Rational Statement

- Utilizing the ‘ChannelDataset’, trying to enhance predictive models and improve the performance of machine learning algorithms, thereby predicting the proper channel classification.

Feature Engineering Techniques

- **Tukey Method**

As commonly known in data science or statistics, the Tukey method is an easy way to detect any outliers with the dataset you are working on.

- **Smote**

Best method to utilize when working on an imbalanced dataset. Academically known as (Synthetic Minority Oversampling Technique), this statistical technique will generate instances from existing minority cases inputted in a balanced way, avoiding bias.

- **SelectFromModel**

Looking for a less robust model to perform model selection, which is not iterative then SFM is the right model to use. One benefit of this model is that; it looks at important features driving the model and, (2) which is the best model amongst the others.

The above methods were efficiently used for this assignment's completion

Insights from ‘ChannelDataset’

With a **standard deviation** of 12647.328865 and **mean** 12000.297727, it is obvious there is a large variance between the data and the statistical average, hence not so reliable.

Looking at ‘Pandas profiling report’ there is a high correlation between the following variables (Milk and Grocery, Milk and Frozen)

Also not forgetting our count statistics of 440 data points for the dataset as seen from the .describe() function.

Insight from Learning Curve

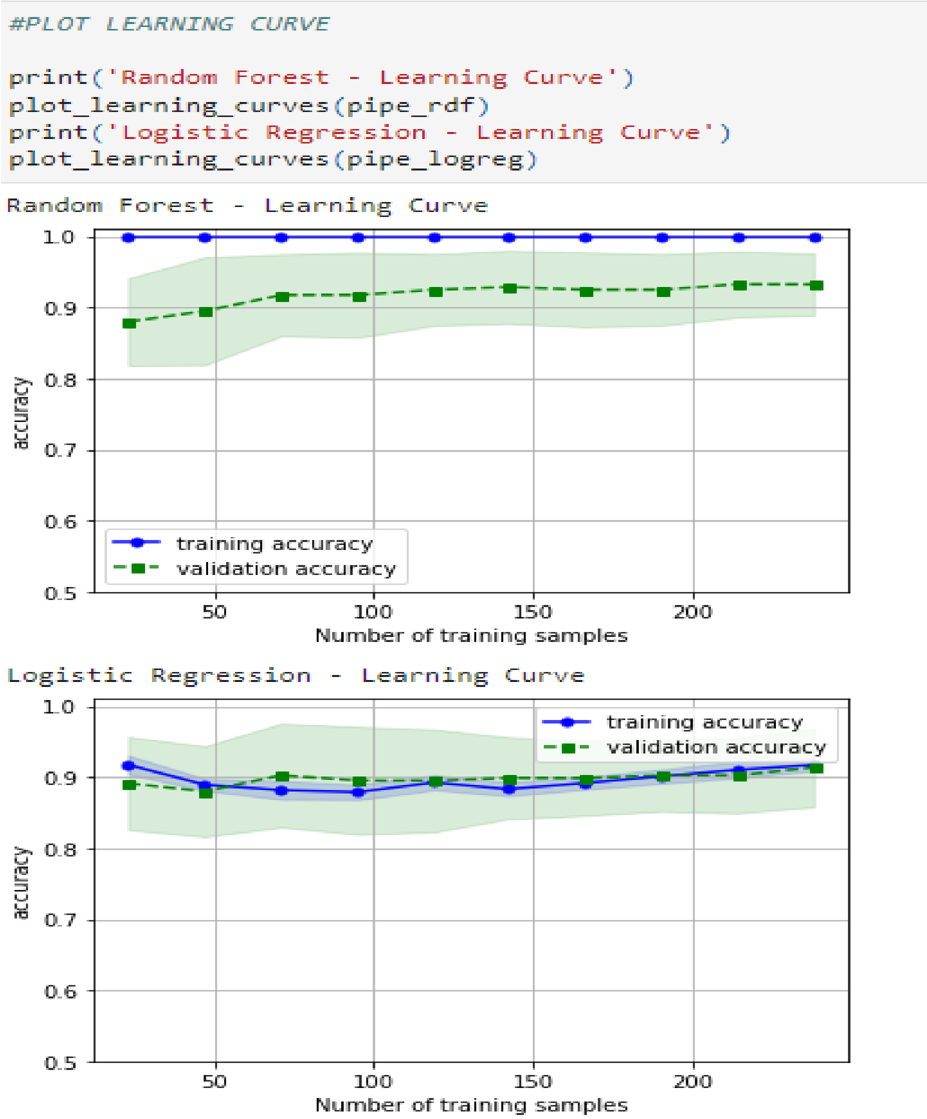
Learning Curve Random Forest (RF)

