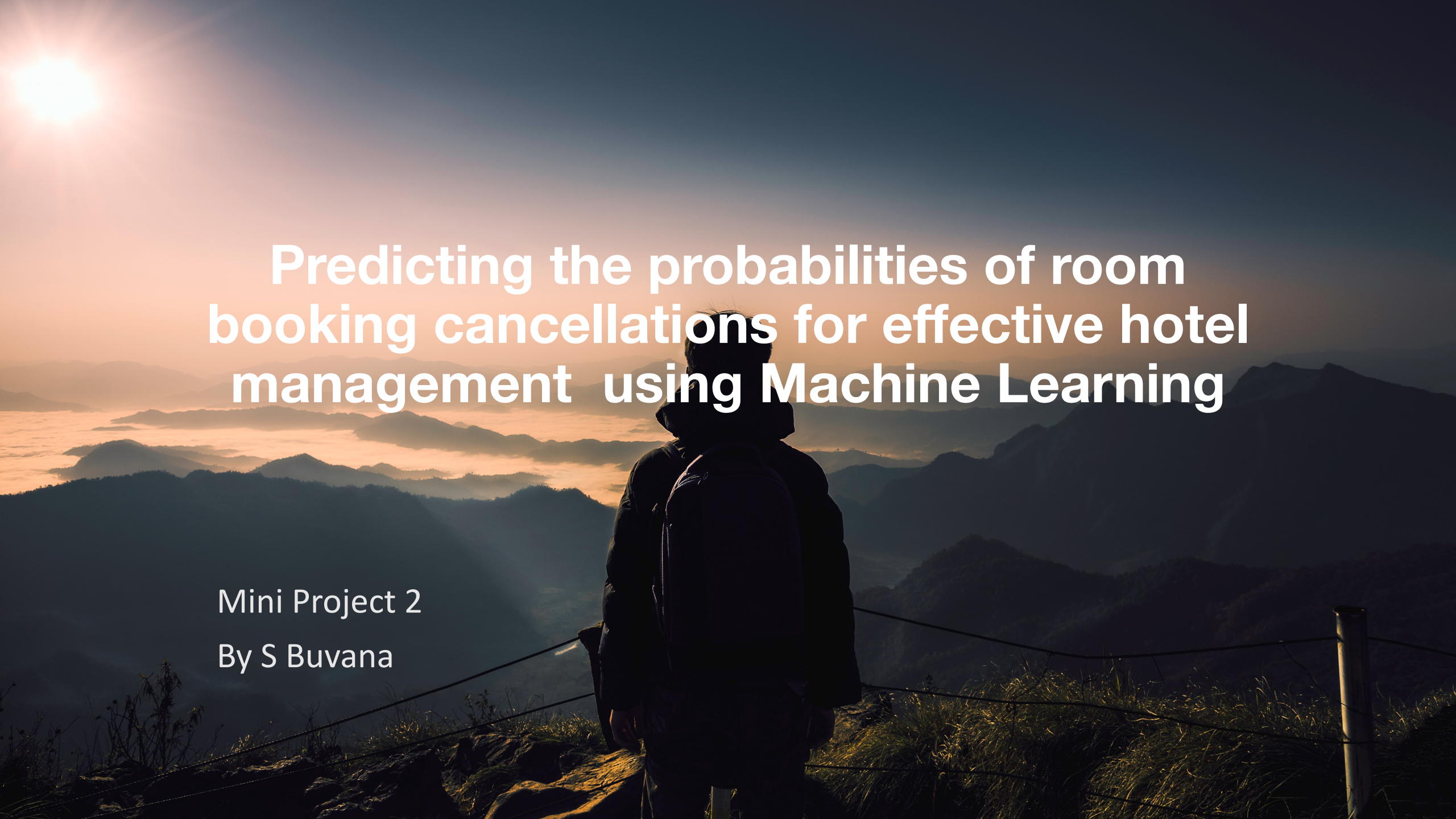


Predicting the probabilities of room booking cancellations for effective hotel management using Machine Learning

