Airline Passenger Satisfaction

CS513 Project Presentation

Section B Group 6

Shivang Medhekar
CWID: 10478510

Mingfang Liang
CWID: 10473184

Yutong Wei
CWID: 2005208

Jiani Yu
CWID: 10474729

Problem Statement

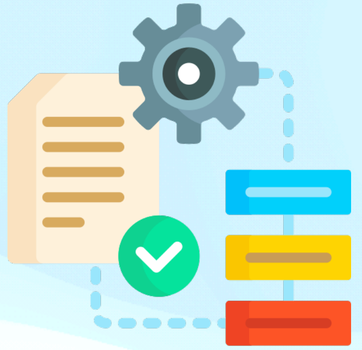
- Airline Market
  - 5000 airlines
  - $785.6 bn
- Airline Challenges
  - Many Features
  - What to improve ?
  - How to differentiate ?

Work Flow

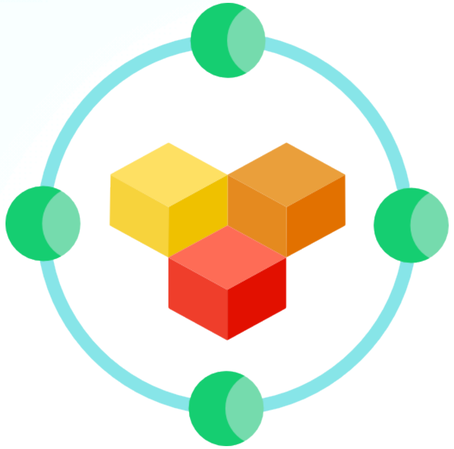
Data Exploration

Data Analysis

Data Cleaning



Data Preprocessing



Modeling

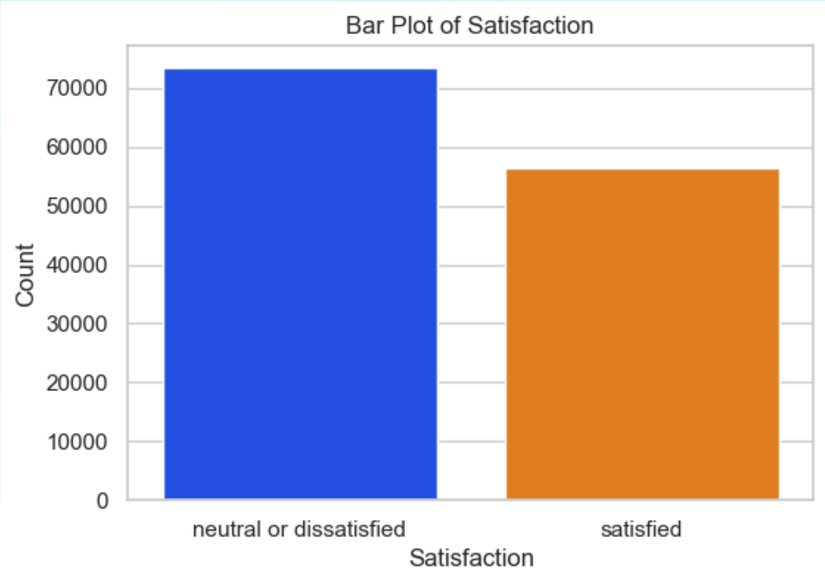
Evaluation

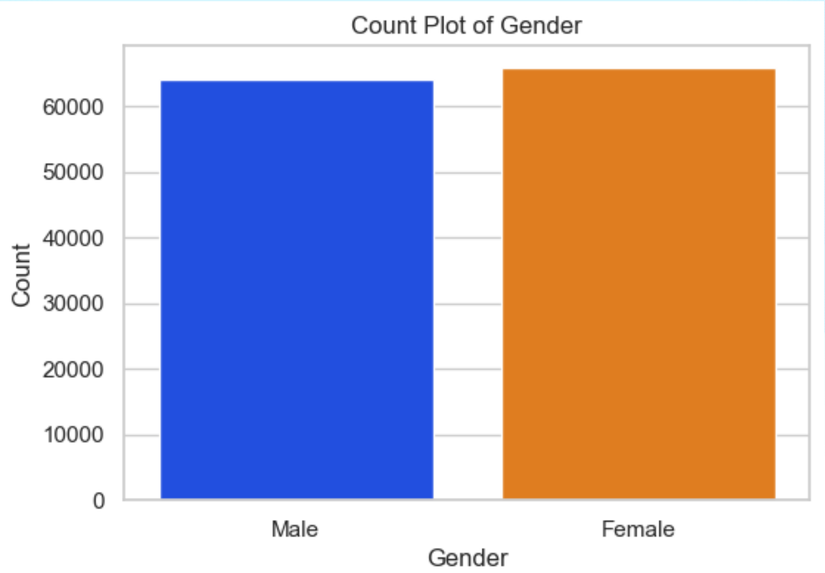
Dataset Exploration

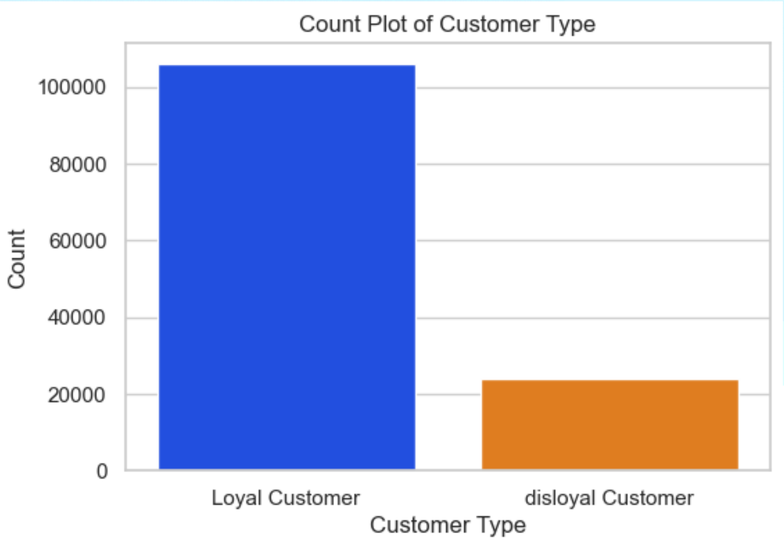
- **Gender**: Gender of the passengers (Female, Male)
- **Customer Type**: The customer type (Loyal customer, disloyal customer)
- **Age**: The actual age of the passengers
- **Type of Travel**: Purpose of the flight of the passengers (Personal Travel, Business Travel)
- **Class**: Travel class in the plane of the passengers (Business, Eco, Eco Plus)
- **Flight distance**: The flight distance of this journey
- **Inflight wifi service**: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5)
- **Departure/Arrival time convenient**: Satisfaction level of Departure/Arrival time convenient
- **Ease of Online booking**: Satisfaction level of online booking
- **Gate location**: Satisfaction level of Gate location
- **Food and drink**: Satisfaction level of Food and drink
- **Online boarding**: Satisfaction level of online boarding
- **Seat comfort**: Satisfaction level of Seat comfort
- **Inflight entertainment**: Satisfaction level of inflight entertainment
- **On-board service**: Satisfaction level of On-board service
- **Leg room service**: Satisfaction level of Leg room service
- **Baggage handling**: Satisfaction level of baggage handling
- **Check-in service**: Satisfaction level of Check-in service
- **Inflight service**: Satisfaction level of inflight service
- **Cleanliness**: Satisfaction level of Cleanliness
- **Departure Delay in Minutes**: Minutes delayed when departure
- **Arrival Delay in Minutes**: Minutes delayed when Arrival
- **Satisfaction**: Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

