WEEK 7

For this week, we developed a random forest model to predict the binary outcome of customer churn. This approach uses ensemble learning, which involves combining multiple decision trees to create a more accurate and robust model. Random forest is a popular choice for handling missing data and outliers since our data had outliers, and it can provide valuable information on feature importance. By ranking the features based on their importance, we can identify the most significant variables that contribute to customer churn. From the table, we noticed that the features, in predicting whether a customer would churn or not, were ranked by importance from highest to lowest as follows: age, estimated salary, credit score, balance, number of products, tenure, is active member, geography_germany, has credit card, gender, geography_france and geography_spain.

```
Feature importance ranking by Random Forest Model:
Age : 0.2385
EstimatedSalary : 0.1468
CreditScore : 0.1432
Balance : 0.1422
NumOfProducts : 0.1328
Tenure : 0.0825
IsActiveMember : 0.0398
Geography_Germany : 0.0206
HasCrCard : 0.0184
Gender : 0.0181
Geography_France : 0.0091
Geography_Spain : 0.008
```

The complexity of the random forest modeling approach depends on the number of trees in the forest, the depth of each tree, and the number of features used to split each mode. Increasing any of these parameters can increase the model's complexity and potentially lead to overfitting. However, random forest has built-in mechanisms such as bagging to reduce overfitting and improve model performance. To optimize our model's performance, we focused on two of the most critical hyperparameters that control the size and depth of the tree ensemble: the number of tress (n_estimator) and maximum depth of the trees (max_depth). We used a grid search to vary these hyperparameters over a range of values to find the best combination of hyperparameters that yielded the highest model performance.

We selected performance metrics of accuracy, precision, recall and F1 score to evaluate the

performance of the random forest in predicting the binary outcome of whether a customer churned or not. Given that our dataset was imbalance, with class 0 being much more frequent than class 1, we paid more attention to precision and recall, as accuracy could be misleading,

To evaluate the performance of the model, we split the dataset into a training and validation set, and calculated the performance metrics for both sets. We then evaluated three variations of the model, which included the base model, the tuned model, and the model with smaller trees.

The base model was the starting point, with no hyperparameter tuning, and achieved almost perfect training accuracy of 0.99. However, the other two models had slightly lower training accuracy, with the tuned model achieving 0.91 and the model with small tress achieving 0.82.

We then tuned the parameters of the model by setting the paramters as {'n_estimators':[60, 80, 100], 'max_depth':[1, 5, 10]}, and selecting the best parameter, which turned out to be n_estimators= 80 and max_depth = 10. Finally, we used smaller trees, with n_estimators= 10 and max_depth = 3, to see how this would affect the performance of the model.

The performance metrics for the training and validation datasets for all three variations are presented in the table below. The tuned model achieved the highest validation accuracy of 0.86, indicating that the hyperparameter tuning improve the model's ability to generalize well to unseen data. Moreover, the validation recall values also followed a similar trend, with the tuned model having the highest validation recall values. This indicates that the tuned model was able to identify a high proportion of customers who were likely to churn out of all the customers who actually churned. Achieving high recall is desirable because it means that the bank can take appropriate actions to retain customers who are likely to churn, ultimately reducing customer attrition rates and improving overall customer satisfaction.

However, it is important to consider multiple evaluation metrics when assessing model performance. The model with small trees had the highest validation precision value, but its recall value and F1 score were lower than that of the other two models. This suggests that the model with small trees was better at identifying true positives but struggled to identify all the positive cases. Therefore, it may not be the best model for this specific scenario, where identifying all churn cases

is important.

Overall, our results suggest that the tuned model with n_estimators= 80 and max_depth = 10 outperformed the other two models. However, it is important to note that further hyperparameter tuning or exploring different models could potentially yield even better results.

Model	Train Accuracy	Train Precision	Train Recall	Train F1-Score	Validation Accuracy	Validation Precision	Validation Recall	Validation F1-Score
Random Forest (base)	0.999805	0.999877	0.999525	0.999701	0.857031	0.820035	0.701071	0.735833
Random Forest (tuned)	0.911523	0.935409	0.792290	0.839865	0.860156	0.832638	0.701637	0.738528
Random Forest (small trees)	0.825195	0.864591	0.581358	0.591378	0.817969	0.866941	0.562918	0.561385

In conclusion, our findings highlight that the performance of the models varies depending on the hyperparameters used, and hyperparameter tuning can effectively improve a models' performance on unseen data. It is crucial to balance model complexity with generalization performance, as the small -trees model demonstrated that simpler models may also have advantages.