Mini Project 2

By S Buvana

Recap to Dataset: Hotel Bookings Demand

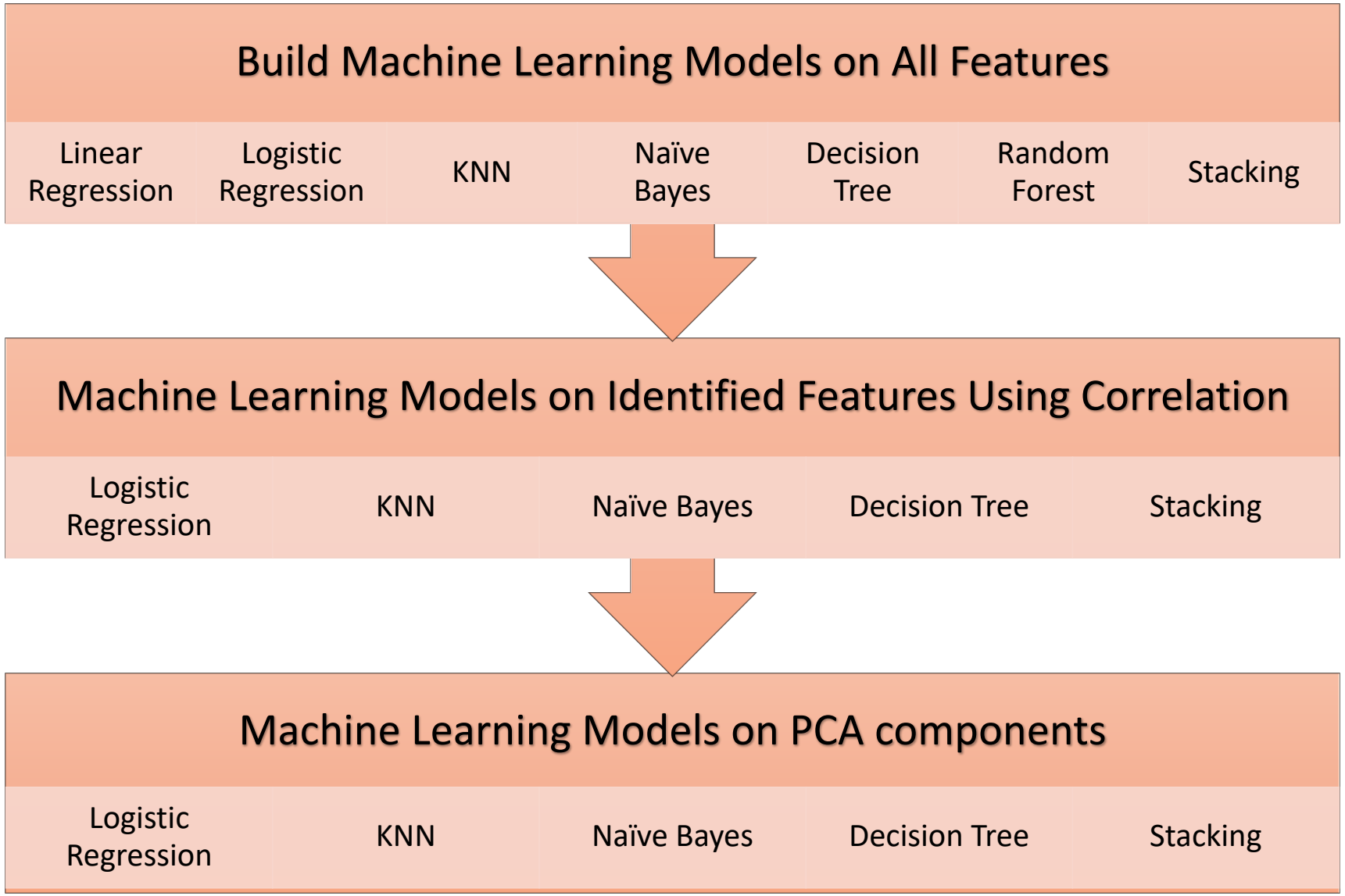
- Dataset contains data about two Hotels; **Resort and City Hotel** based at Portugal
  - Key information such as the types of hotel guests, arrival information, duration of stays, cancellation rates, allocation of room types and lead time
- Dataset entails the information of hotel guests who have booked their stays from **July 2015 to Aug 2017.**

Problem Statement

Predict the probabilities of cancellations to allow for hotel management to plan for future booking strategies.

- Aids hotel operators to better manage their resources by predicting the cancellation rates of their reservations.

Iterative Process



Pre-processing of Data

Encoding of Data:27 Columns were transformed to 559 columns.

```
Hotel_bookings_with_dummies = pd.get_dummies(data = Hotel_bookings, columns = ['hotel','arrival_date_month', 'arriv
                  'distribution_channel', 'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_ty
                                        prefix = ['hotel','arrival month', 'arrival year','meal','market segment',
                  'distribution channel', 'reserved room', 'assigned room', 'deposit', 'customer','lead'] )
Hotel_bookings_with_dummies.head()
```

	is_canceled	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	is_repeated_guest	previous_cancella
0	0	27	1	0	0	2	0	
1	0	27	1	0	0	2	0	
2	0	27	1	0	1	1	0	
3	0	27	1	0	1	1	0	
4	0	27	1	0	2	2	0	

Objectives

- To determine if feature engineering affects the Models' predictive performances.
- Evaluate Model Results to configure the best ML technique to resolve the problem statement.

Salient Points to Note

- To keep uniformity across all the models testing, the cross validation folds were kept to 10 folds.
- Train -Test Splits : 80%-20%.

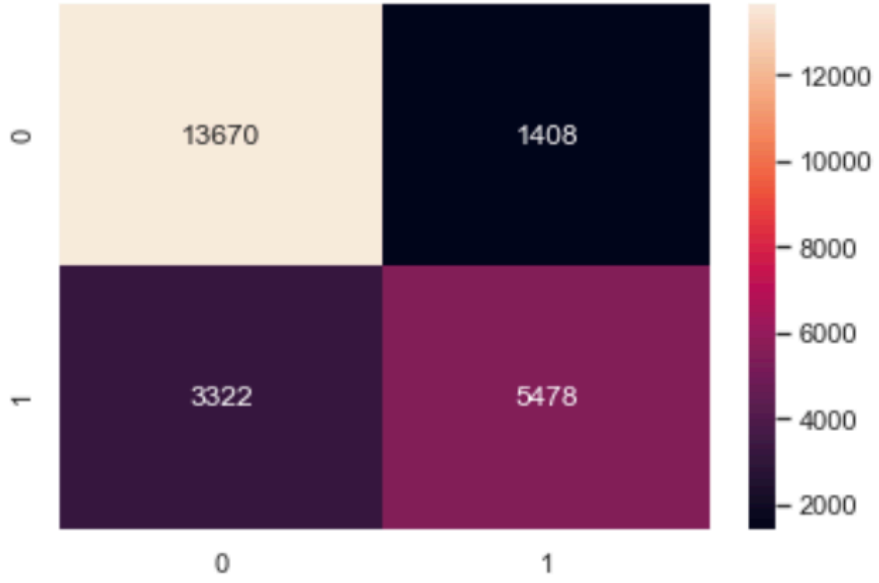
ML Techniques

On all features

Logistic Regression: Performance

- Cross Validation Accuracy Score: 80%

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
0	0.80	0.91	0.85	15078
1	0.80	0.62	0.70	8800
accuracy			0.80	23878
macro avg	0.80	0.76	0.78	23878
weighted avg	0.80	0.80	0.80	23878

- AUC scores for the case for both train and test sets are 0.88, representing a good classifier. Overfitting is unlikely.
- GridSearch CV Optimisation- Value remains at 80%