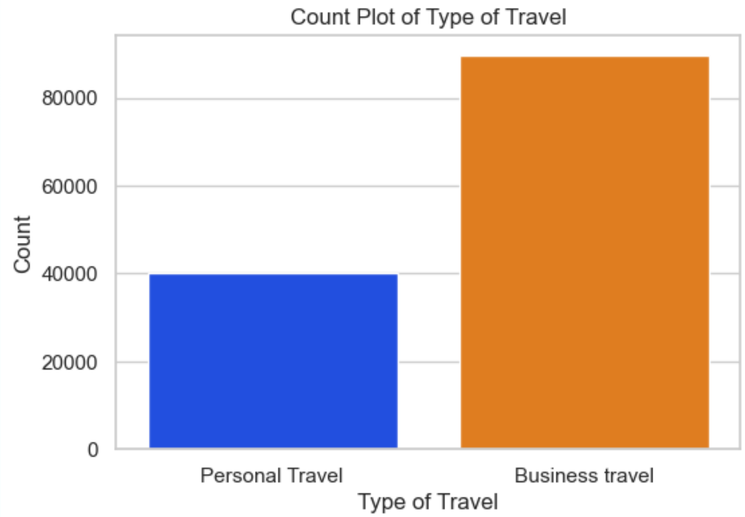
- Dataset: airline passenger satisfaction survey.
- 23 Variables and 129,880 Observations

