**Implementation of Model Training & Intermediate Result**

**Naïve Bayes Model**

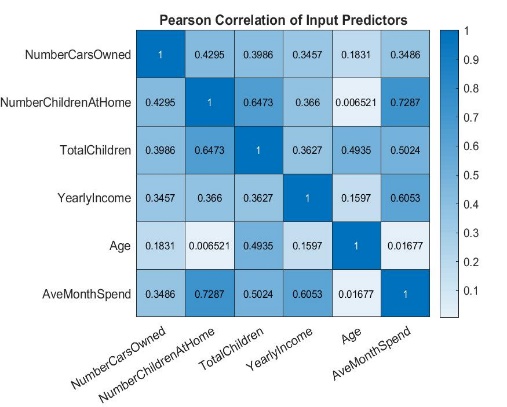
Chart

Description automatically generatedBaseline Model is construct with default settings i.e. assume numeric predictors as normally distributed and all predictors are independent from each other. Class Prior = no. buyer/ total customer

1. **Feature Selection**

* *Removal of Redundant Predictors in Categorical input*

Amongst all predictors, 3 categorical input shows locational characteristics and are inter-explainable. Consider ‘City’ could not be isolated from ‘StateProvinceName’ to identify the location, two models are trained for either removal of ‘City + State’ and ‘CountryRegionName’. However, both models do not perform better than baseline model.



* *Removal of Highly Correlated Predictors*

As Naïve Bayes assumes independent input variables, highly correlated predictors are removed one by one and 6 models are trained. Results suggest that compared with ‘AveMonthSpend’, loss of NumberCarOwned is more significant and shows rise in F1 score. NumberCarOwned is removed.

1. **Discretization of Numeric Predictors**

Scholars[[1]](#footnote-1) suggest a well-defined discretization approach could manage bias and variance in Naïve Bayes Model. Kernel distribution is assumed to foster flexible training.

* *Age Binning (Fixed Interval)*

Age is assumed to be binned in fixed-interval instead of fixed number of bin, as its maximum value is assumed as 100. Intervals range from 4 to 20 are trained are results suggests interval=4 can improves F1 score with neglectable trade off in validation error.

* *Yearly Income Binning (Fixed-bin)*

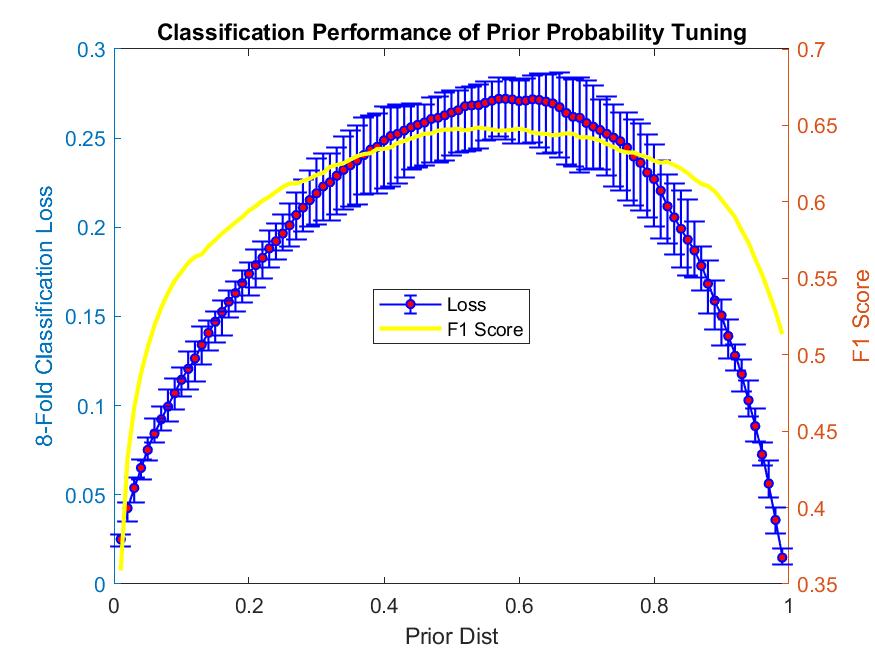
Yearly Income is discretized with a fixed number of bins, as the maximum value in test set is unknown and any value above max. value in training data can be discretized into the last bin. Bin number ranged from 4-20 are trained and results show bin=12 can improve both performance.