- Noticeably, there is a gap in the curves. But let us look at other insights, we get an accuracy close to 93% and the training set reaching and stable at 100%

Learning Curve Logistic Regression (LR)

- There is a clear drop in LR Curve, where accuracy falls close to 90% as compared to RF where the curve stood at 93%.

- Random forest seems to do a better job than the logistic Regression which has a drop in both training and validation



```
#PLOT LEARNING CURVE

print('Random Forest - Learning Curve')
plot_learning_curves(pipe_rdf)
print('Logistic Regression - Learning Curve')
plot_learning_curves(pipe_logreg)
```

```
Random Forest - Learning Curve
```

```
Logistic Regression - Learning Curve
```

Channel Classification Models

- **Logistic Regression**

We have two classes for the ‘channeldataset’ (1 & 2) of which LR is a powerful algorithm intended for two classes and linear classification. Although it can be extended to handle more than two classes, it is rarely used for that purpose.

- **Random Forest**

Known for its effectiveness with classification and regression problems, this supervised technique helped classify, build decision trees with bagging to bring out accurate outputs for classification.

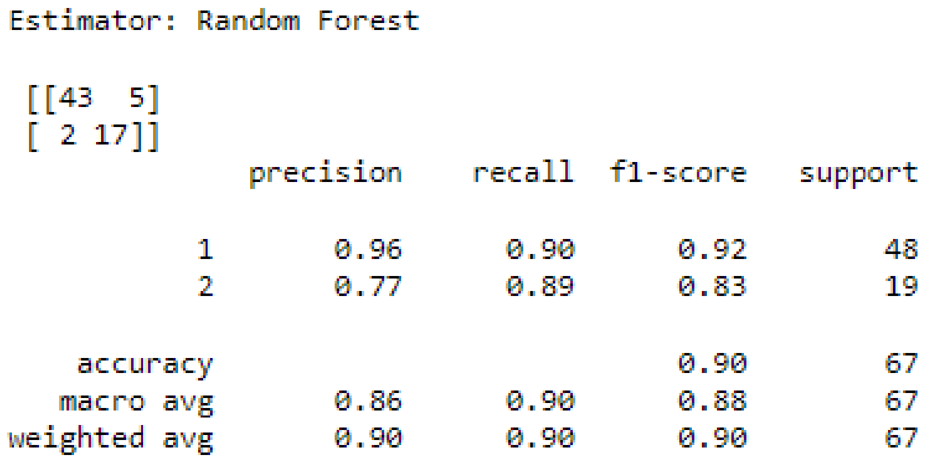
- **AdaBoost**

Used to boost the performance of the other machine learning algorithms, it should be noted this model is commonly used and can be seen to work best with one algorithm known as ‘Decision Tree’, (1) converts weak learner to strong learners, (2) learns from previous mistakes.

Classification Report Insights

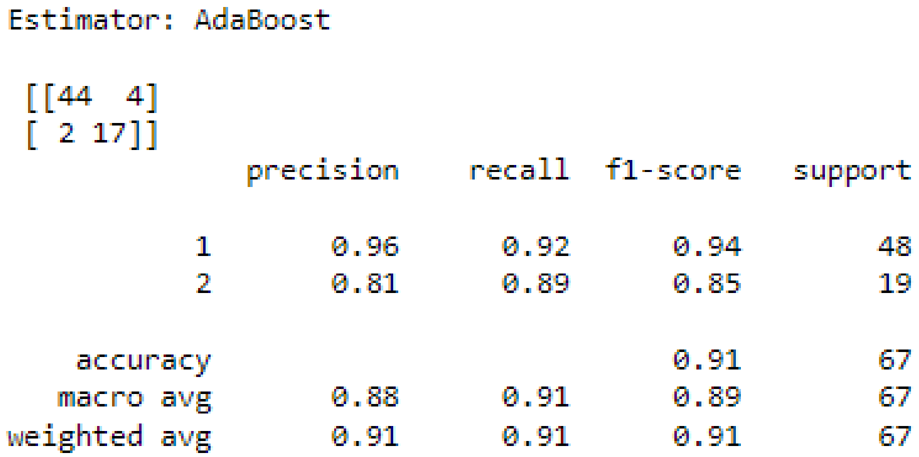
Random Forest (RF)