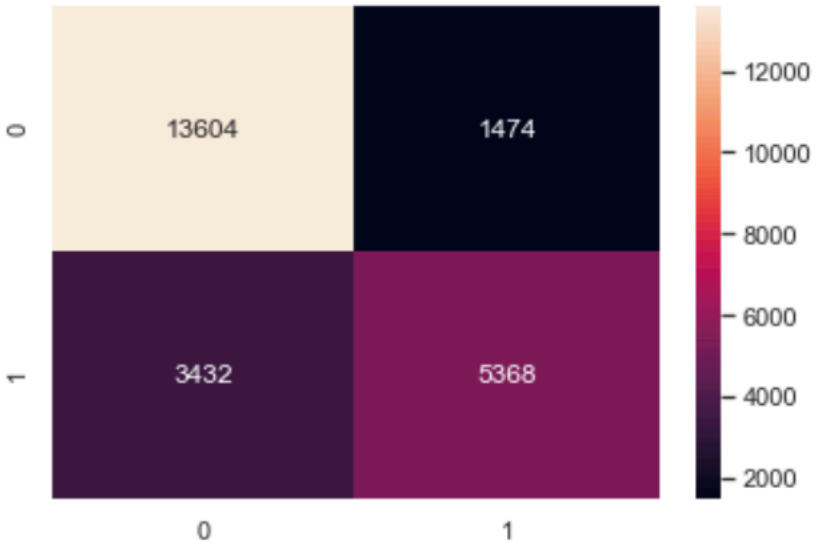
KNN Classifier- Performance

- Cross Validation Accuracy Score: 79.4%
- Generally classifies data correctly

Classification Report

	precision	recall	f1-score	support
0	0.80	0.90	0.85	15078
1	0.78	0.61	0.69	8800
accuracy			0.79	23878
macro avg	0.79	0.76	0.77	23878
weighted avg	0.79	0.79	0.79	23878

Confusion Matrix



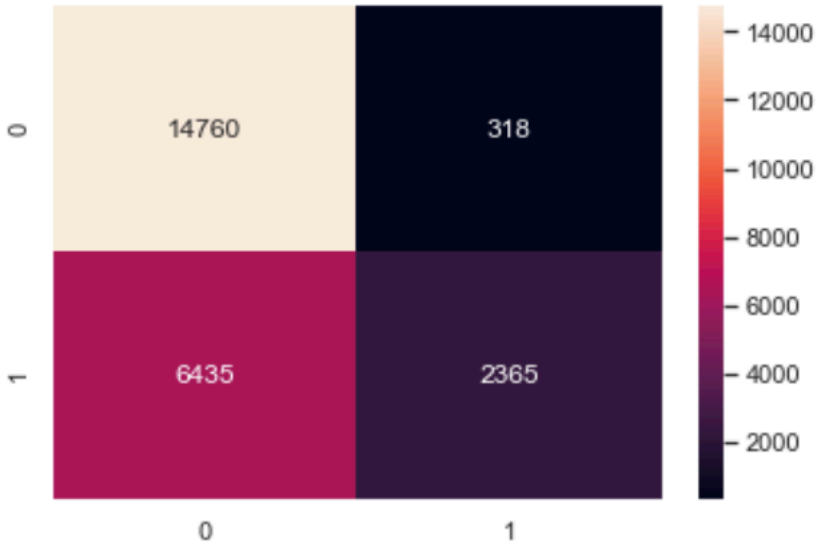
Evaluation

- Random initialisation of n=4
- Evaluation of accuracy score (testing data): 75.1%. Overfitting is likely.
- With hyper parameter tuning GridSearch CV accuracy values is optimised to 83.4%

Naïve Bayes- Performance

- Cross Validation Accuracy Score: 71.9%

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
0	0.70	0.98	0.81	15078
1	0.88	0.27	0.41	8800
accuracy			0.72	23878
macro avg	0.79	0.62	0.61	23878
weighted avg	0.76	0.72	0.67	23878

Evaluation

- Cross Validation accuracy for testing data : 70.6%
- Generally a poor classifier
- Naïve Bayes holds a major assumption that features in a class is independent

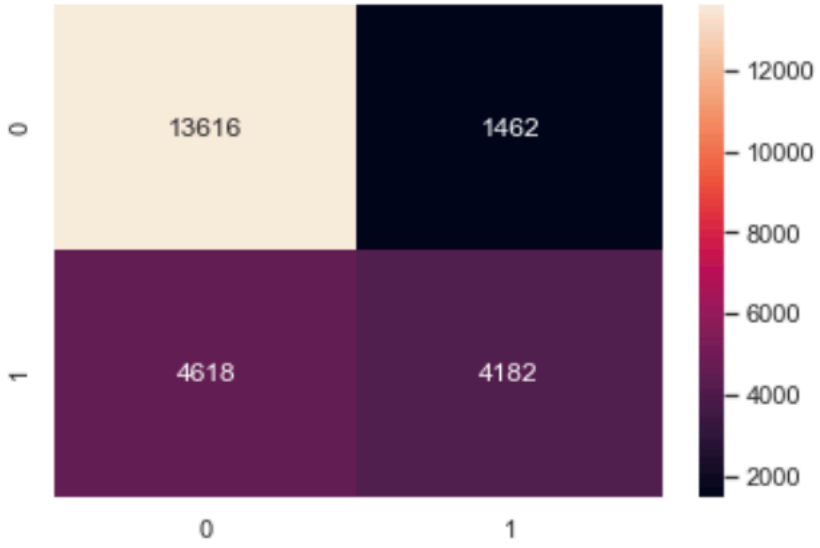
Decision Tree: Performance

- Cross Validation Accuracy Score: 74.5%

Classification Report

	precision	recall	f1-score	support
0	0.75	0.90	0.82	15078
1	0.74	0.48	0.58	8800
accuracy			0.75	23878
macro avg	0.74	0.69	0.70	23878
weighted avg	0.74	0.75	0.73	23878

