MSc Computer Science

School of Computing, Engineering and Digital technologies

***Machine Learning* CIS 4037-N**

Academic Year: 2023/24

Predicting Air Passenger Satisfaction Levels with Machine Learning

Author: Arifa Saeed

Student Id: D3658893

I. Introduction

In today's hyperconnected world, the airline industry stands as a beacon of modernity, facilitating global mobility and connectivity at an unprecedented scale. Within this dynamic landscape, airlines vie not only for market share but, crucially, for customer satisfaction. Understanding and predicting the intricate nuances of passenger sentiment is no longer just a competitive advantage; it's a strategic imperative. Within this context, our project aims to utilize machine learning to decipher and predict the satisfaction levels of air passengers traveling with Ratra Airlines.

Goals of the Study :

The primary goals of this investigation are outlined below:

1. Implementing data pre-processing and preparation methods to acquire refined data.

2. Constructing machine learning models capable of forecasting customer satisfaction using data attributes.

3. Assessing and contrasting model performance to select the optimal model.

Problem Statement:

Ratra Airlines, like many others in the industry, is acutely aware of the pivotal role passenger satisfaction plays in ensuring long-term viability and success. However, quantifying and preempting passenger satisfaction poses a formidable challenge. Traditional feedback mechanisms often fall short, capturing only snapshots of the passenger experience and lacking the predictive capabilities necessary for proactive intervention. Thus, our project aims to bridge this gap by developing a robust classification model capable of discerning between satisfied passengers and those who are neutral or dissatisfied based on a myriad of factors captured in a comprehensive survey.

II. Literature Review

In the dynamic landscape of the airline industry, where customer satisfaction is paramount, understanding the determinants of passenger sentiment has been a subject of extensive research. This literature review seeks to synthesize key findings from previous studies related to passenger satisfaction in air travel, with a particular focus on predictive modeling using machine learning techniques.

Traditional Approaches to Passenger Satisfaction Analysis:
Traditional methods of assessing passenger satisfaction have primarily relied on post-flight surveys and feedback mechanisms. Studies such as [1] have explored the efficacy of these surveys in capturing passengers' perceptions of various aspects of their travel experience, including service quality, amenities, and overall satisfaction. While valuable, these approaches often suffer from limitations such as response bias, sample representativeness, and lag in feedback processing.

Predictive Modeling in Airline Passenger Satisfaction:
With the advent of machine learning, researchers have increasingly turned to predictive modeling techniques to anticipate passenger satisfaction levels proactively. Studies such as [2] have demonstrated the feasibility of leveraging historical flight data, customer demographics, and satisfaction ratings to develop predictive models capable of forecasting passenger sentiment accurately. Through the utilization of algorithms like decision trees, random forests, and support vector machines, researchers have attained significant accomplishments in forecasting satisfaction outcomes and pinpointing crucial factors that influence passenger perceptions.

Feature Selection and Model Performance:
An essential aspect of predictive modeling in air travel satisfaction is feature selection, wherein researchers identify the most relevant predictors of satisfaction from a plethora of available variables. Research endeavors like [3] have investigated diverse feature selection techniques,

such as recursive feature elimination and principal component analysis, aiming to streamline model inputs and bolster predictive efficacy. Additionally, scholarly inquiries have scrutinized model evaluation metrics, as exemplified by studies such as [4], which advocate for a comprehensive assessment incorporating metrics like accuracy, precision, recall, and F1-score. This approach offers a nuanced comprehension of model performance.

Cross-Industry Insights and Transferability:
While much of the literature on passenger satisfaction originates from the airline industry, there exists a wealth of cross-industry insights that can inform predictive modeling efforts. Studies such as [5] have drawn parallels between customer satisfaction in airlines and other service industries, highlighting common themes such as service quality, perceived value, and customer expectations. By leveraging insights from diverse domains, researchers can enrich predictive models and enhance their transferability across industries.

