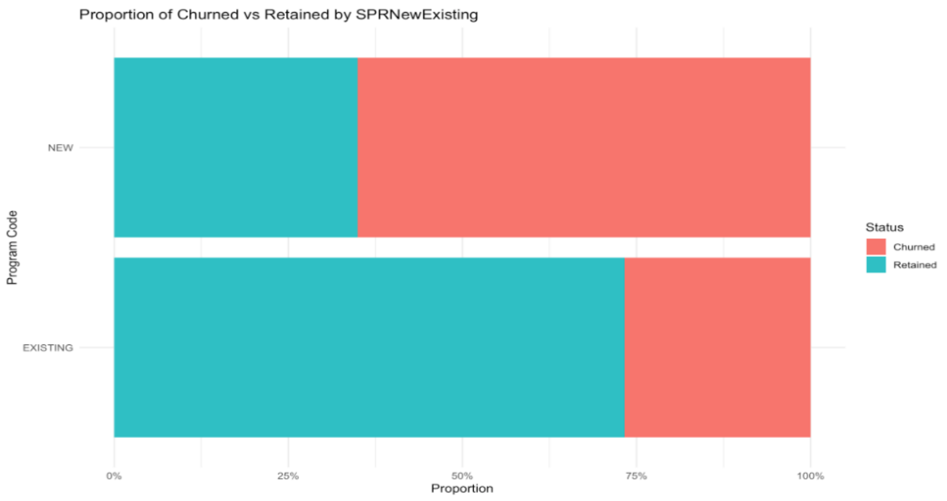
As confirmed by the above plot, the proportion of retained customers is higher for single grade trips. We recommend that the company focus on promoting single-grade trips more aggressively. It could also develop and tailor packages specifically for single grades, considering factors such as age-appropriate activities and curriculum alignment to make these trips more attractive to schools.

2. <u>Annual vs. Non-Annual Trips</u>: 'IsNonAnnual' is also a significant predictor, which indicates whether if the group from this school typically skips a year in between programs affects retention.



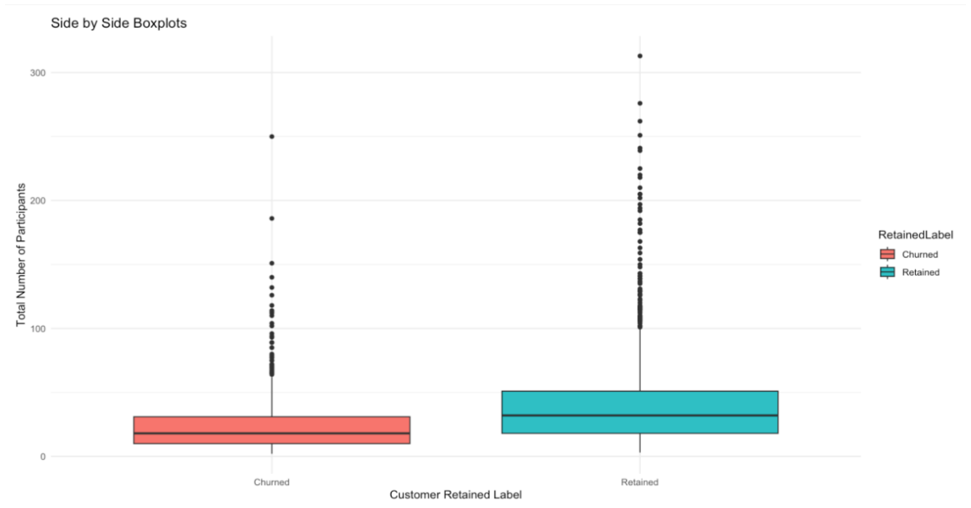
As confirmed by the plot above, the proportion of retained customers is lower for schools which skip a year. We recommend they strengthen relationships with schools that have annual trips, as these are likely to have higher retention rates. For schools organizing non-annual trips, the company could offer incentives or loyalty programs to encourage them to plan future trips.

3. <u>New vs. Existing Customer Programs</u>: The 'SPRNewExisting' variable suggests that the retention could be influenced by whether the customers are new or existing.