- Having an (accuracy of 90) which is not bad, and a macro average of 88
- The Precision of the model does better in predicting hotel/Café (1) but not so good with class (2) for Retail channel.
- Recall comes at 90, that is how good the model is at predicting results if ran again. 90 is good but a score of 95-99 could be better



```
Estimator: Random Forest

 [[43  5]
 [ 2 17]]
              precision    recall  f1-score   support

           1       0.96      0.90      0.92        48
           2       0.77      0.89      0.83        19

    accuracy                           0.90        67
   macro avg       0.86      0.90      0.88        67
weighted avg       0.90      0.90      0.90        67
```

Classification Report Insights



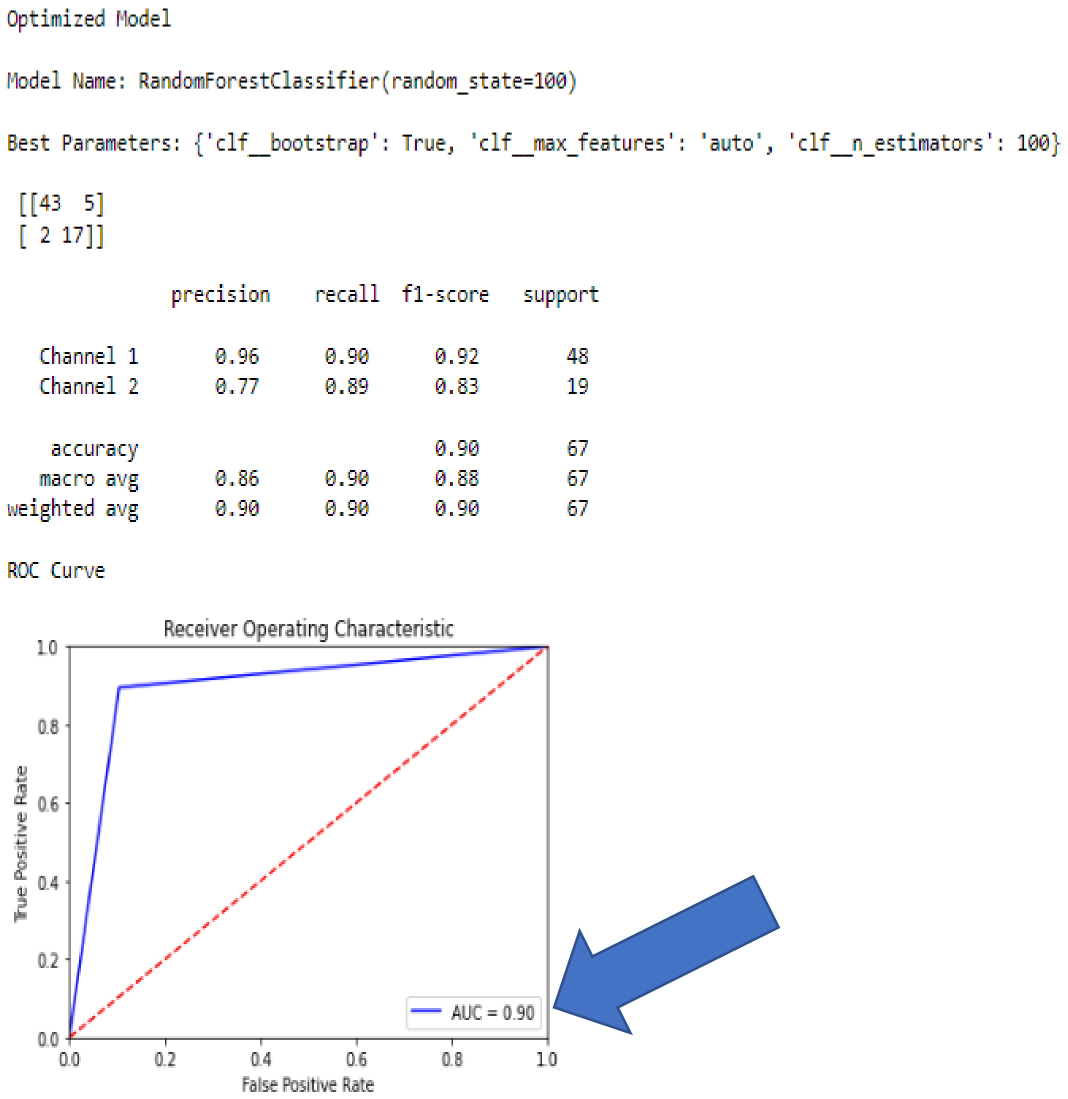
```
Estimator: AdaBoost

 [[44  4]
 [ 2 17]]
              precision    recall  f1-score   support

           1       0.96      0.92      0.94        48
           2       0.81      0.89      0.85        19

    accuracy                           0.91        67
   macro avg       0.88      0.91      0.89        67
weighted avg       0.91      0.91      0.91        67
```