Confusion Matrix



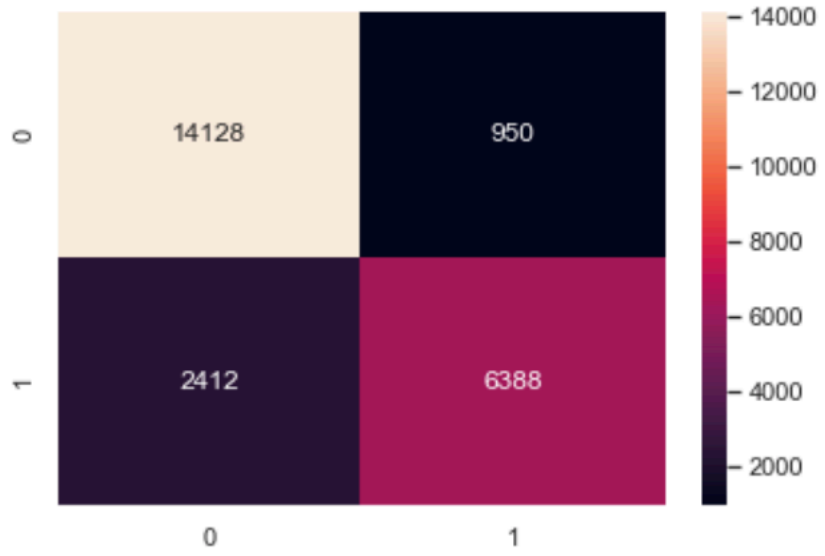
Evaluation

- Cross Validation accuracy for testing data : 72.3%
- Overfitting is likely
- With hyper parameter tuning GridSearch CV accuracy values is optimised to 81.5%

Random Forest: Performance

- Cross Validation Accuracy Score: 85.7%
- Well trained model thus far

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
0	0.85	0.94	0.89	15078
1	0.87	0.73	0.79	8800
accuracy			0.86	23878
macro avg	0.86	0.83	0.84	23878
weighted avg	0.86	0.86	0.86	23878

Evaluation

- Random Forest performs better than decision tree. They consist of multiple decision trees each based on random sample of training data

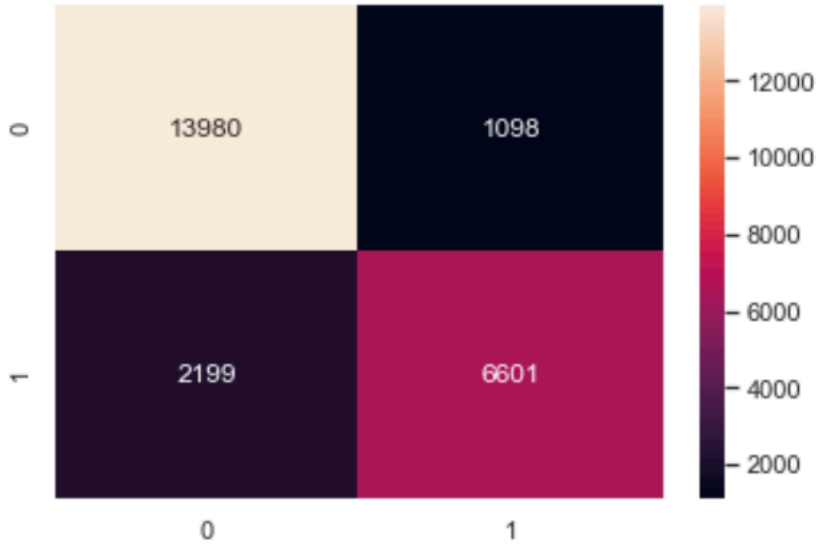
Stacking Performance

- Cross Validation Accuracy Score: 85.5%

Classification Report

	precision	recall	f1-score	support
0	0.86	0.93	0.89	15078
1	0.86	0.75	0.80	8800
accuracy			0.86	23878
macro avg	0.86	0.84	0.85	23878
weighted avg	0.86	0.86	0.86	23878

Confusion Matrix



Evaluation

- Predictive performance almost similar to Random Forest Technique

Machine Models -Predictive Performances

Training Models	Linear Regression	Logistic Regression	KNN Classifier	Naïve Bayes	Decision Tree	Random Forest	Stacking
CV Accuracy Scores (Training Set- With dummy variables 558 columns)	35%	80%	79.4%	71.9%	83.2%	85.7%	85.5%