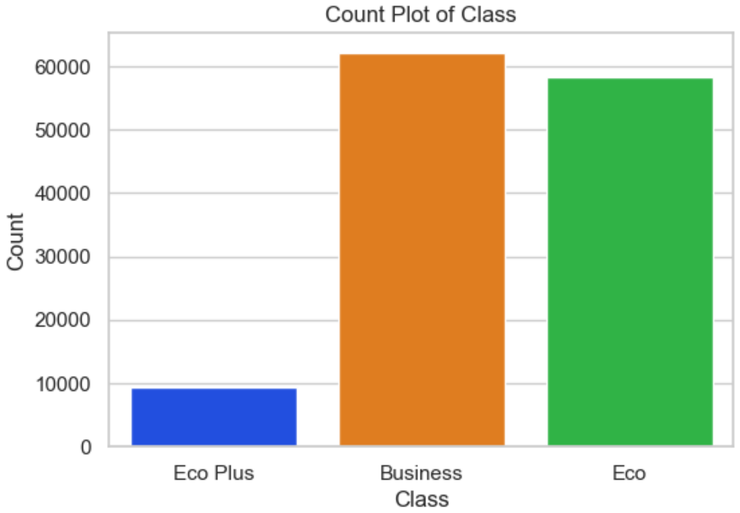
Data Analysis: Categorical Data





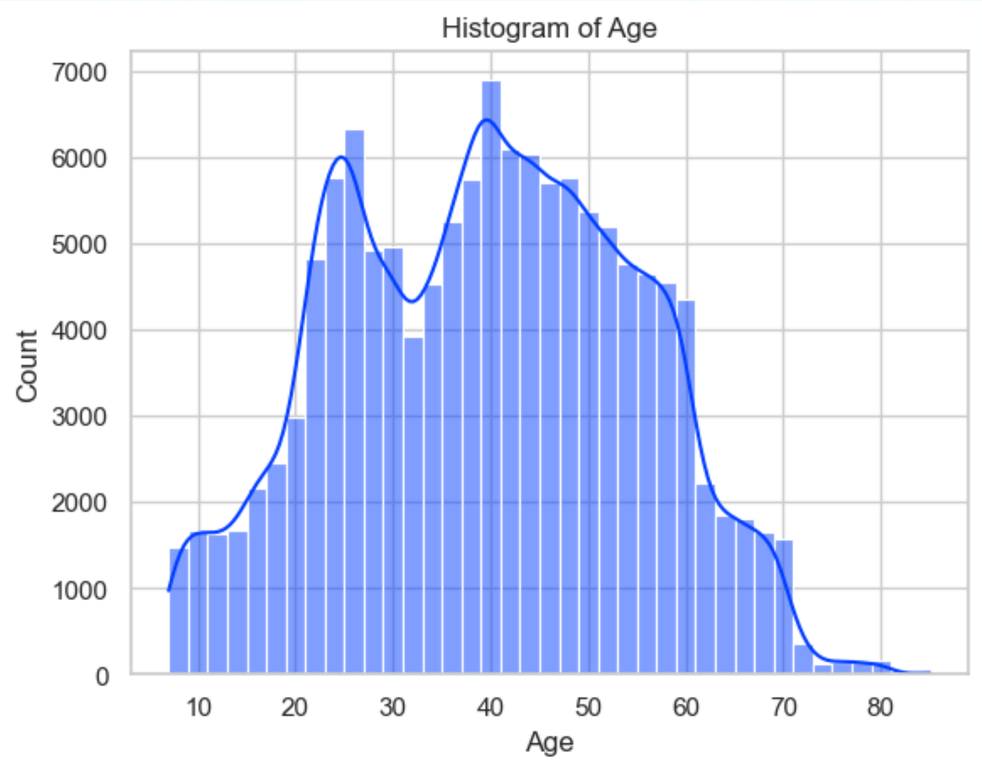


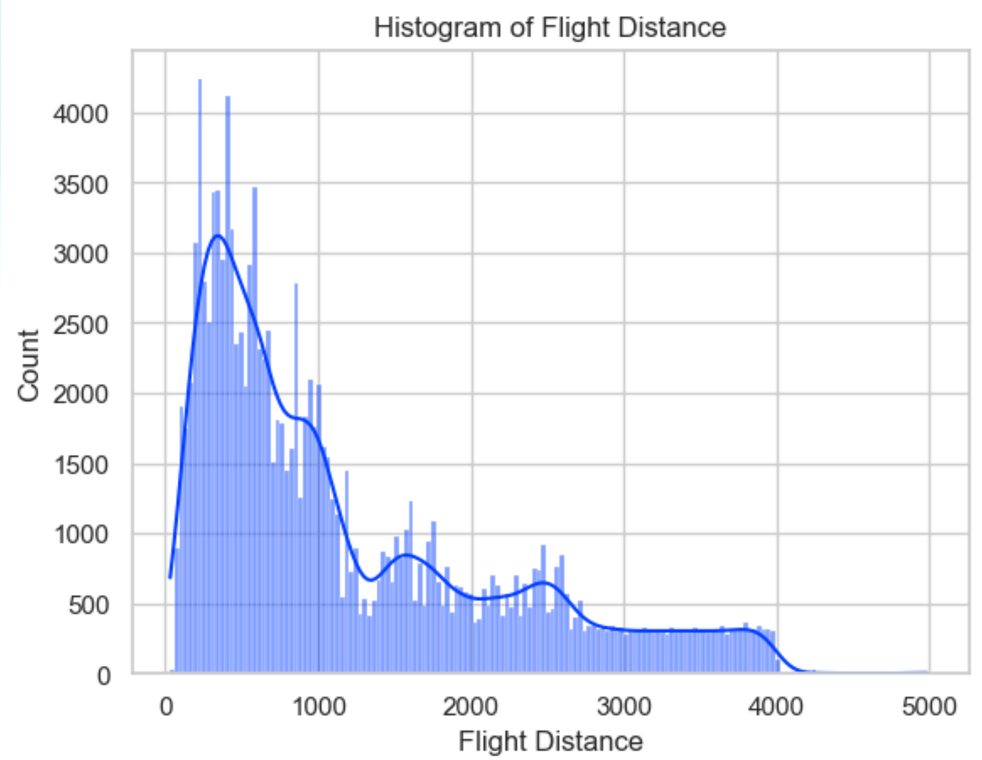


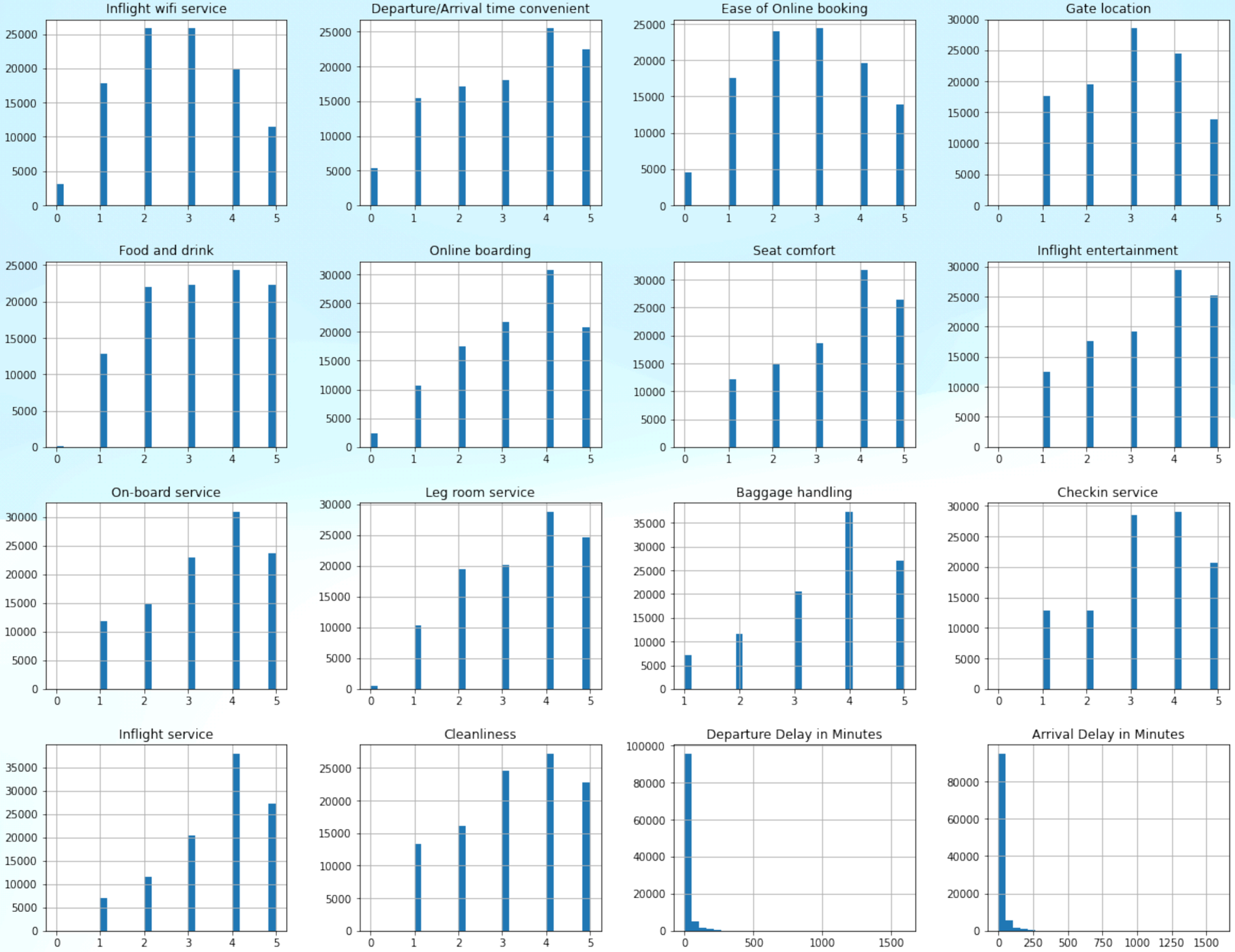


Data Analysis

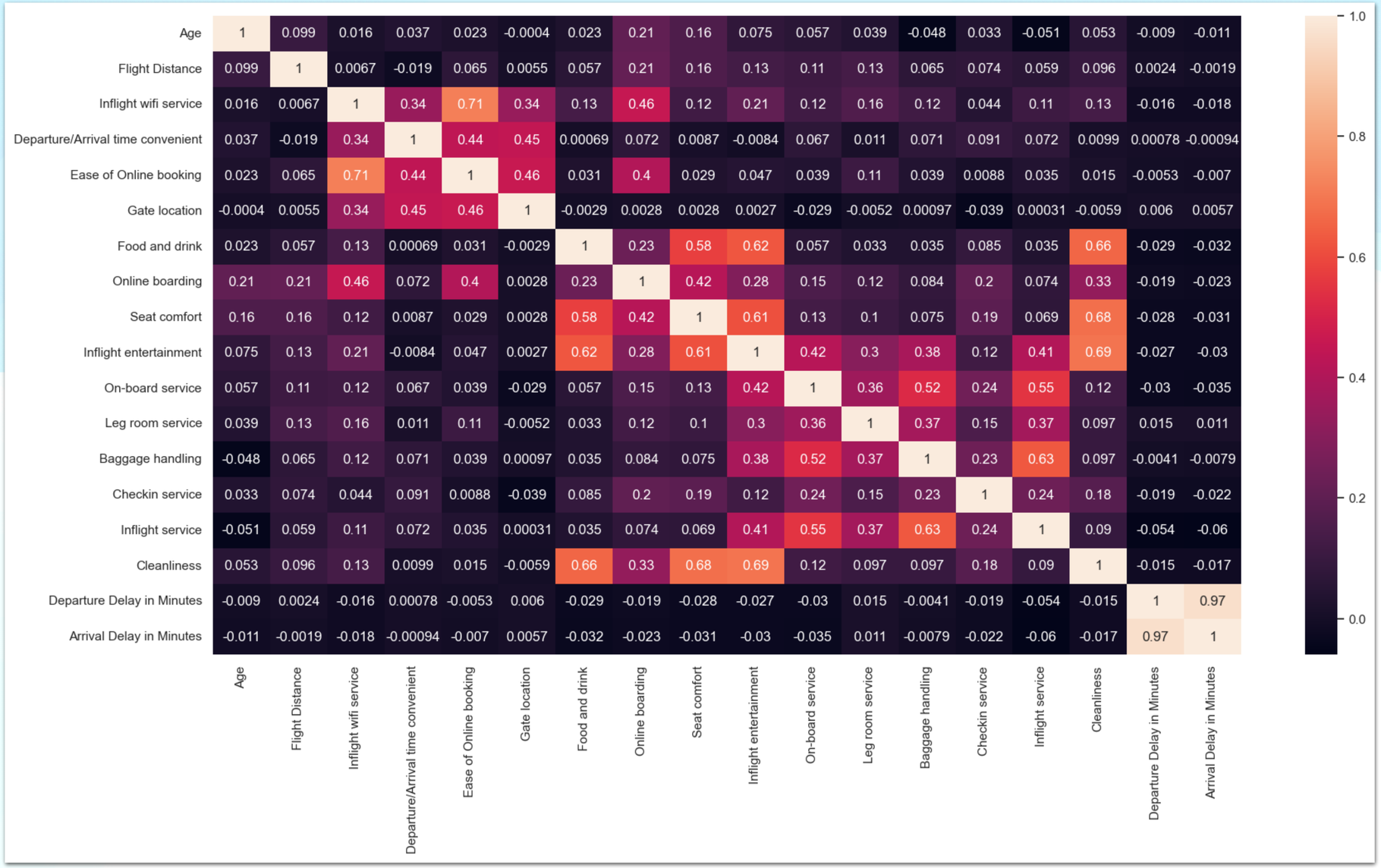
Numerical Data






Inflight wifi service
Departure/Arrival time convenient
Ease of Online booking
Gate location
Food and drink
Online boarding
Seat comfort
Inflight entertainment
On-board service
Leg room service
Baggage handling
Checkin service
Inflight service
Cleanliness
Departure Delay in Minutes
Arrival Delay in Minutes