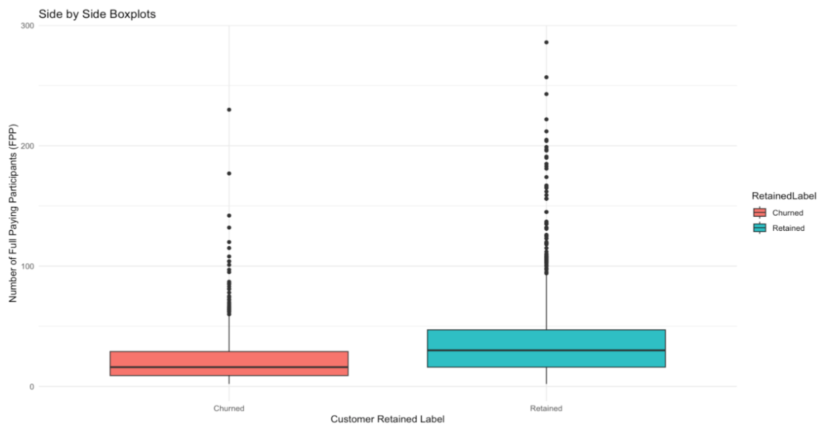
As confirmed by the above plot, the proportion of retained customers is higher for existing customers. We recommend the company design targeted marketing strategies for new and existing customers. For new customers, they could provide educational materials and support to ensure a successful first trip. For existing customers, they could consider offering loyalty discounts or perks for repeat bookings.

4. <u>Total Number of Pax (Passengers)</u>: The 'TotalPax' variable implies that the number of passengers in a trip could influence the likelihood of retention.



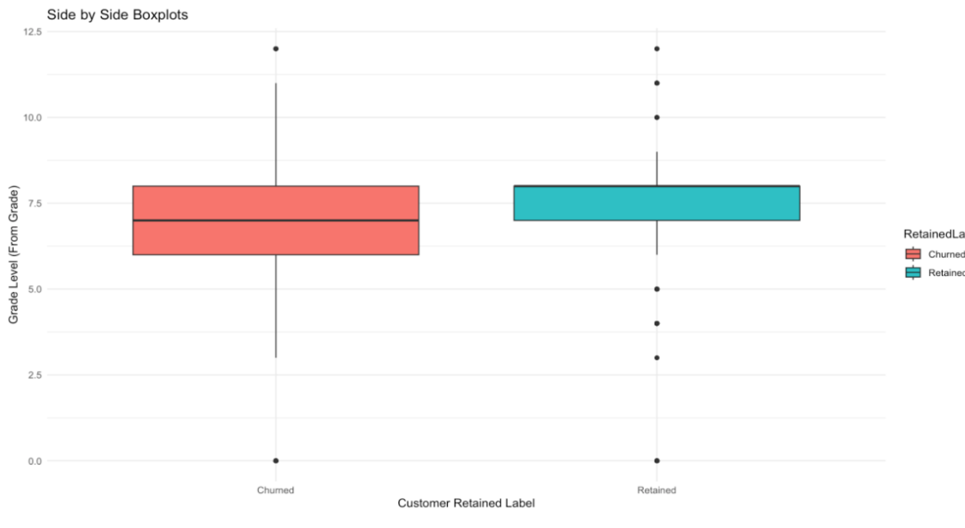
As confirmed by the above plot, the retained group shows a higher median for number of passengers. This confirms that higher participation leads to higher retention. We recommend that they investigate the optimal group size for high retention and develop offers that encourage organizers to reach or maintain this group size.

5. <u>Payment Factors: The 'FPP' (Full Paying Participants)</u> variable is indicative of financial commitment to the trip.



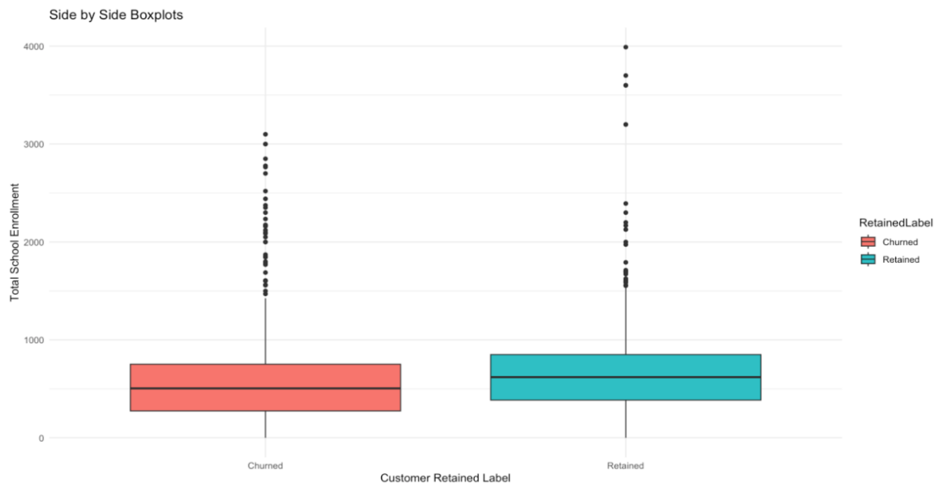
As confirmed by the above plot, the retained group shows a higher median for full paying participants. We recommend that they implement flexible payment plans or early bird discounts that might encourage full payments ahead of time. A higher full payment percentage might increase the commitment level of the participants and lead to higher retention.