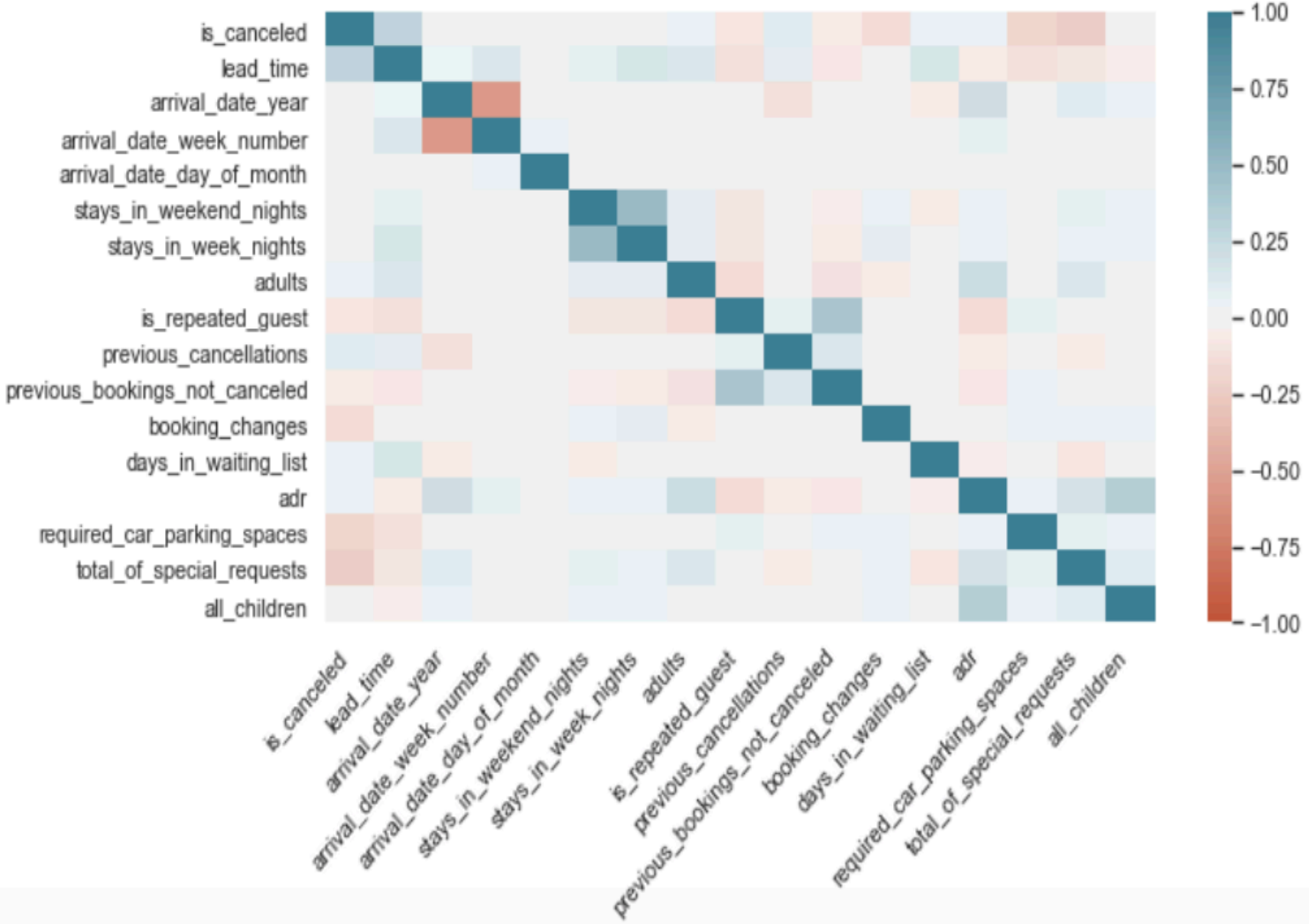
Feature Selection

Identification of relevant features

Feature Selection

- We would need to perform machine learning models to train the dataset with the key features.
- Removing the noisy features will help with memory, computational cost and the accuracy of our model. This might prevent overfitting of the models as well.
- Re-run of ML Models on 5 identified variables. Identify important features using Correlation and PCA to compare the models' results.

Correlation Heatmap



- Identified 5 features as X:

1. Lead Time
2. Previous Cancellations
3. Adults
4. Days in waiting List
5. Stays in weeknights

- Evaluation of models' performances against previous metrics

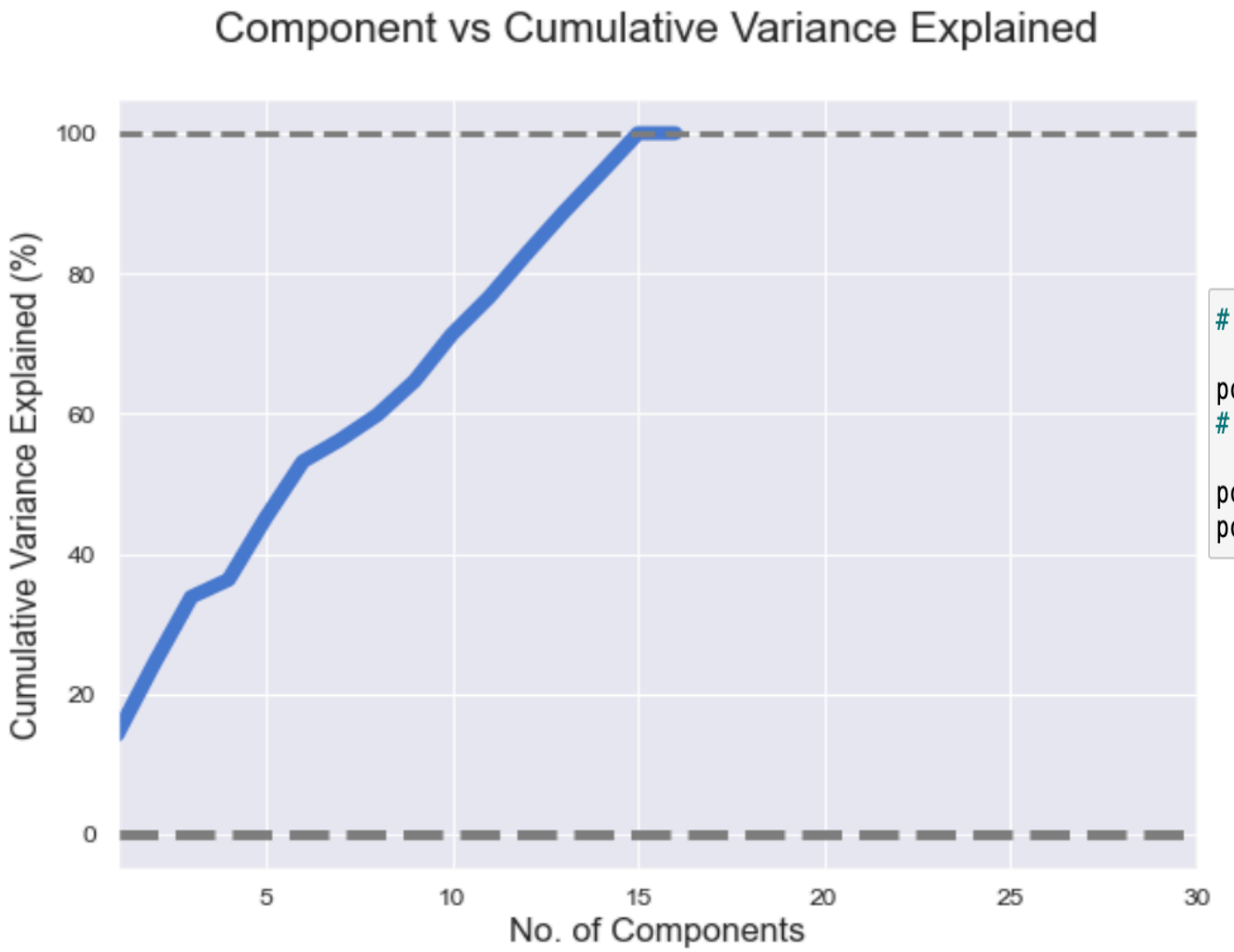
Summary- Performance Metrics

Training Models6.8%	Logistic Regression	KNN Classifier	Naïve Bayes	Decision Tree	Random Forest	Stacking
CV Accuracy Scores (Training Sets)– 558 columns Dummy Variables	80.0%	79.4%	71.9%	83.2%	85.7%	85.5%
CV Accuracy Scores (Training Sets)– Identified 5 features using Correlation	68.2%	72.3%	68.1%	74.6%	74.4%	74.4%