Heat Map


Age 1 0.099 0.016 0.037 0.023 -0.0004 0.023 0.21 0.16 0.075 0.057 0.039 -0.048 0.033 -0.051 0.053 -0.009 -0.011
Flight Distance 0.099 1 0.0067 -0.019 0.065 0.0055 0.057 0.21 0.16 0.13 0.11 0.13 0.065 0.074 0.059 0.096 0.0024 -0.0019
Inflight wifi service 0.016 0.0067 1 0.34 0.71 0.34 0.13 0.46 0.12 0.21 0.12 0.16 0.12 0.044 0.11 0.13 -0.016 -0.018
Departure/Arrival time convenient 0.037 -0.019 0.34 1 0.44 0.45 0.00069 0.072 0.0087 -0.0084 0.067 0.011 0.071 0.091 0.072 0.0099 0.00078 -0.00094
Ease of Online booking 0.023 0.065 0.71 0.44 1 0.46 0.031 0.4 0.029 0.047 0.039 0.11 0.039 0.0088 0.035 0.015 -0.0053 -0.007
Gate location -0.0004 0.0055 0.34 0.45 0.46 1 -0.0029 0.0028 0.0028 0.0027 -0.029 -0.0052 0.00097 -0.039 0.00031 -0.0059 0.006 0.0057
Food and drink 0.023 0.057 0.13 0.00069 0.031 -0.0029 1 0.23 0.58 0.62 0.057 0.033 0.035 0.085 0.035 0.66 -0.029 -0.032
Online boarding 0.21 0.21 0.46 0.072 0.4 0.0028 0.23 1 0.42 0.28 0.15 0.12 0.084 0.2 0.074 0.33 -0.019 -0.023
Seat comfort 0.16 0.16 0.12 0.0087 0.029 0.0028 0.58 0.42 1 0.61 0.13 0.1 0.075 0.19 0.069 0.68 -0.028 -0.031
Inflight entertainment 0.075 0.13 0.21 -0.0084 0.047 0.0027 0.62 0.28 0.61 1 0.42 0.3 0.38 0.12 0.41 0.69 -0.027 -0.03
On-board service 0.057 0.11 0.12 0.067 0.039 -0.029 0.057 0.15 0.13 0.42 1 0.36 0.52 0.24 0.55 0.12 -0.03 -0.035
Leg room service 0.039 0.13 0.16 0.011 0.11 -0.0052 0.033 0.12 0.1 0.3 0.36 1 0.37 0.15 0.37 0.097 0.015 0.011
Baggage handling -0.048 0.065 0.12 0.071 0.039 0.00097 0.035 0.084 0.075 0.38 0.52 0.37 1 0.23 0.63 0.097 -0.0041 -0.0079
Checkin service 0.033 0.074 0.044 0.091 0.0088 -0.039 0.085 0.2 0.19 0.12 0.24 0.15 0.23 1 0.24 0.18 -0.019 -0.022
Inflight service -0.051 0.059 0.11 0.072 0.035 0.00031 0.035 0.074 0.069 0.41 0.55 0.37 0.63 0.24 1 0.09 -0.054 -0.06
Cleanliness 0.053 0.096 0.13 0.0099 0.015 -0.0059 0.66 0.33 0.68 0.69 0.12 0.097 0.097 0.18 0.09 1 -0.015 -0.017
Departure Delay in Minutes -0.009 0.0024 -0.016 0.00078 -0.0053 0.006 -0.029 -0.019 -0.028 -0.027 -0.03 0.015 -0.0041 -0.019 -0.054 -0.015 1 0.97
Arrival Delay in Minutes -0.011 -0.0019 -0.018 -0.00094 -0.007 0.0057 -0.032 -0.023 -0.031 -0.03 -0.035 0.011 -0.0079 -0.022 -0.06 -0.017 0.97 1
Age
Flight Distance
Inflight wifi service
Departure/Arrival time convenient
Ease of Online booking
Gate location
Food and drink
Online boarding
Seat comfort
Inflight entertainment
On-board service
Leg room service
Baggage handling
Checkin service
Inflight service
Cleanliness
Departure Delay in Minutes
Arrival Delay in Minutes
1.0
0.8
0.6
0.4
0.2
0.0

Data Cleaning

NA Values

```
dataset.isnull().any()
```

25 rows | Length: 25, dtype: bool

	data
Unnamed: 0	False
id	False
Gender	False
Customer Type	False
Age	False
Type of Travel	False
Class	False
Flight Distance	False
Inflight wifi service	False
Departure/Arrival time convenient	False
Ease of Online booking	False
Gate location	False
Food and drink	False
Online boarding	False
Seat comfort	False
Inflight entertainment	False
On-board service	False
Leg room service	False
Baggage handling	False
Checkin service	False
Inflight service	False
Cleanliness	False
Departure Delay in Minutes	False
Arrival Delay in Minutes	True
satisfaction	False

```
dataset.isnull().sum()
```

23 rows | Length: 23, dtype: int64

	data
Gender	0
Customer Type	0
Age	0
Type of Travel	0
Class	0
Flight Distance	0
Inflight wifi service	0
Departure/Arrival time convenient	0
Ease of Online booking	0
Gate location	0
Food and drink	0
Online boarding	0
Seat comfort	0
Inflight entertainment	0
On-board service	0
Leg room service	0
Baggage handling	0
Checkin service	0
Inflight service	0
Cleanliness	0
Departure Delay in Minutes	0
Arrival Delay in Minutes	393
satisfaction	0

Replaced **NA** with **Mean** of the Column **Arrival Data in Minutes**

Data Preprocessing

Binary Encoding

- Customer Type - ['Loyal Customer', 'disloyal Customer’]
- Gender - ['Male', ‘Female']
- Type of Travel - ['Personal Travel', 'Business travel’]
- Satisfaction - ['neutral or dissatisfied', ‘satisfied']

One Hot Encoding

- Class - ['Eco Plus', 'Business', ‘Eco']

Data Preprocessing

Data Split and Scale

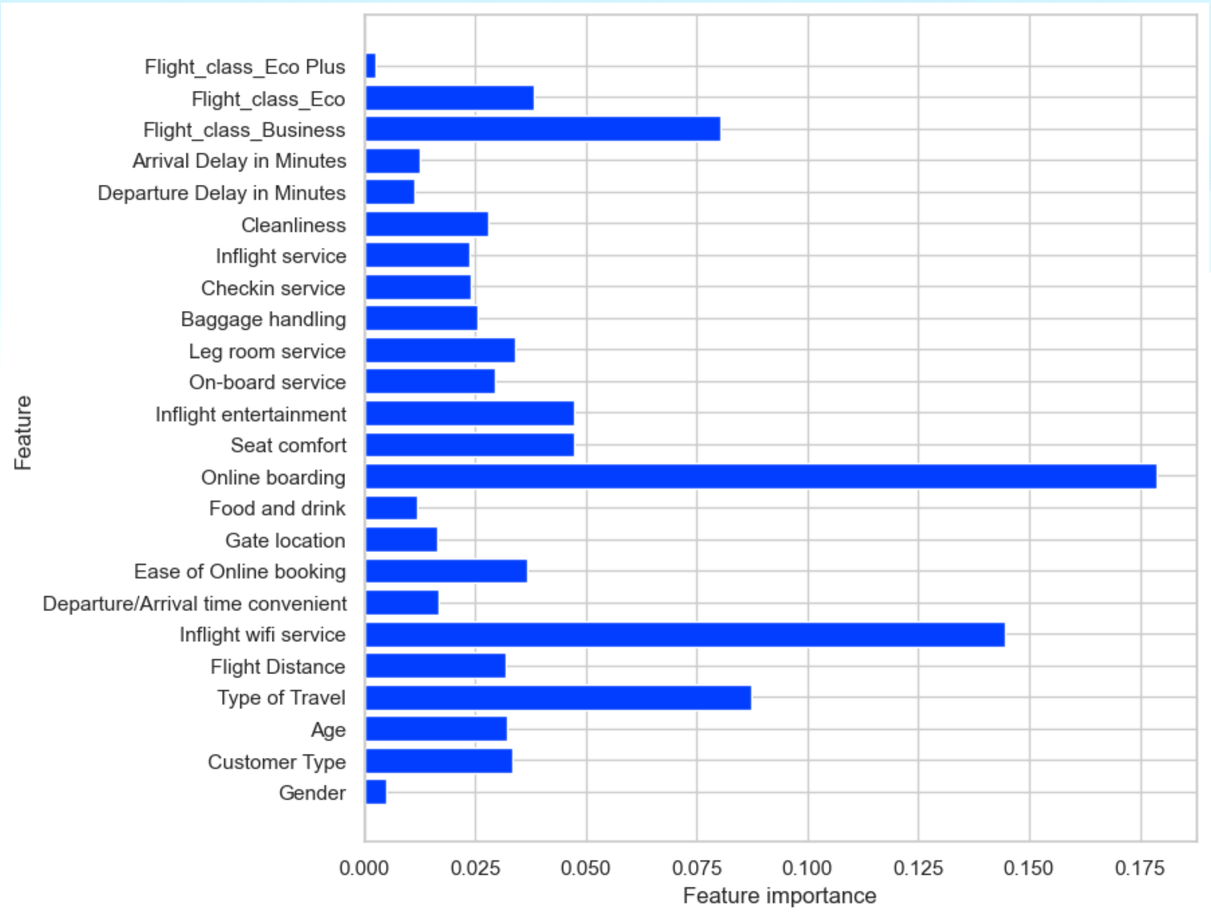
Split Data in **70 %** Test and **30 %** Train
And Scale between **-1** and **1** using
MinMaxScaler

Data Preprocessing

Feature Selection

Recursive feature elimination with cross-validation RFECV with RandomForestClassifier

1. Online boarding
2. Inflight wifi service
3. Type of Travel
4. Flight_class_Business
5. Seat comfort
6. Inflight entertainment
7. Flight_class_Eco
8. Ease of Online booking
9. Leg room service
10. Customer Type
11. Age
12. Flight Distance
13. On-board service
14. Cleanliness
15. Baggage handling
16. Checkin service
17. Inflight service
18. Departure/Arrival time convenient
19. Gate location
20. Arrival Delay in Minutes
21. Food and drink
22. Departure Delay in Minutes
23. Gender
24. Flight_class_Eco Plus



Logistic Regression with Grid SearchCV

Logistic Regression

- A Machine Learning classification algorithm
- Used to predict the probability of certain classes based on some dependent variables.