Challenges and Future Directions:
Despite the progress made in predictive modeling of passenger satisfaction, there are persistent challenges such as data sparsity, variations in passenger preferences, and the necessity for real-time analysis and feedback integration. Future research avenues could involve incorporating new data sources like social media sentiment analysis, exploring advanced modeling techniques like deep learning, and devising hybrid models that blend machine learning with expert domain knowledge.

In summary, the body of literature on predictive modeling of airline passenger satisfaction illustrates a dynamic and evolving field. It merges conventional approaches with state-of-the-art machine learning techniques. By drawing insights from prior research and tackling existing obstacles, scholars can continue expanding the boundaries of knowledge and contribute to enhancing customer-centric strategies within the airline sector.

Machine Learning Techniques:

Numerous techniques exist for addressing regression problems, ranging from Linear Regression and Ridge Regression to Artificial Neural Networks, Decision Trees, and Random Forests. This project aims to evaluate multiple modeling techniques and subsequently select the ones that demonstrate superior performance.

Handling Imbalanced Data:

In real-world datasets, class imbalances are common, where one class significantly outweighs the other. For instance, there may be far more satisfied passengers than dissatisfied ones, or vice versa. To mitigate this issue, various techniques can be employed:

1. **Resampling Methods:**
   - Oversampling the minority class (e.g., dissatisfied passengers) or undersampling the majority class (e.g., satisfied passengers) can help balance the dataset.
2. **Appropriate Metrics:**
   - Metrics like precision, recall, F1-score, and ROC-AUC are more informative than accuracy when dealing with imbalanced data.
3. **Algorithmic Approaches:**
   - Certain algorithms, such as Random Forest or SVM with class weights, inherently handle imbalanced data more effectively.

Model Evaluation and Performance Metrics:

Model Evaluation:

Assessing the performance of machine learning models is crucial for determining their effectiveness in addressing the problem at hand.

Performance Metrics:

Several metrics are commonly used to gauge model performance:

1. Accuracy:
   - Measures the proportion of correctly classified instances.

2. Precision:
   - Indicates the proportion of true positive predictions among all positive predictions.

3. Recall:

- Measures the proportion of true positive predictions among all actual positive instances.

4. F1-score:

- Represents the harmonic mean of precision and recall, offering a balanced assessment.

5. ROC-AUC:

- Denotes the area under the Receiver Operating Characteristic curve, reflecting the model's ability to distinguish between classes.

Evaluation Techniques:

Various techniques, including cross-validation, train-test splits, and confusion matrices, are employed to evaluate model performance effectively.

III. Methodology

A. Data Stories

Prior to immersing ourselves in the intricacies of model development, grasping the contextual nuances of the data and its relevance to the problem at hand is imperative.

```
data
```

	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Infli ser
0	1	1	1	48	0	0	821	3	3	3	...	5	3	2	5	4	
1	2	0	0	35	0	0	821	2	2	2	...	5	5	5	5	3	
2	3	1	0	41	0	0	853	4	4	4	...	3	3	3	3	4	
3	4	1	0	50	0	0	1905	2	2	2	...	5	5	5	5	3	
4	5	0	0	49	0	0	3470	3	3	3	...	3	3	4	3	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
49995	80712	0	0	33	0	0	920	3	3	5	...	1	1	2	1	3	
49996	80713	0	1	24	0	1	920	3	3	3	...	3	2	3	4	2	
49997	80714	1	0	37	0	0	2483	2	2	2	...	2	2	1	2	3	
49998	80715	0	0	22	0	0	3001	1	4	4	...	1	4	2	3	3	
49999	80716	0	1	36	0	0	920	2	2	2	...	4	4	4	5	3	

50000 rows × 24 columns

This step sets the foundation for subsequent data processing and modeling tasks by providing valuable context and guiding feature selection. For instance, explore whether certain demographics or flight classes correlate with higher satisfaction levels. Explanation for the data columns containing in our dataset:

B. Data Cleaning and Pre-processing

Data cleaning and pre-processing are crucial steps to ensure the quality and consistency of the dataset for analysis.

```
In [9]: data.describe(include="all").T
```

Out[9]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
id	50000.0	NaN	NaN	NaN	42942.46444	28035.881899	1.0	12500.75	50719.5	68216.25	80716.0
Gender	50000	2	Female	25224	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Customer Type	50000	2	Loyal Customer	40883	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Age	50000.0	NaN	NaN	NaN	39.58824	15.139425	7.0	27.0	40.0	51.0	85.0
Type of Travel	50000	2	Business travel	34598	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Class	50000	3	Business	24513	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Flight Distance	50000.0	NaN	NaN	NaN	1180.47108	1017.147731	56.0	399.0	802.0	1744.0	4963.0
Inflight wifi service	50000.0	NaN	NaN	NaN	2.72356	1.328961	0.0	2.0	3.0	4.0	5.0
Departure/Arrival time convenient	50000.0	NaN	NaN	NaN	3.06092	1.52819	0.0	2.0	3.0	4.0	5.0
Ease of Online booking	50000.0	NaN	NaN	NaN	2.7577	1.409706	0.0	2.0	3.0	4.0	5.0
Gate location	50000.0	NaN	NaN	NaN	2.97972	1.282974	1.0	2.0	3.0	4.0	5.0
Food and drink	50000.0	NaN	NaN	NaN	3.20054	1.32933	0.0	2.0	3.0	4.0	5.0
Online boarding	50000.0	NaN	NaN	NaN	3.25346	1.351819	0.0	2.0	3.0	4.0	5.0
Seat comfort	50000.0	NaN	NaN	NaN	3.43982	1.320223	0.0	2.0	4.0	5.0	5.0
Inflight entertainment	50000.0	NaN	NaN	NaN	3.35838	1.336076	0.0	2.0	4.0	4.0	5.0
On-board service	50000.0	NaN	NaN	NaN	3.39716	1.282832	0.0	2.0	4.0	4.0	5.0
Leg room service	50000.0	NaN	NaN	NaN	3.36218	1.31648	0.0	2.0	4.0	4.0	5.0
Baggage handling	50000.0	NaN	NaN	NaN	3.64194	1.178915	1.0	3.0	4.0	5.0	5.0
Checkin service	50000.0	NaN	NaN	NaN	3.3213	1.259054	0.0	3.0	3.0	4.0	5.0
Inflight service	50000.0	NaN	NaN	NaN	3.65948	1.171475	0.0	3.0	4.0	5.0	5.0
Cleanliness	50000.0	NaN	NaN	NaN	3.285	1.31437	0.0	2.0	3.0	4.0	5.0
Departure Delay in Minutes	50000.0	NaN	NaN	NaN	15.40152	41.83543	0.0	0.0	0.0	11.0	1592.0
Arrival Delay in Minutes	50000.0	NaN	NaN	NaN	16.173849	42.336815	0.0	0.0	0.0	13.0	1584.0
satisfaction	50000	2	neutral or dissatisfied	28322	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Transform categorical variables into numerical format using encoding techniques. Next, standardize or normalize numerical features to maintain consistent scales.

Categorical and continous column distribution

here we will distribute features into categorical and continuous columns based on types variations in features. If feature has more than 6 types it will treated as continuous else categorical.

```
0]: categorical_col,continuous_col=[],[]    #creating two categories for column distribution
    def categorical_continuous_columns(data):   #defining a function


     for col in data.columns:   #for loop to check for all column in data.columns
        if data[col].value_counts().count()<7:  #cheking if varieties are less than 7
            categorical_col.append(col)         #if condition is true column will add into categorical_col
        else:
            continuous_col.append(col)          #if above condition is false col will add in continuous_col
     return categorical_col,continuous_col
```