AdaBoost (AB)

- Precision of the model for class (1) is same as that of Random Forest at (96), doing better with first class than with class (2) at 81. This algorithm is best suited for one class (1) as compared to the other (2).
- The accuracy of the model comes at 91, just a little better than RF.
- Macro and Weighted average of 91, still slightly different from what RF outputted.

Insights from ROC/AUC Curve

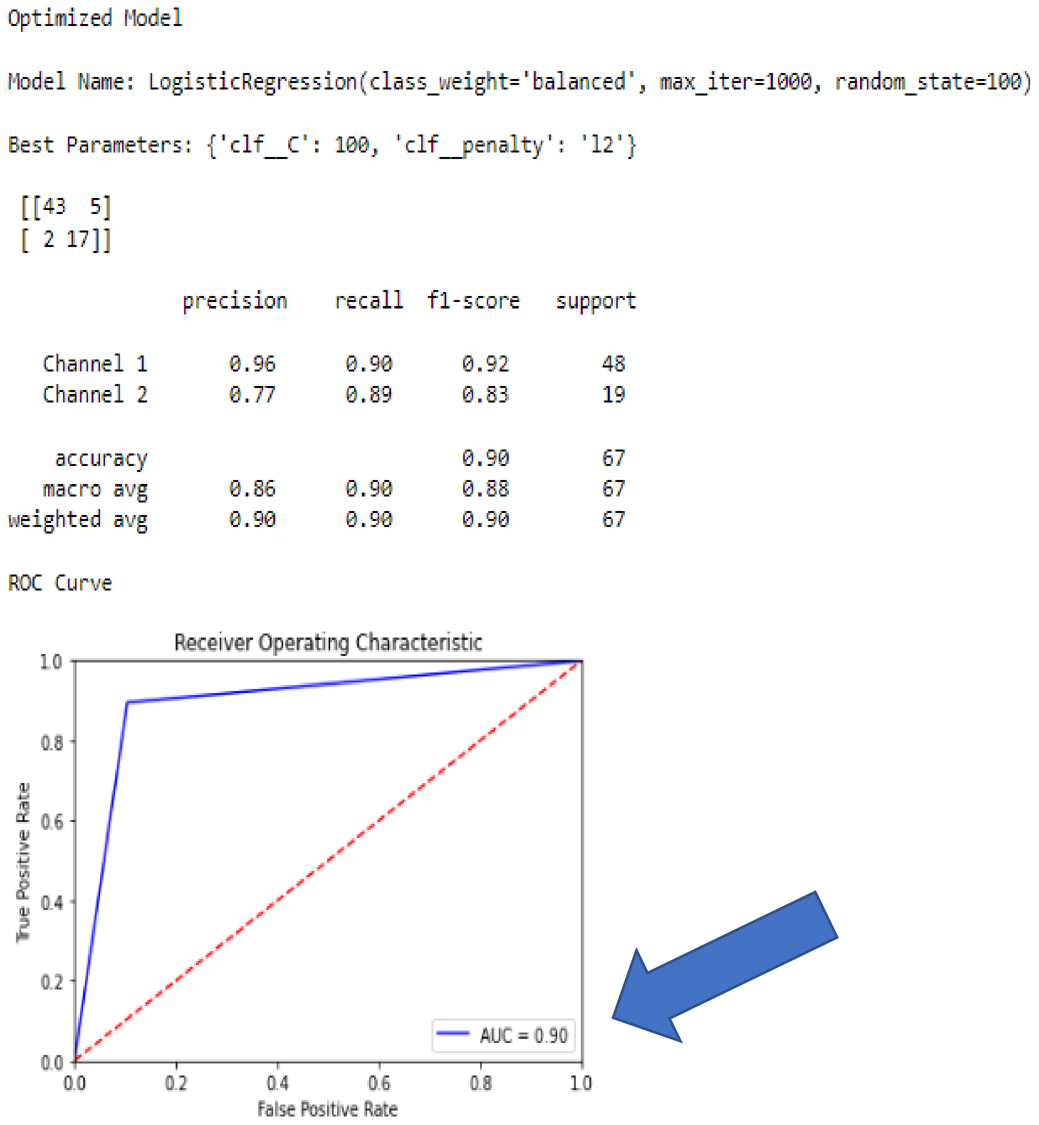
- We can see the Area under the Curve (AUC) is at 90, matching close to recall and accuracy of the optimized model



```
Optimized Model

Model Name: RandomForestClassifier(random_state=100)

Best Parameters: {'clf__bootstrap': True, 'clf__max_features': 'auto', 'clf__n_estimators': 100}

 [[43  5]
 [ 2 17]]

              precision    recall  f1-score   support

   Channel 1       0.96      0.90      0.92        48
   Channel 2       0.77      0.89      0.83        19

    accuracy                           0.90        67
   macro avg       0.86      0.90      0.88        67
weighted avg       0.90      0.90      0.90        67

ROC Curve
```