GridSearchCV

- A function that is in sklearn’s model_selection package.
- specify the different values for each hyperparameter
- training and testing using cross validation of your dataset — hence the acronym “CV” in GridSearchCV.

.

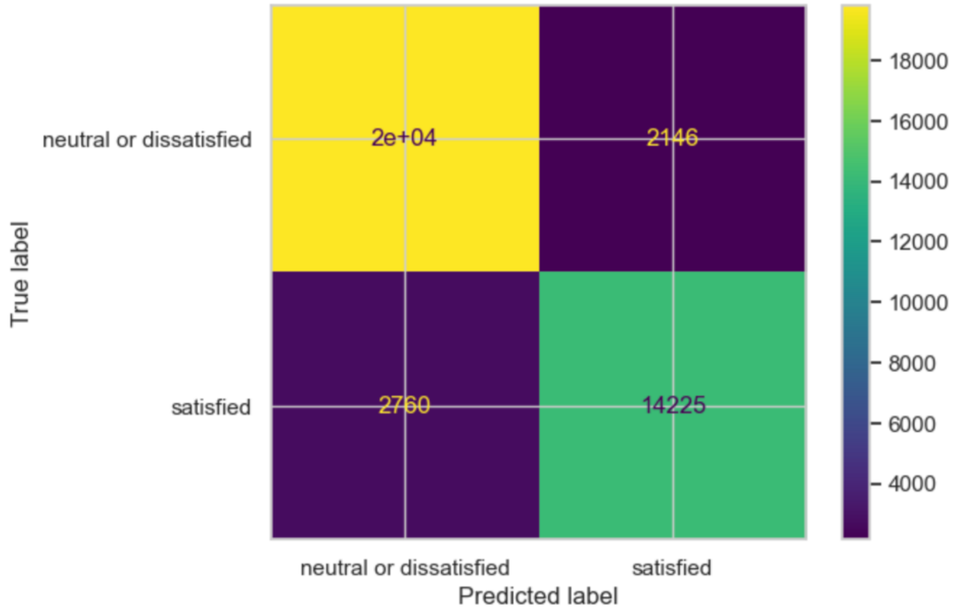
LogisticRegression with GridSearchCV

In [39]:
```
def get_logr(mode = 'load'):

    path = DIRPATH + "/Models/"
    filename = 'log_r.sav'

    if mode == 'load':
        log_r = pickle.load(open(path + filename, 'rb'))
    else:

        log_r_param_grid = {
            'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000],
            'penalty': ['l2'],
        }

        log_r = GridSearchCV(
            estimator = LogisticRegression(max_iter = 1000),
            param_grid = log_r_param_grid,
            cv = 5,
            verbose = 1,
            scoring = 'accuracy',
            n_jobs = -1
        )
        log_r.fit(X_train, y_train)

        pickle.dump(log_r, open(path + filename, 'wb'))

    return log_r
```

In [40]:
```
log_r = get_logr()
```

In [41]:
```
predictions_lr = log_r.predict(X_test)
print_metrics(y_true=y_test, y_pred=predictions_lr)
```

```
AUC-ROC:  0.8699325123187744
Classification:
                          precision    recall  f1-score   support

neutral or dissatisfied       0.88      0.90      0.89     21979
              satisfied       0.87      0.84      0.85     16985

               accuracy                           0.87     38964
              macro avg       0.87      0.87      0.87     38964
           weighted avg       0.87      0.87      0.87     38964
```



Random Forest with Randomized Search CV

- *Random Forest*
  - *Supervised Machine Learning Algorithm*
  - *used widely in Classification and Regression problems*
- **Randomized Search CV**
  - **implements a “fit” and a “score” method**
  - **a fixed number of parameter settings is sampled from the specified distributions**

.

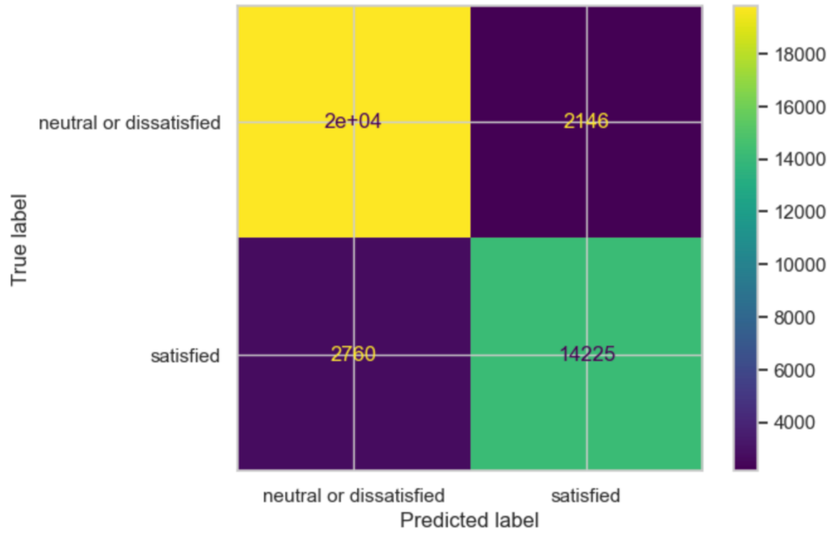
Random Forest with Randomized Search CV

In [45]:
```
def get_rf_rs(mode = "load"):

    path = DIRPATH + "/Models/"
    filename = 'rf_rs.sav'

    if mode == "load":
        rf_rs = pickle.load(open(path + filename, 'rb'))

    else:
        rf_parameters_grid = {
            'max_depth': [5, 15],
            'min_samples_leaf': [2, 8],
            'n_estimators': [50, 100],
            'max_features': [5, 10]
        }

        # define grid search
        rf_rs = RandomizedSearchCV(
            estimator = RandomForestClassifier(),
            param_distributions = rf_parameters_grid,
            cv = 10,
            verbose = 1,
            n_jobs = -1
        )
        rf_rs.fit(X_train, y_train)
        pickle.dump(knn, open(path + filename, 'wb'))

    return rf_rs
```

In [46]:
```
rf_rs = get_rf_rs()
```

In [47]:
```
predictions_rf_rs = rf_rs.predict(X_test)
print_metrics(y_true = y_test, y_pred = predictions_rf_rs)
```

```
AUC-ROC:  0.8699325123187744
Classification:
                           precision    recall  f1-score   support

neutral or dissatisfied       0.88      0.90      0.89     21979
              satisfied       0.87      0.84      0.85     16985

               accuracy                           0.87     38964
              macro avg       0.87      0.87      0.87     38964
           weighted avg       0.87      0.87      0.87     38964
```