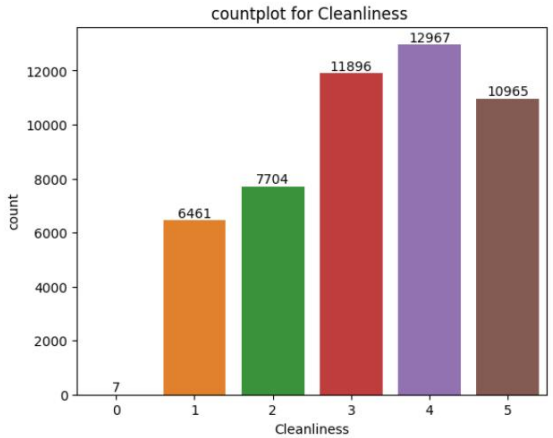
```
1]: categorical_col,continuous_col=categorical_continuous_columns(data)
```

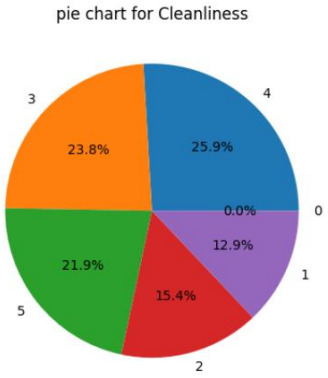
```
2]: #Categorical column
    categorical_col
```

```
2]: ['Gender',
     'Customer Type',
     'Type of Travel',
     'Class',
     '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',
     'satisfaction']
```

C. Exploratory Data Analysis

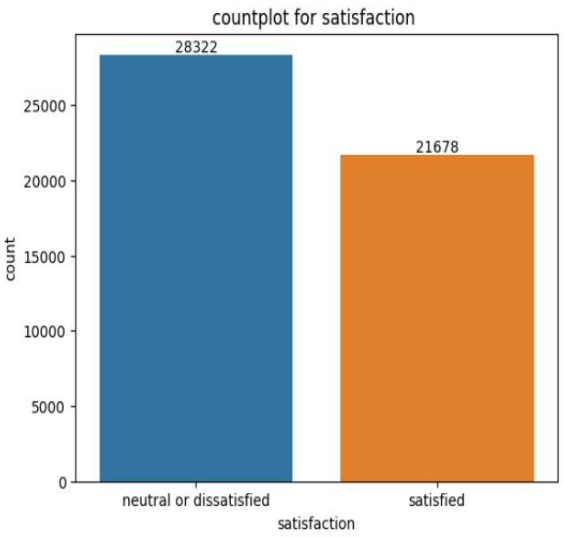
Exploratory Data Analysis (EDA) involves visualizing and summarizing key characteristics of the data-set to gain a deeper understanding of its underlying structure and relationships.