6. <u>Grade Level: 'FromGrade'</u> has some importance, suggesting that the grades of customers participating in the trips play a role in retention.



As confirmed by the above plot, the retained group are from higher grade levels, the lowest grade of students on a trip from the retained group is 7.5. We recommend they tailor trips to students above the 7th grade, and consider the developmental and educational needs of these grades. Differentiating the trip offerings by grade can also help teachers and parents see the value in annual trips as part of their educational progression.

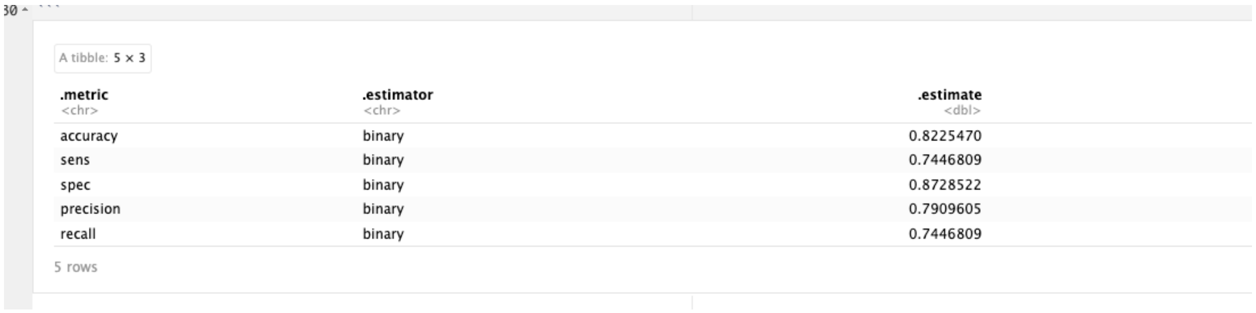
7. <u>School Size and Engagement:</u> 'TotalSchoolEnrollment' is reflects the student population the school.



As confirmed by the above plot, the retained group are from larger schools. We recommend they develop strategic partnerships with large schools to tap into a bigger customer base and work on programs that can be integrated into the schools' yearly activities.

Appendix

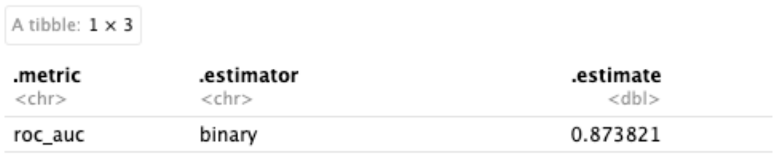
Random Forest Metrics:



A tibble: 5 × 3

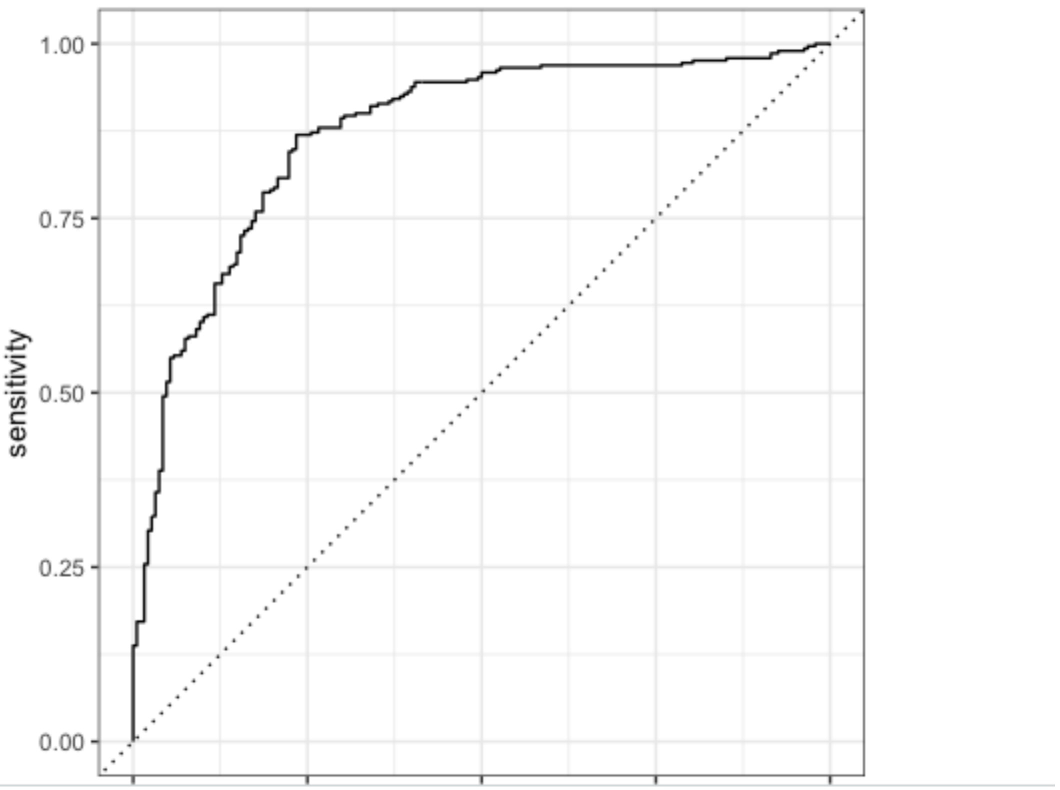
.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.8225470
sens	binary	0.7446809
spec	binary	0.8728522
precision	binary	0.7909605
recall	binary	0.7446809