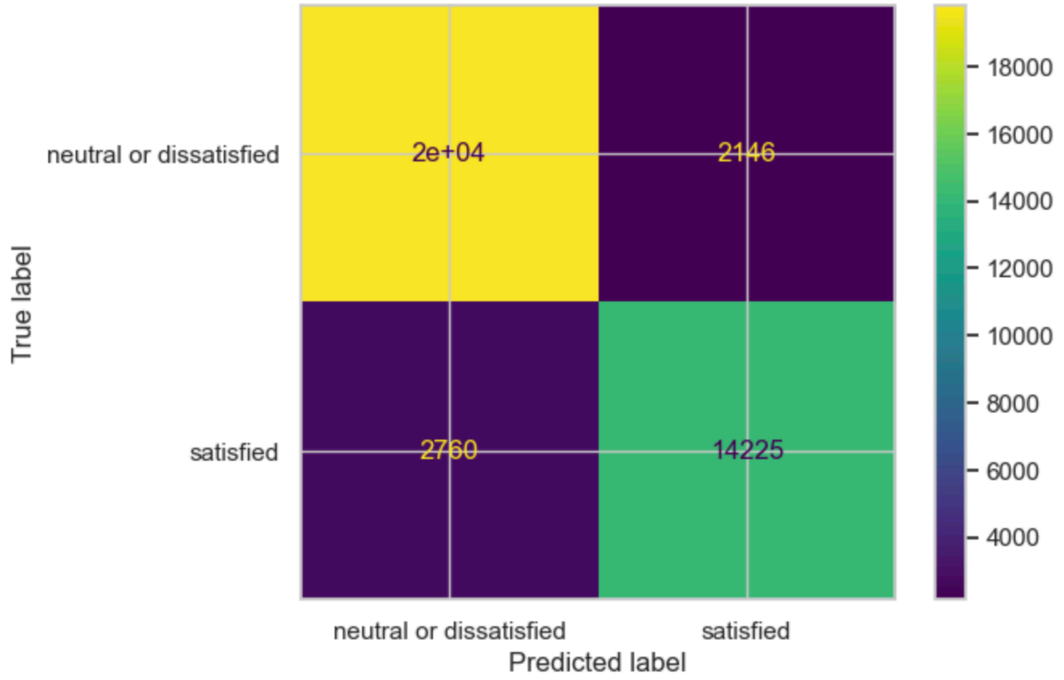


K-nearest neighbor with Grid Search CV

- Neighbors-based classification is a type of instance-based learning or non-generalizing learning.
- Key parameters search using grid search
  - n_neighbors: *int, default=5*
  - Weights: *{'uniform', 'distance'} or callable, default='uniform'*

AUC-ROC: 0.8699325123187744
Classification:

	precision	recall	f1-score	support
neutral or dissatisfied	0.88	0.90	0.89	21979
satisfied	0.87	0.84	0.85	16985
accuracy			0.87	38964
macro avg	0.87	0.87	0.87	38964
weighted avg	0.87	0.87	0.87	38964

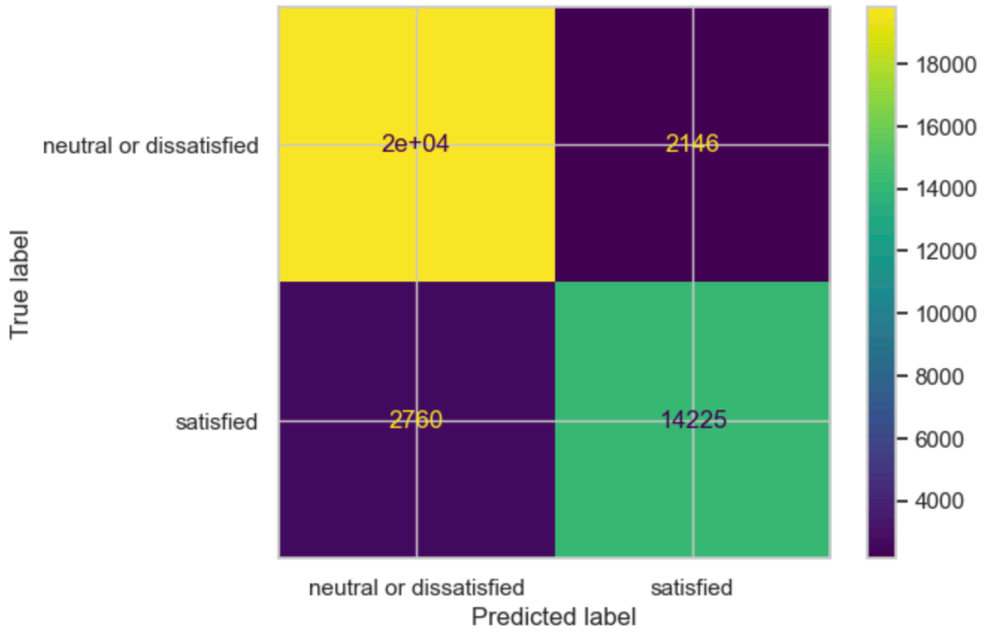


Decision Tree Classifier with Grid Search CV

- Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.
- Key parameters search using grid search
  - min_samples_split: *int or float, default=2*
  - max_depth: *int, default=None*

AUC-ROC: 0.8699325123187744
Classification:

	precision	recall	f1-score	support
neutral or dissatisfied	0.88	0.90	0.89	21979
satisfied	0.87	0.84	0.85	16985
accuracy			0.87	38964
macro avg	0.87	0.87	0.87	38964
weighted avg	0.87	0.87	0.87	38964



Support Vector Machine with Grid Search CV

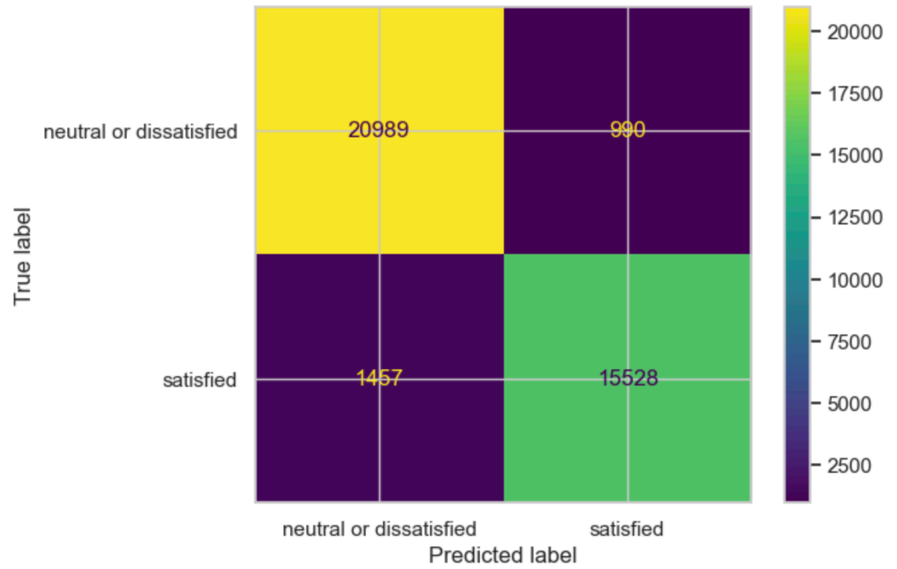
Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

```python
def get_svm(mode = 'load'):

    path = DIRPATH + "/Models/"
    filename = 'svc.sav'

    if mode == "load":
        svc = pickle.load(open(path + filename, 'rb'))
    else:
        # defining parameter range
        param_grid = {'C': [0.1, 1, 10],
                      'kernel': ['rbf'],
                      'gamma': ['scale', 'auto']}

        svc = GridSearchCV(SVC(random_state=False), param_grid, verbose = 3, cv=5, n_jobs=-1)

        svc.fit(X_new_train, y_train)
        pickle.dump(knn, open(path + filename, 'wb'))

    return svc
```

```
AUC-ROC:  0.9345877162190156
Classification:
                           precision    recall  f1-score   support

neutral or dissatisfied       0.94      0.95      0.94     21979
              satisfied       0.94      0.91      0.93     16985

               accuracy                           0.94     38964
              macro avg       0.94      0.93      0.94     38964
           weighted avg       0.94      0.94      0.94     38964
```