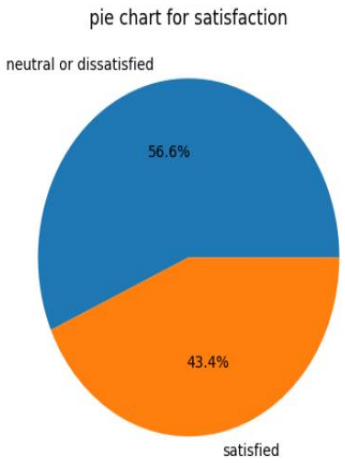




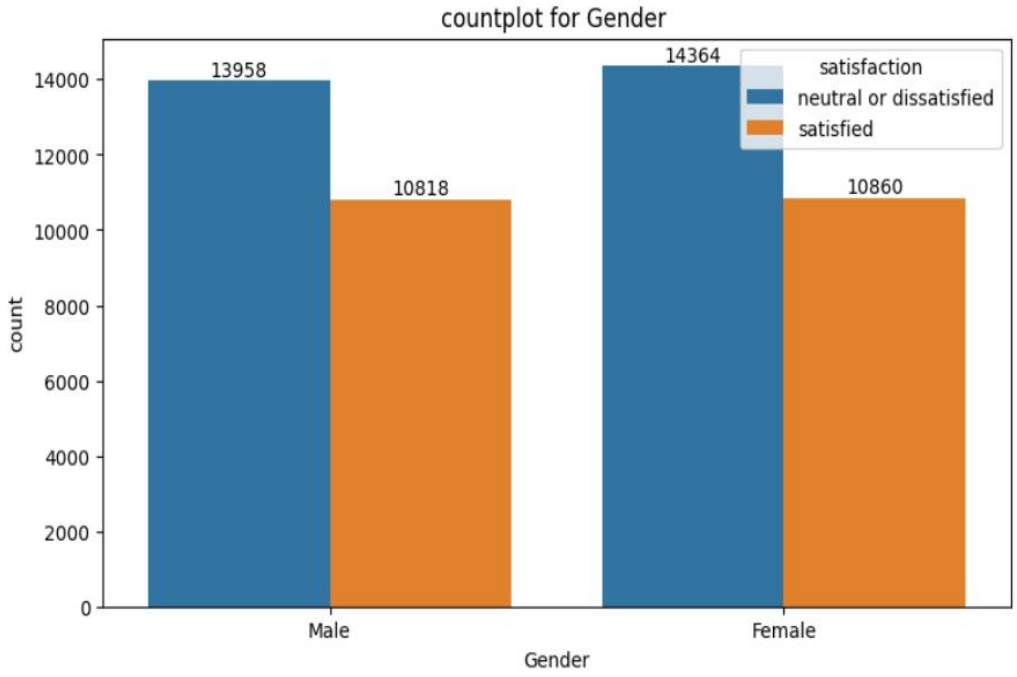

```
**************************************************
value_counts for category satisfaction
```

Conduct thorough exploratory data analysis to visualize feature distributions, correlations, and relationships.

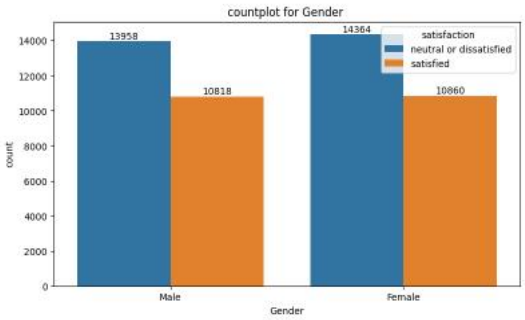




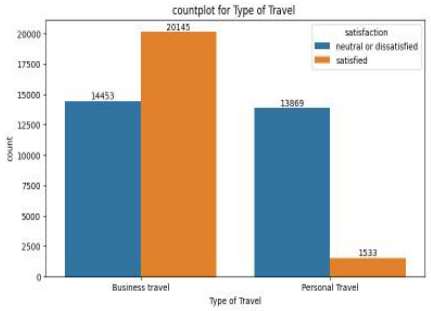
Count Plot for satisfaction dissatisfaction conduct thorough exploratory data analysis to visualize feature distributions, correlations, and relationships.



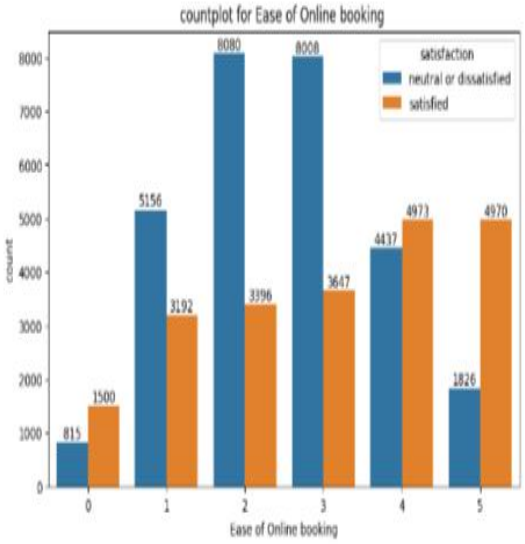
Count Plot for Gender discrimination using satisfaction ,dissatisfaction conduct thorough exploratory data analysis to visualize feature distributions, correlations, and relationships.