5 rows

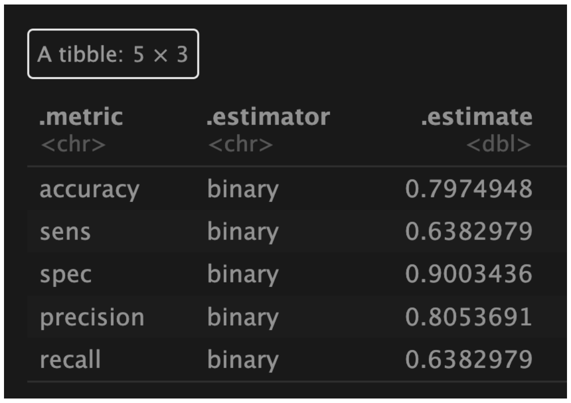


A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
roc_auc	binary	0.873821

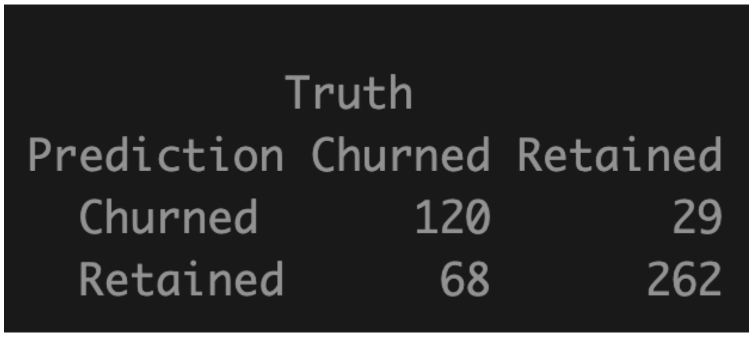


KNN Metrics:

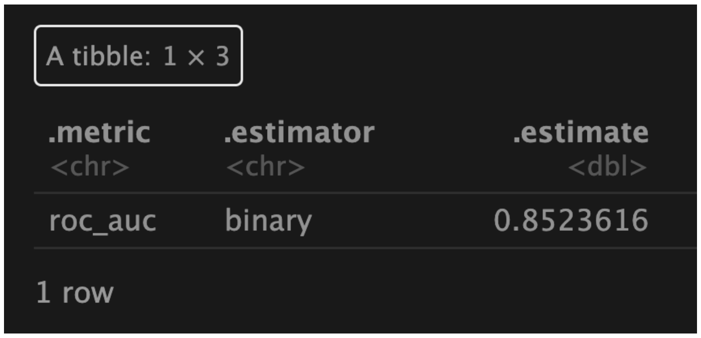


A tibble: 5 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.7974948
sens	binary	0.6382979
spec	binary	0.9003436
precision	binary	0.8053691
recall	binary	0.6382979



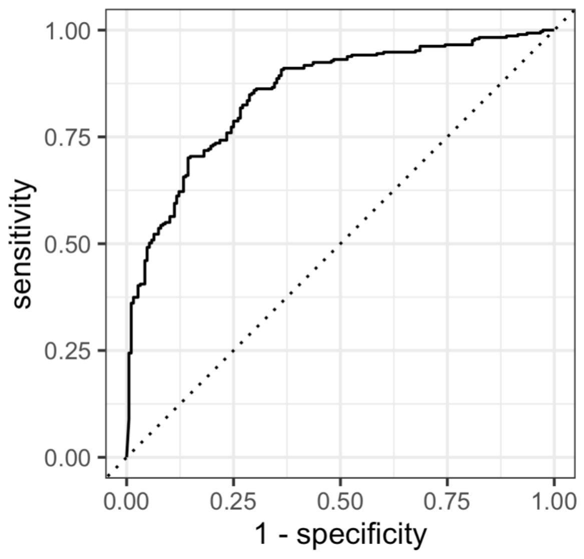
```
          Truth
Prediction Churned Retained
  Churned      120       29
  Retained      68      262
```



A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
roc_auc	binary	0.8523616

1 row


sensitivity
1 - specificity