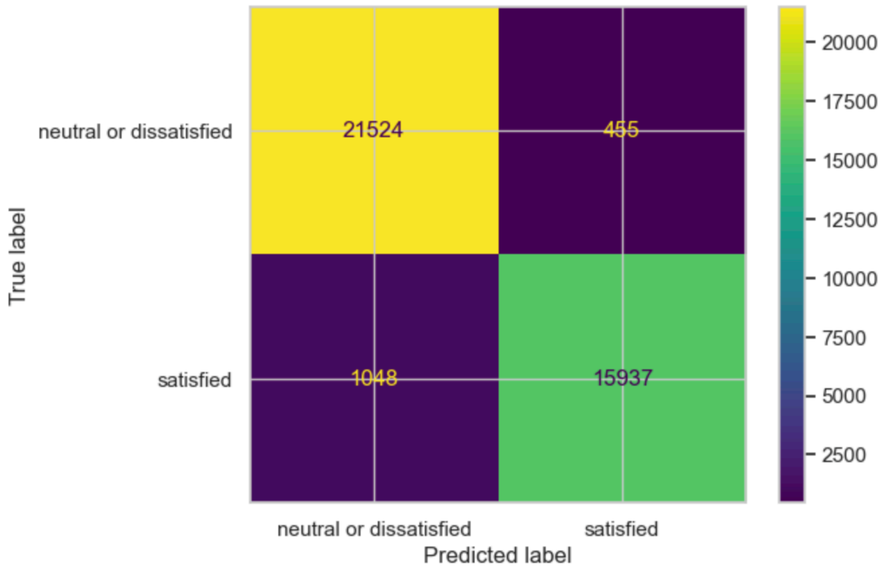
Bagging along with Random Forest

Random forests or **random decision forests** is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.

.

AUC-ROC: 0.9587984599477786
Classification:

	precision	recall	f1-score	support
neutral or dissatisfied	0.95	0.98	0.97	21979
satisfied	0.97	0.94	0.95	16985
accuracy			0.96	38964
macro avg	0.96	0.96	0.96	38964
weighted avg	0.96	0.96	0.96	38964



Multilayer Classifier

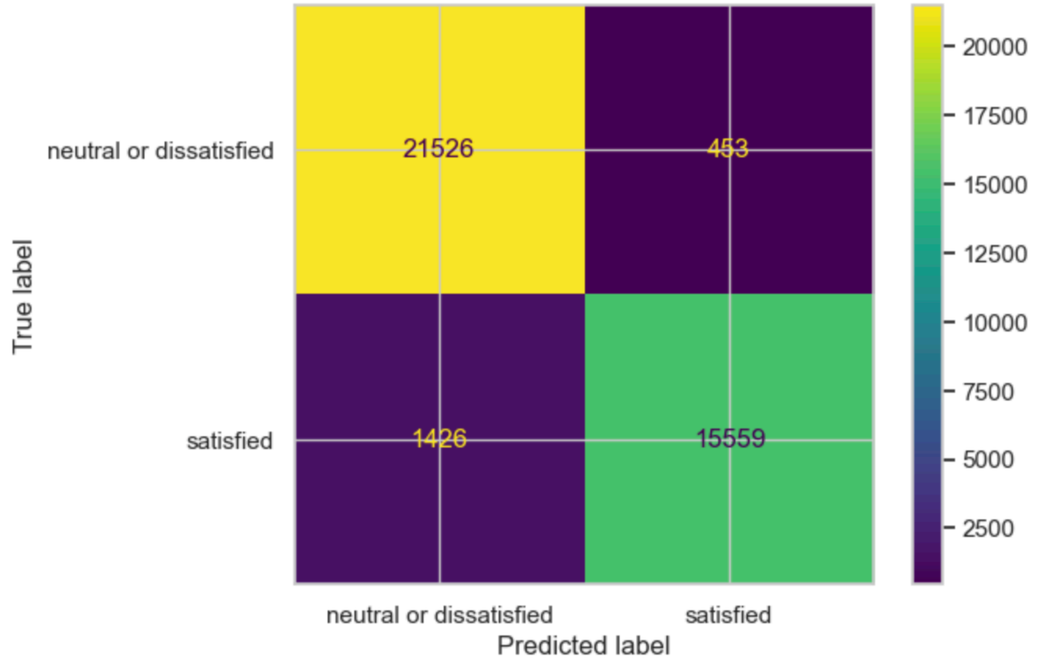
```
Metal device set to: Apple M2
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_20 (Dense)            (None, 22)                550

 dropout_5 (Dropout)         (None, 22)                0

 dense_21 (Dense)            (None, 11)                253

 dense_22 (Dense)            (None, 11)                132

 dense_23 (Dense)            (None, 1)                 12

=================================================================
```

Total params: 947

Trainable params: 947

Non-trainable params: 0

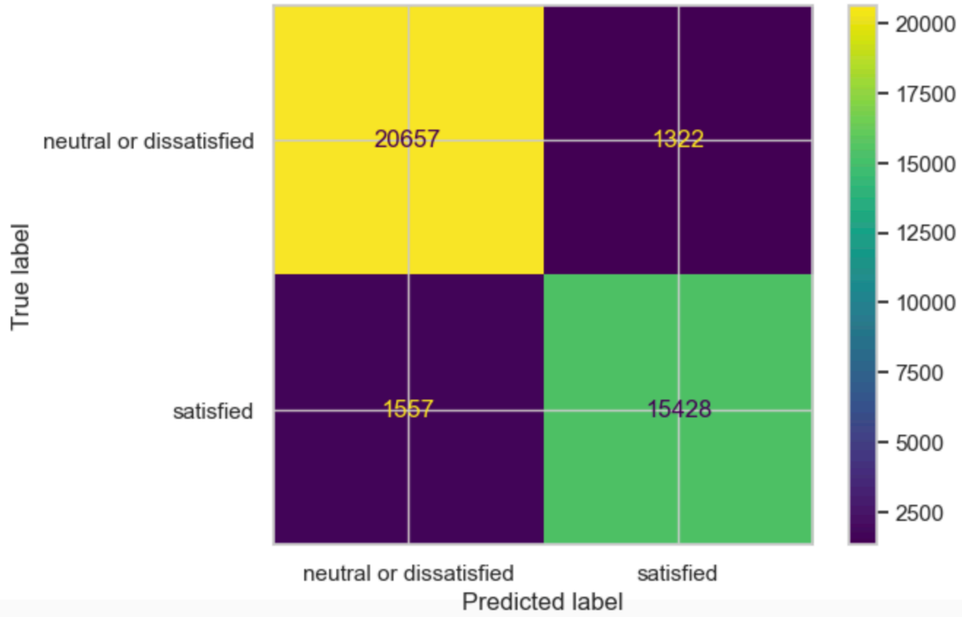
```
AUC-ROC:  0.9477164925124624
Classification:
                           precision    recall  f1-score   support

neutral or dissatisfied       0.94      0.98      0.96     21979
              satisfied       0.97      0.92      0.94     16985

               accuracy                           0.95     38964
              macro avg       0.95      0.95      0.95     38964
           weighted avg       0.95      0.95      0.95     38964
```



AdaBoost Classifier

AUC-ROC: 0.9240912783943964
Classification:

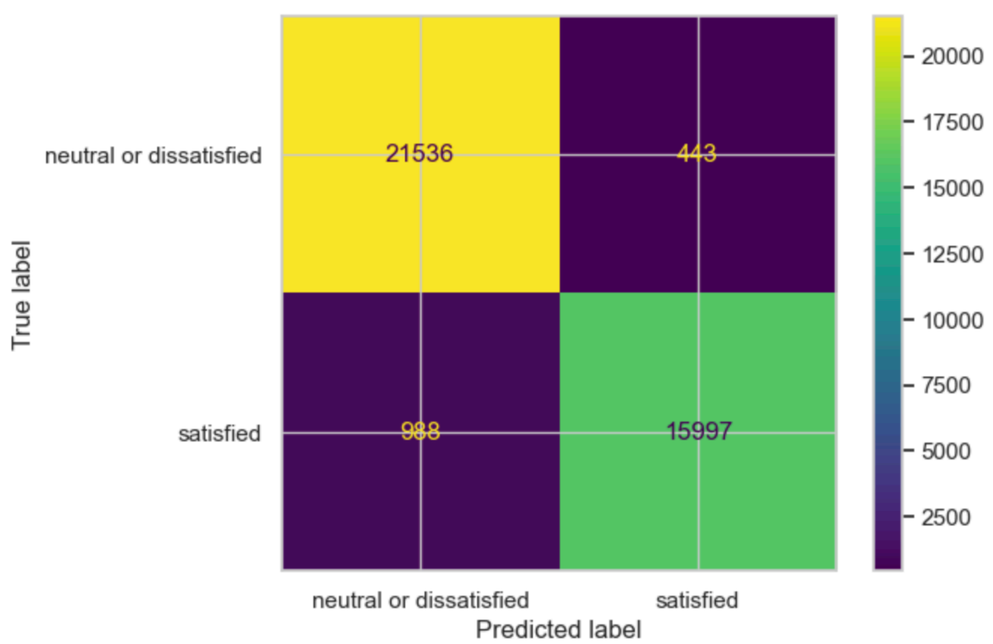
	precision	recall	f1-score	support
neutral or dissatisfied	0.93	0.94	0.93	21979
satisfied	0.92	0.91	0.91	16985
accuracy			0.93	38964
macro avg	0.93	0.92	0.92	38964
weighted avg	0.93	0.93	0.93	38964