Feature selection

With PCA Dataset

PCA Analysis- Pre Processing of Data




```
# Apply dimensionality reduction to Xs using transform

pca1 = PCA(n_components=15)
# Fit Xs

pca1.fit(Xs)
pca_X = pca1.fit_transform(Xs)
```

- 15 PCA components are chosen reduced dimensional dataset

Random Forest - Performance with PCA

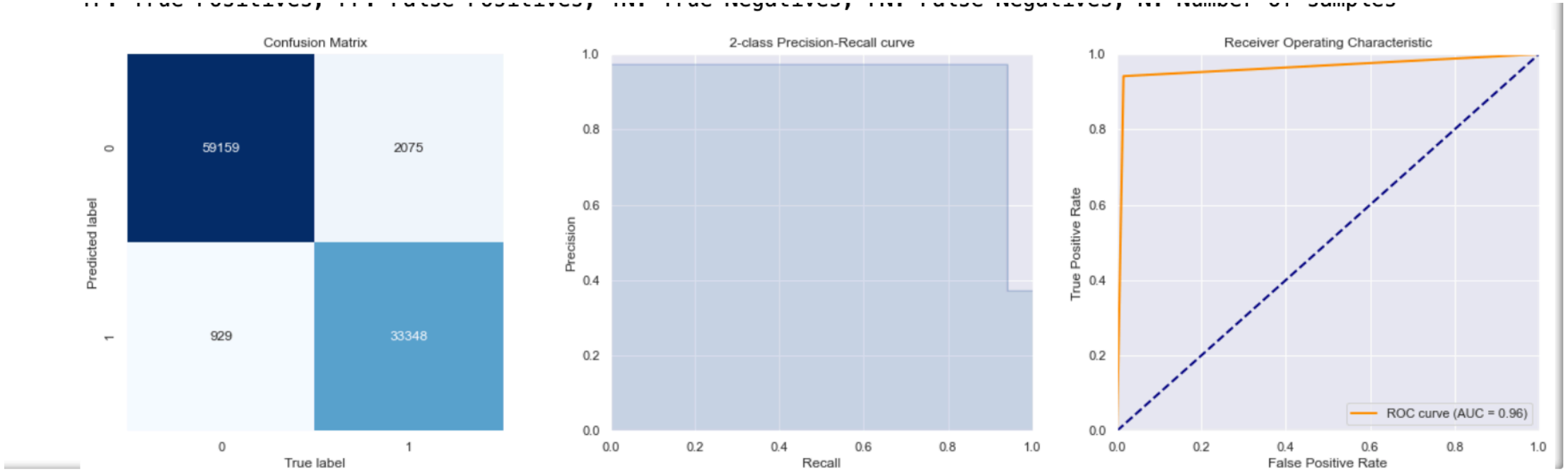
Results-Training Set

- Cross Validation Accuracy Score: 81.4%

```
******************
* Random Forest *
******************
Accuracy : 0.9685 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.9729 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.9414 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.9630                                                                     Best: 1, Worst: < 0.5
--------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```

Random Forest- Performance with PCA

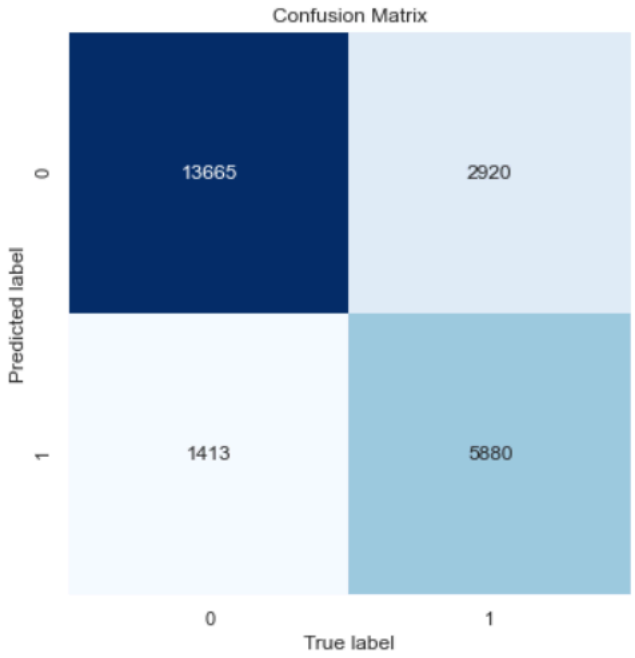
Confusion Matrix | Precision-Recall Curve | ROC Curve

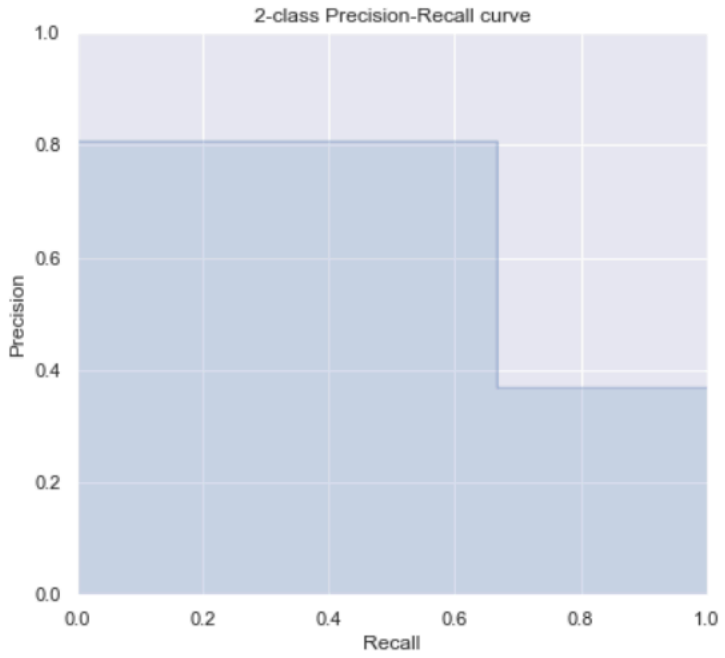


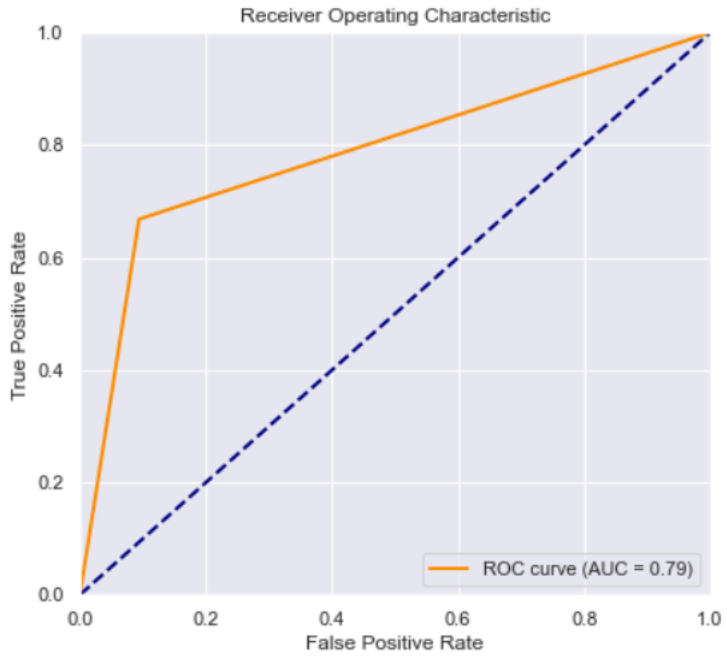
Random Forest: Performance with PCA

Results-Testing Set