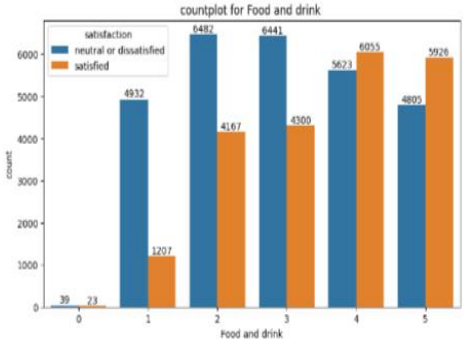



```
**************************************************
value_counts for category Customer Type

Customer Type
Loyal Customer       48883
disloyal Customer     9117
Name: count, dtype: int64
```

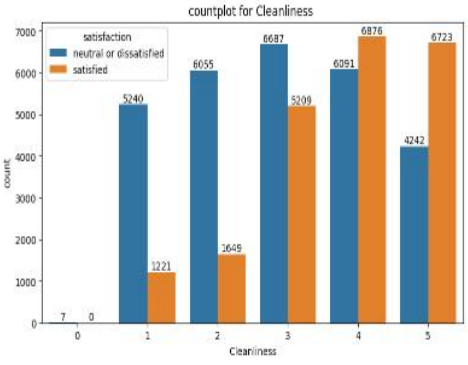



```
**************************************************
value_counts for category Class

Class
Business    24513
Eco         21996
Eco Plus     3491
Name: count, dtype: int64
```

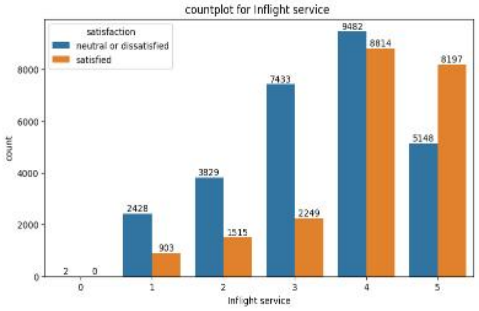





```
**************************************************
value_counts for category Online boarding

Online boarding
4    14873
3    10455
5     9997
2     8400
1     5031
0     1244
Name: count, dtype: int64
```