* *Ave Monthly Spend Binning (Fixed-bin)*

Same training process as previous is performed and results show bin= 20 improve both performance.

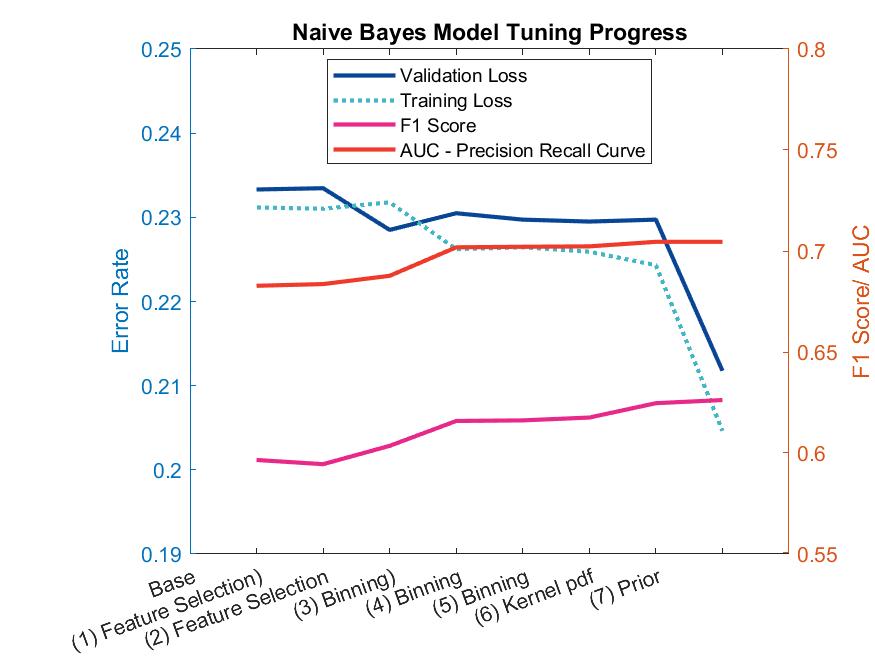
1. **Hyperparameter Optimization**

Distribution Name, kernel type and corresponding bandwidth is optimized by grid research to fit in the data. Data is scaled to [0,1] before performing the grid for optimizing the bandwidth for all numeric predictors. Optimized Kernel is ‘box’ and width = 0.12938. With neglectable trade off in overall accuracy, F1 score has significantly improved.

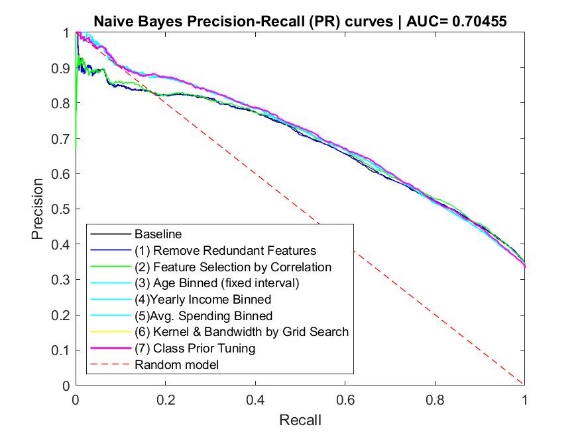
1. **Additive smoothening**

Additive smoothing adapted for categorical predictors to reduce impact of Zero Conditional Probability. Distribution of categorical data are set as ‘Multivariate multinomial distribution’.

1. **Class Weight/ Prior Tuning**

As the dataset is imbalanced, its necessary to tune to prior to maximize the positive accuracy under the same Area Under Curve. Range from 0.01 to 0.99 with interval 0.01 are trained. Considering the variation and bias, the optimal prior = [0.17 ,0.83]. More weight is given to minority class.