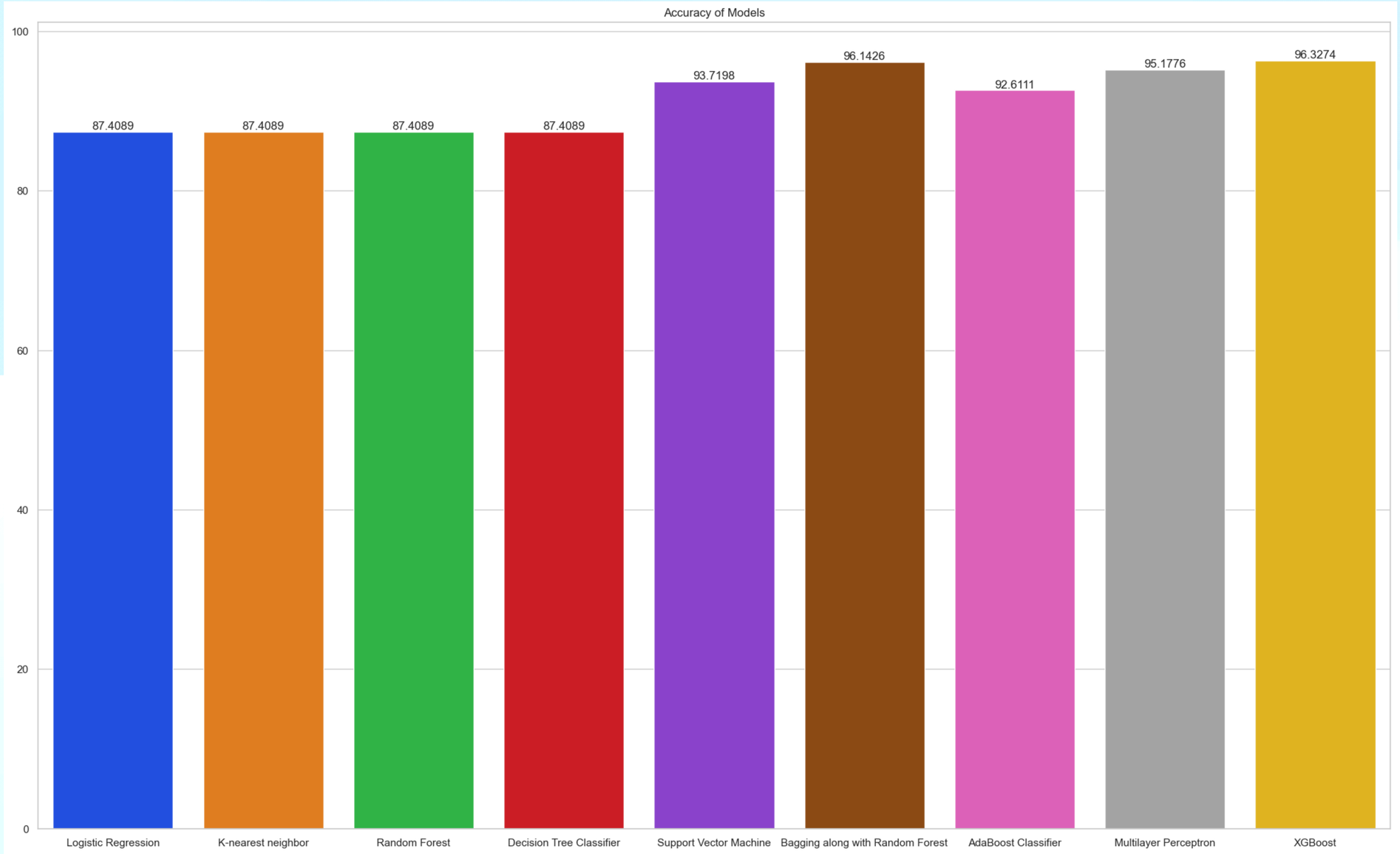
Gradient Boosting XGBoost

AUC-ROC: 0.9608377121507172
Classification:

	precision	recall	f1-score	support
neutral or dissatisfied	0.96	0.98	0.97	21979
satisfied	0.97	0.94	0.96	16985
accuracy			0.96	38964
macro avg	0.96	0.96	0.96	38964
weighted avg	0.96	0.96	0.96	38964



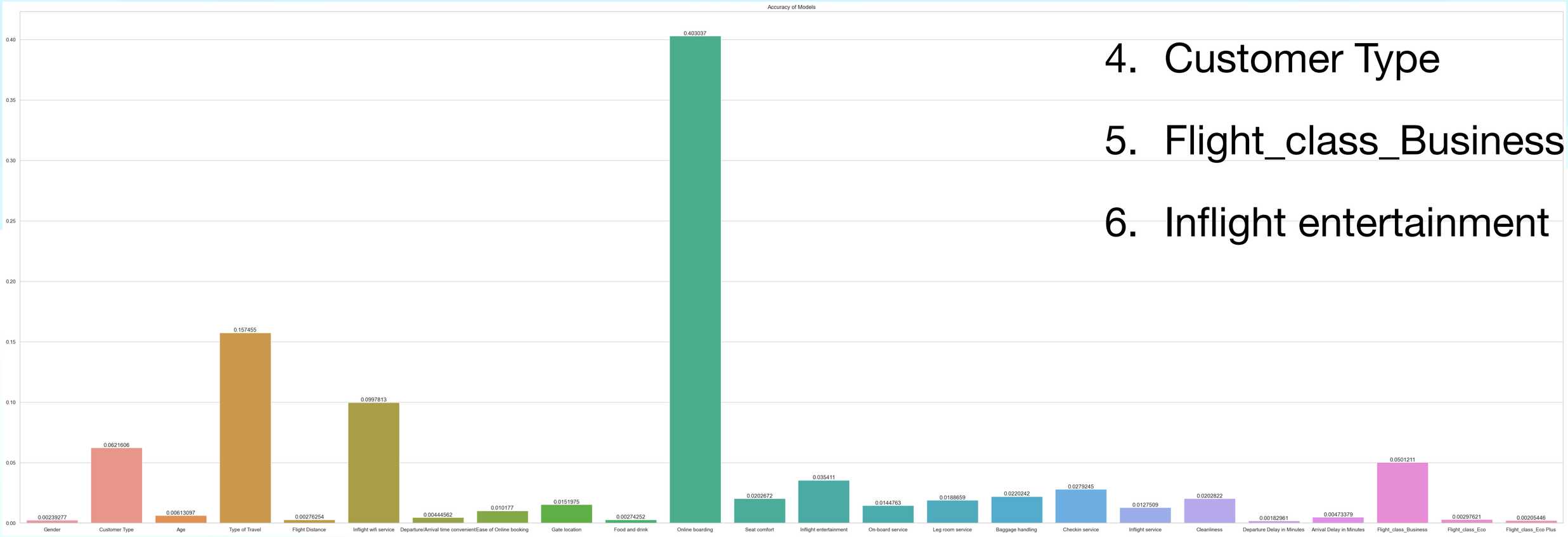
Accuracy of Models



1. XGBoost
2. Bagging with RF
3. MLP
4. SVM
5. AdaBoost
6. LR
7. KNN
8. RF
9. DTC

Important features XGB

1. Online boarding
2. Type of Travel
3. Inflight wifi service
4. Customer Type
5. Flight_class_Business
6. Inflight entertainment



Thank You