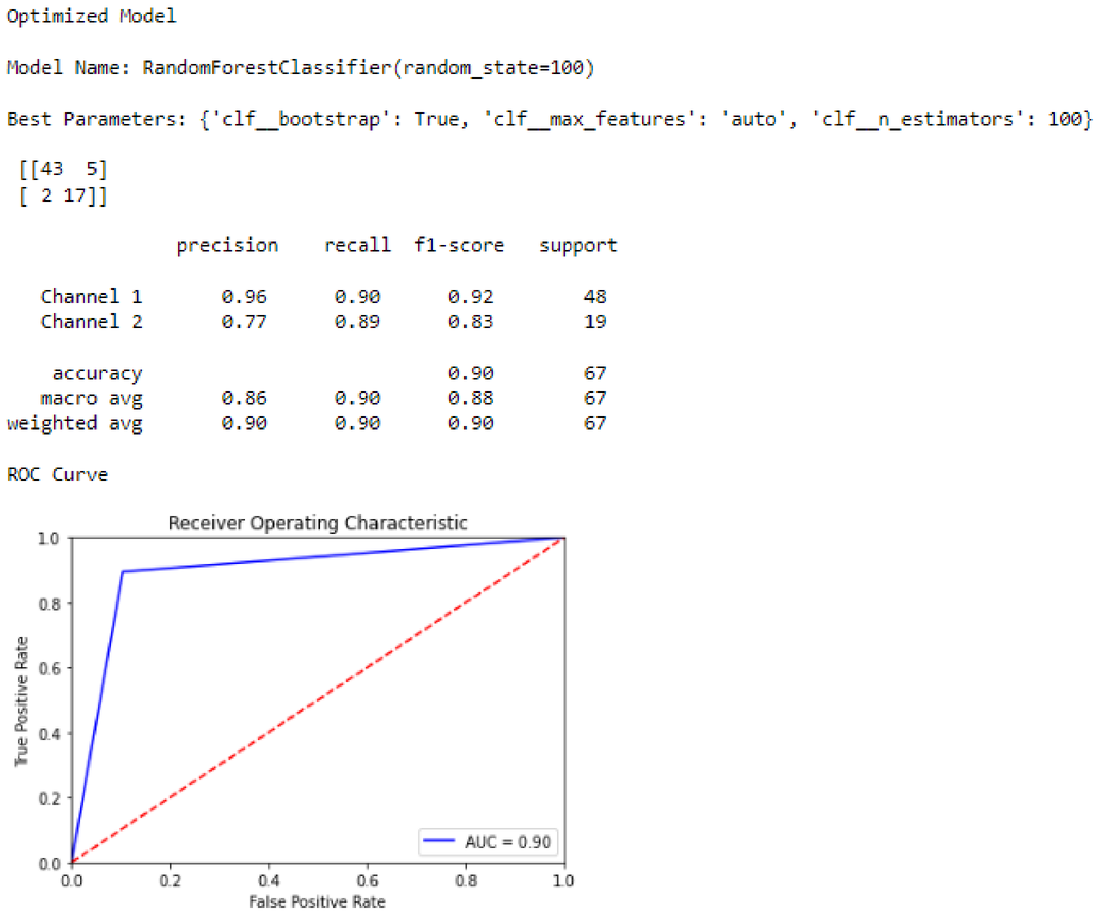
Insights from ROC/AUC Curve (cont)