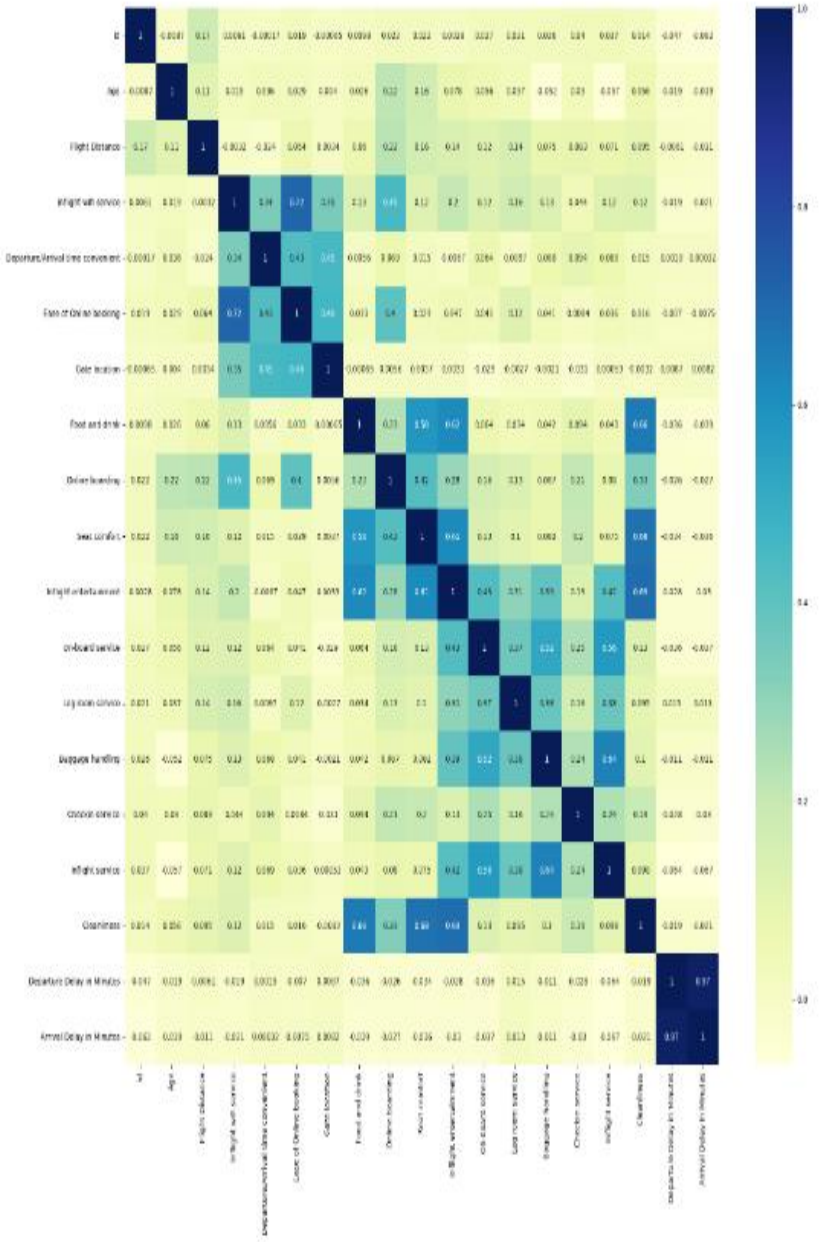



```
**************************************************
value_counts for category satisfaction

satisfaction
neutral or dissatisfied    28322
satisfied                  21678
Name: count, dtype: int64
```