1. **Conclusion:** Improvement and lag back as follows:

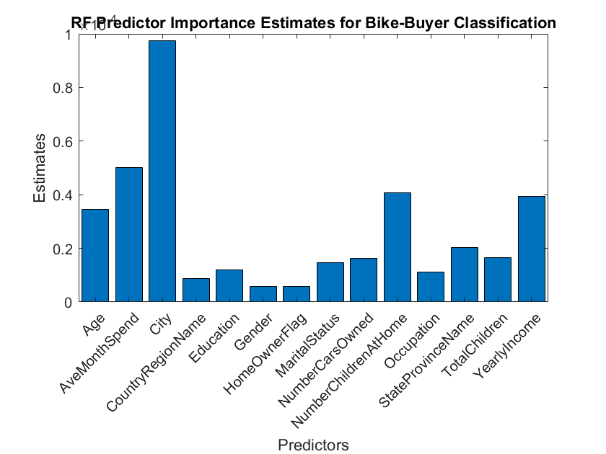


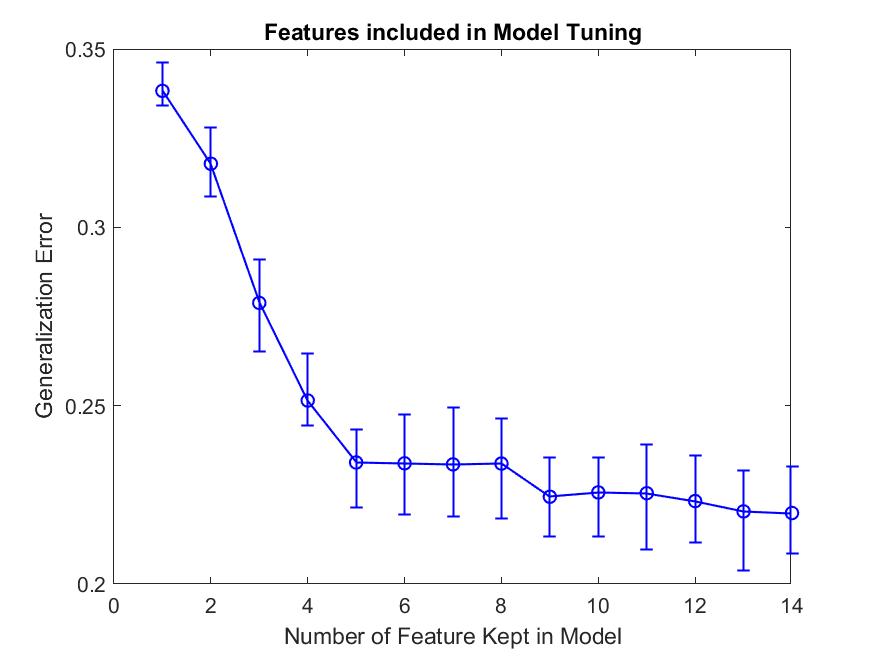
**Random Forest Model**

Baseline Model is construct with default settings with ensemble and bagging method, where MaxNumSplit= Total observation -1; MinLeafSize =1; Prior = No. of Class / Total Observation; NumVariablesToSample = all (14); Split criterion = Gini index.

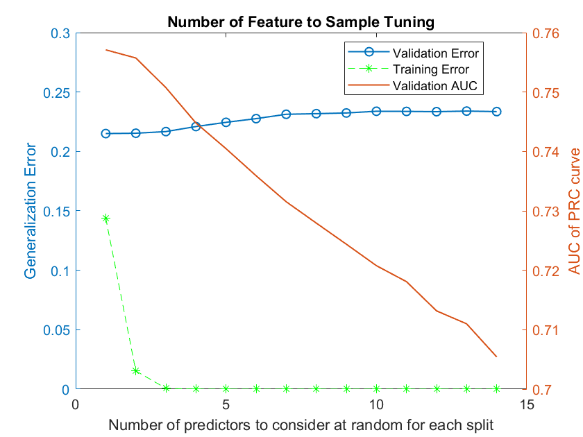
1. **Number of Learning Cycles/ Trees Grown**

RF model is first run to find the number of tree grown such that a low and stable error rate would be obtained. At least No. of tree grown = 800 after observing their cumulative error.