- The Area under the Curve (AUC) comes at 90, close to the model’s accuracy and recall
- This is evident as the blue line indicating the curve for AUC hits above 0.8 and close to 10

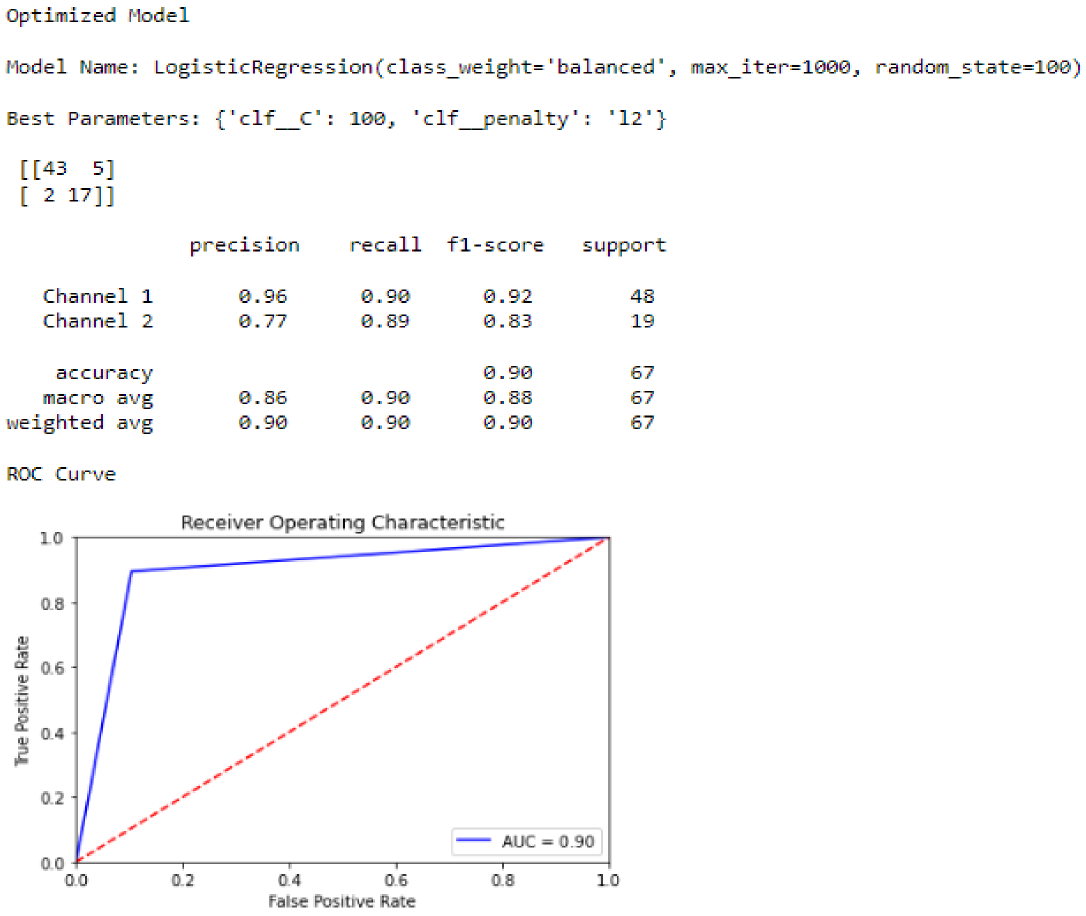


```
Optimized Model

Model Name: LogisticRegression(class_weight='balanced', max_iter=1000, random_state=100)

Best Parameters: {'clf__C': 100, 'clf__penalty': 'l2'}

 [[43  5]
 [ 2 17]]

              precision    recall  f1-score   support

   Channel 1       0.96      0.90      0.92        48
   Channel 2       0.77      0.89      0.83        19

    accuracy                           0.90        67
   macro avg       0.86      0.90      0.88        67
weighted avg       0.90      0.90      0.90        67

ROC Curve
```