```
Accuracy : 0.8185 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.8063 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.6682 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.7872                                                                     Best: 1, Worst: < 0.5
------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```







Stacking- Performance with PCA

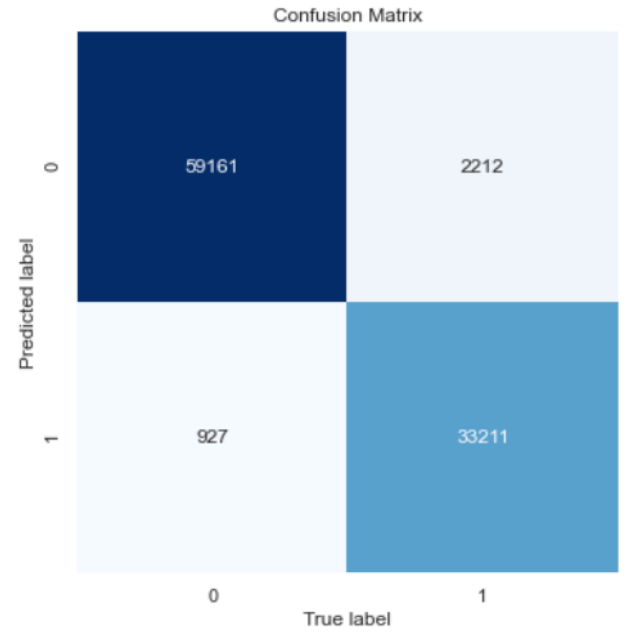
Results-Training Set

- Cross Validation Accuracy Score: 81.8%

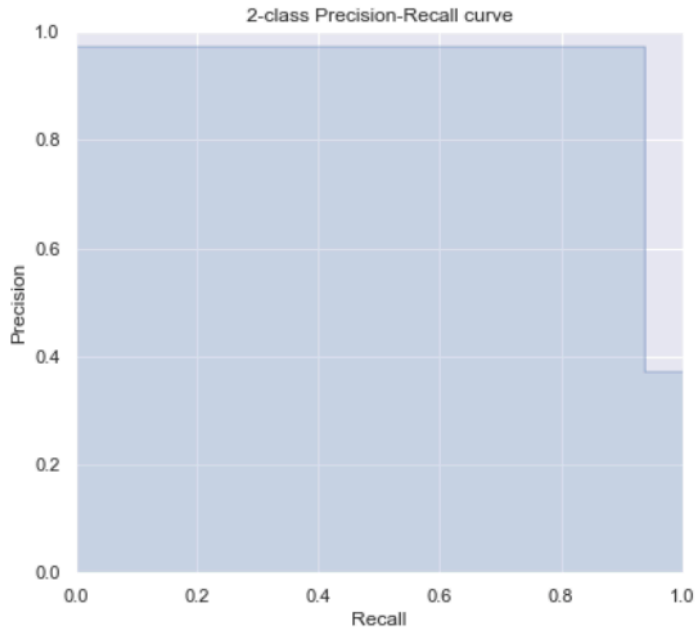
```
*************
* Stacking *
*************
Accuracy : 0.9671 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.9728 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.9376 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.9611                                                                     Best: 1, Worst: < 0.5
--------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```

Decision Tree - Performance with PCA

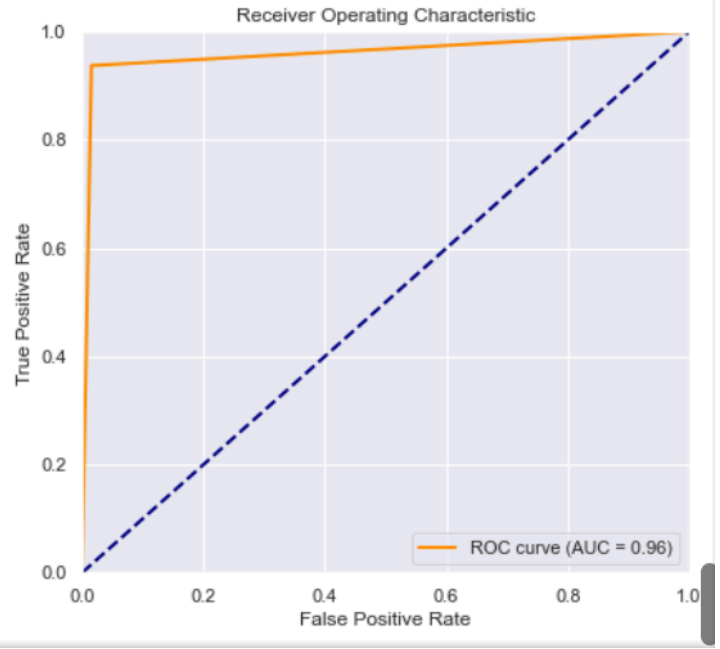
Confusion Matrix



Precision-Recall Curve



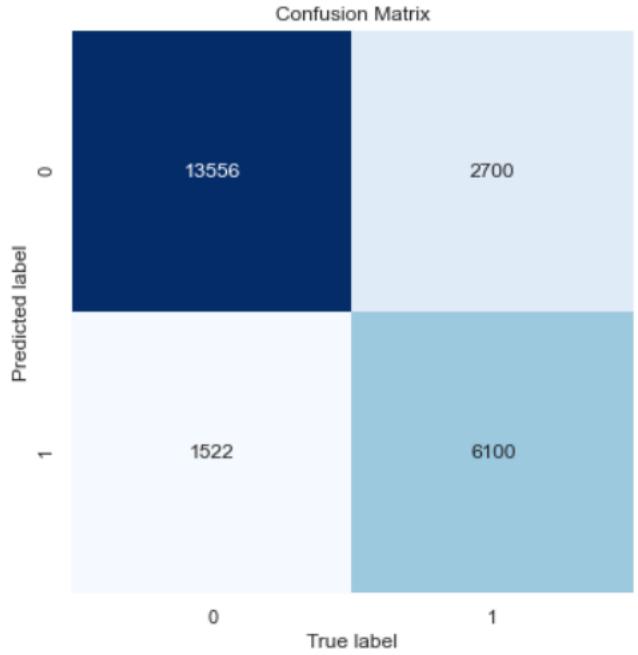
ROC Curve

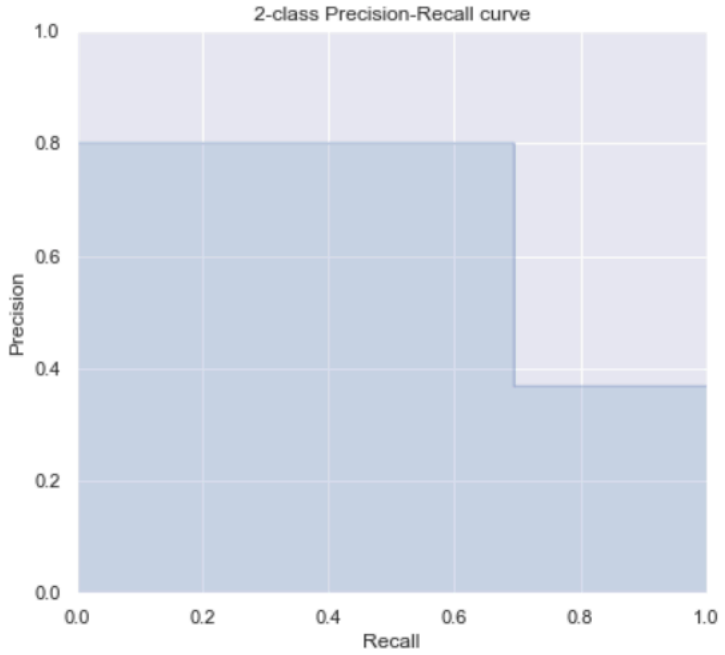


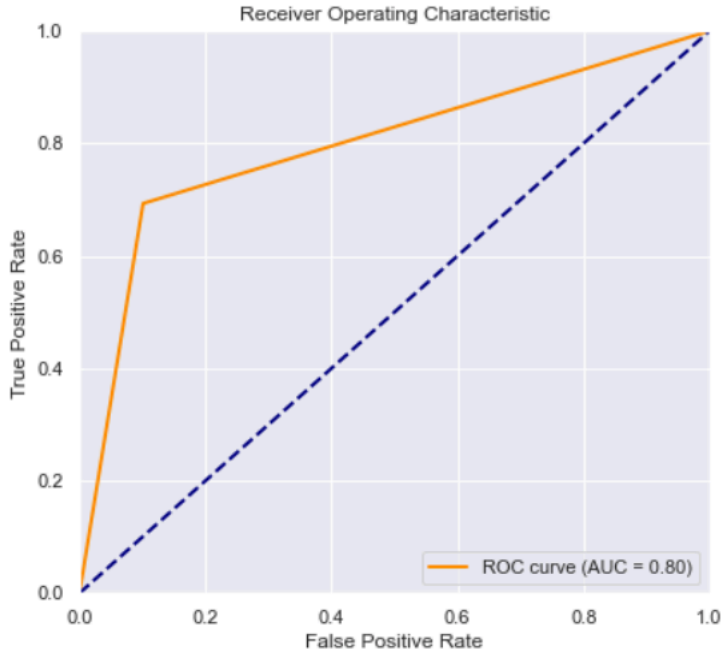
Stacking-Performance with PCA

Results-Testing Set

```
Accuracy : 0.8232 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.8003 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.6932 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.7961                                                                     Best: 1, Worst: < 0.5