1. **Feature Selection – Predictor Importance**

All input predictors are assessed to estimate their importance in tree model by summing changes in risk due to splits on every predictor and dividing by sum of branch nodes. Predictors which stands out from others are the top 5 ones, including City, AveMonthSpend, Number of Children, Yearly Income and Age. 14 models are trained if we start removing features from the least important one, whether the accuracy can be improved. As aligned with the left chart, results show that inclusion of top 5 important predictors significantly reduce error. Yet the error rate is constant when adding the 6-8th, and 9-14th importance predictors. This may imply the noises within two set of predictors are similar/ providing same information. However, results also suggest the inclusion of least important predictors minimized the overall error rate.

1. **Number of Feature to Sample in each split**

Models are trained from 1 feature to all 14 features. Considering the number of trees grown, error rate and F1 score, feature=2 is selected although F1 score is not at its maximum. All performance metric shows improvement.

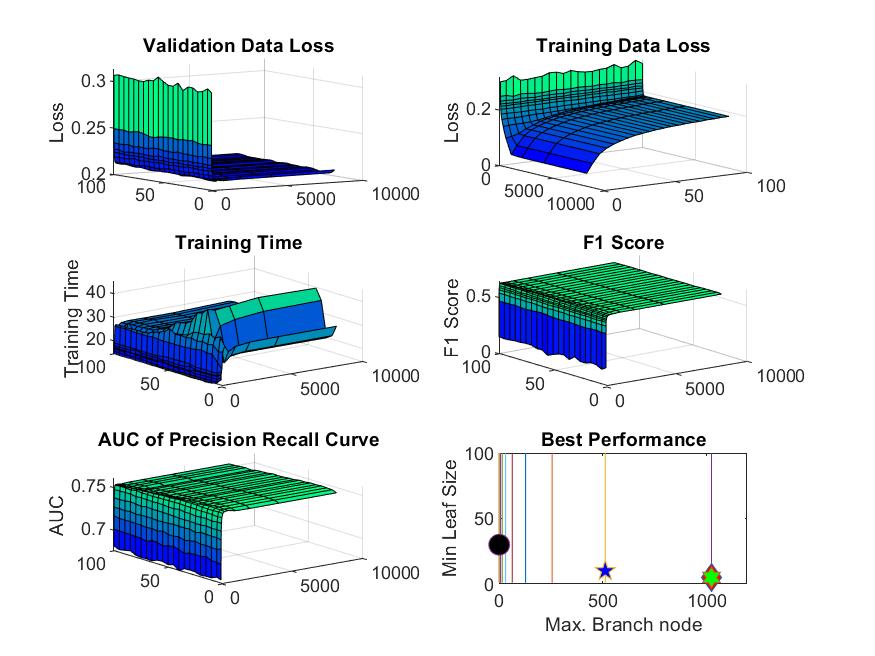
1. **Control Tree Depth**

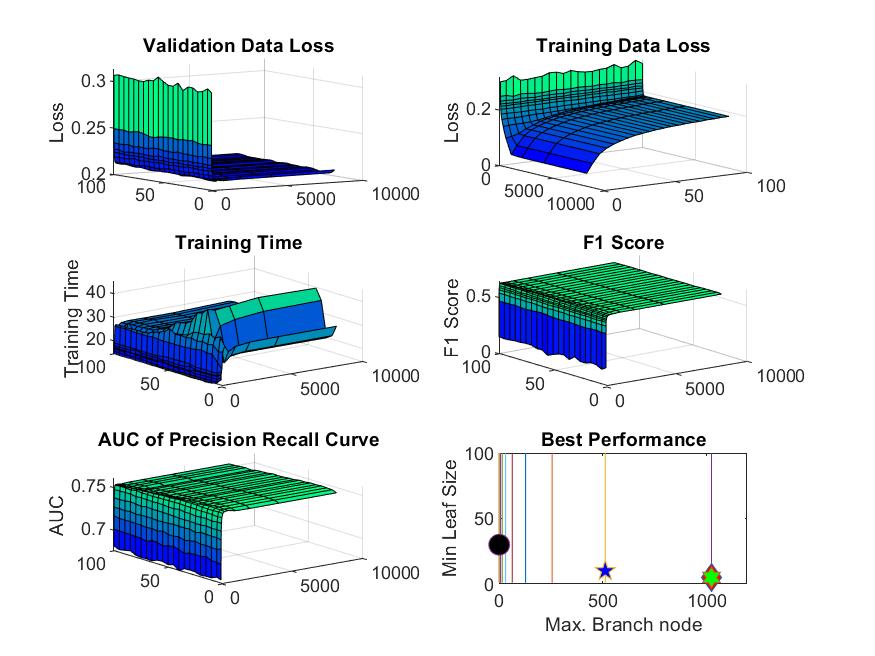