After analyzing the outputs, it is safe to say there is no marginal difference nor better optimized model. Both models are giving the same precision, recall and accuracy. No better model amongst the two for use, they will do well when applied



```
Optimized Model

Model Name: RandomForestClassifier(random_state=100)

Best Parameters: {'clf__bootstrap': True, 'clf__max_features': 'auto', 'clf__n_estimators': 100}

 [[43  5]
 [ 2 17]]

              precision    recall  f1-score   support

   Channel 1       0.96      0.90      0.92        48
   Channel 2       0.77      0.89      0.83        19

    accuracy                           0.90        67
   macro avg       0.86      0.90      0.88        67
weighted avg       0.90      0.90      0.90        67

ROC Curve
```

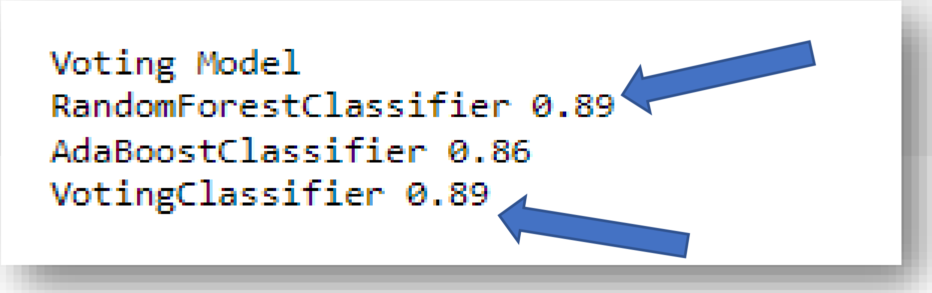


```
Optimized Model

Model Name: LogisticRegression(class_weight='balanced', max_iter=1000, random_state=100)

Best Parameters: {'clf__C': 100, 'clf__penalty': 'l2'}

 [[43  5]
 [ 2 17]]

              precision    recall  f1-score   support

   Channel 1       0.96      0.90      0.92        48
   Channel 2       0.77      0.89      0.83        19

    accuracy                           0.90        67
   macro avg       0.86      0.90      0.88        67
weighted avg       0.90      0.90      0.90        67

ROC Curve
```

Ensemble Voting Model

With all the models being used and as a matter of classification, we are always in search for the better model and how to ameliorate failed or successful performances. The need for a better classification model to predict the proper channel classification, led to utilizing the ensemble voting model.

Conclusion



```
Voting Model
RandomForestClassifier 0.89
AdaBoostClassifier 0.86
VotingClassifier 0.89
```

- The models performed well with the dataset, but I believe it can do better with larger datasets and would be interested in running the test on it. Predicting was well done by Random Forest, and I would recommend this algorithm to be used for now and future classification dataset problems.
- Random Forest performed better even when optimized
- Random Forest prevails 89 while AdaBoost at 86
- Random forest boxplot was stable with an average of 92 and no outlier

Recommendations

We could use more training samples to build up the model

Without a doubt we are working with a small dataset, it would be good to get more data (historical data) and re-run the models for better results. Since as most of the models work best with bigger datasets