----------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```







Summary- Predictive Performance Metrics

Training Models6.8%	Logistic Regression	KNN Classifier	Naïve Bayes	Decision Tree	Random Forest	Stacking
CV Accuracy Scores (Training Sets)– 558 columns Dummy Variables	80.0%	79.4%	71.9%	83.2%	85.7%	85.5%
CV Accuracy Scores (Training Sets)– Identified 5 features using Correlation	68.2%	72.3%	68.1%	74.6%	74.4%	74.4%
CV Accuracy Scores (Training Sets)- 15 PCA components	79.8%	79.8%	68.3%	78.5%	81.4%	81.8%

Conclusion

Evaluation of Overall Results

- Generally overfitting of ML models-algorithms on testing datasets, most of the models perform poorly.
- Overfitting implies that the models are well on the training data but has poor performance when applied to new data.
- Generally, models predictive performances are better when the whole dataset is applied without feature selection.
- Could work to gather more datasets of Hotel Bookings from more hotels to gather insightful data and for better predictive models

Conclusion

- Based on the data collected, we can use Machine Learning model — **Random Forest** to help with the booking cancellation prediction. It provide the best accuracy at the rate of **85.7%.**
- Based on the insight obtained from the above, hotel management can use it to derive different mitigation plans to help minimize the risk of booking cancellation and increase their business revenue generation.


Thank you