Loop within loop are run to consolidate both impacts from Leaf Size & Branch node and to minimize noise capture in classification.

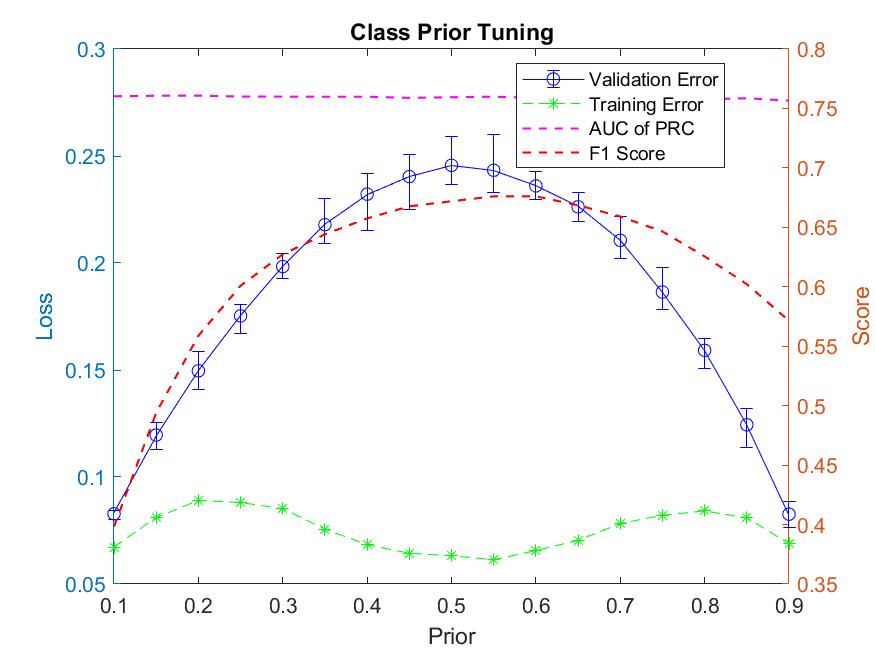
* *Maximum Split of Branch Code (x-axis)*

Binary Split is assumed in each branch code, hence models are trained in the power of 2, from 0 to log2(observation -1) as suggested.

* *Minimum Sample in each Leaf (y-axis)*

Leaf size are trained from 1 to 100 samples within each leaf, with interval of 5 with considering in the long training time. Stable change are observed until a final decision was made.



1. **Class Weighting Tuning**

In ensemble and bagging method, if the class prior probabilities are highly or skewed, the software tend to oversample unique observation from the majority class. Hence, certain weight shall be given to the training model as the simplest sampling approach. Prior is trained from 0.1 to 0.9, with interval of 0.05. F1 score peaks with the error rate. Therefore, only prior showing improvement in F1 score and accuracy are considered. If more than one are discovered, the optimal one [0.27,0.73] is the one with low variations.

1. Y. Ying & W. Geoffrey, " Discretization for naive-Bayes learning: Managing discretization bias and variance " in Machine Learning, vol. 74, pp. 39-74, 2009, doi: 10.1007/s10994-008-5083-5. [↑](#footnote-ref-1)