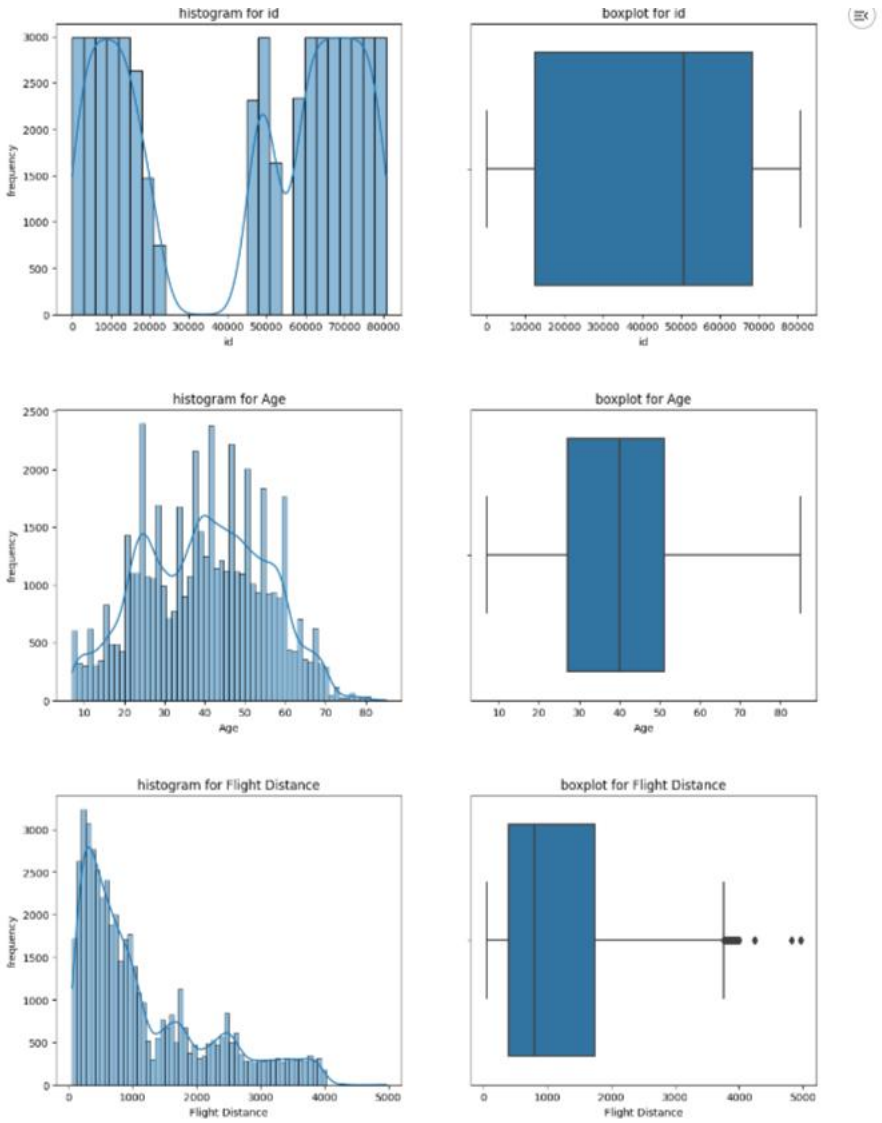

```
**************************************************
value_counts for category Cleanliness

Cleanliness
4    12967
3    11896
5    10965
2     7704
1     6461
0        7
Name: count, dtype: int64
```

Translate correlations and relationships between variables into numerical format using encoding techniques. Then, standardize or normalize numerical features to ensure uniform scales.



Uni variate analysis for Continuous columns

- *using histogram and boxplot*



D. Feature Engineering

Feature engineering is pivotal in predictive model development, as it involves crafting new variables or modifying existing ones to encapsulate pertinent information for the task at hand. Generate novel features or adapt current ones to bolster model efficacy. Employ techniques like feature scaling, polynomial features, or domain-specific transformations to achieve this goal.

Feature Scaling

To ensure optimal performance of all algorithms with our dataset, feature scaling is necessary. We'll utilize the `StandardScaler()` function from the widely-used Scikit-Learn Python package for this purpose. This function standardizes features by subtracting the mean and scaling to unit variance,

operating independently on each feature. For a value \(x \) of a feature \(F \), the `StandardScaler()` function performs the following operation:

Scaling of data ¶

```
In [24]: #importing standard scaler to scale features
         from sklearn.preprocessing import StandardScaler
         scale=StandardScaler()

In [25]: x_train=scale.fit_transform(x_train)    #scaling training data by standard scaler
         x_test=scale.transform(x_test)  #transforming scale into test data
         x_val=scale.transform(x_val)  #transforming scale into test data

In [26]: #loading training data to check if it scaled or not
         x_train.shape,x_val.shape,x_test.shape,y_train.shape,y_val.shape,y_test.shape

Out[26]: ((32000, 23), (8000, 23), (10000, 23), (32000,), (8000,), (10000,))
```

We've designated the test set size to be 25% of the entire dataset, leaving 75% for the training dataset. Consequently, we now have four subsets: X_{train}, X_{test}, y_{train}, and y_{test}. These subsets will be utilized accordingly: X_{train} and y_{train} for training the model, and X_{test} and y_{test} for testing and evaluating the model. Henceforth, X_{train} and y_{train} will be referred to as the training dataset, while X_{test} and y_{test} will be referred to as the test dataset. Figure 10 illustrates an example of the functionality of **`train_test_split()`**.

E. Fitting Machine Learning Model

Once the dataset is prepared and feature engineering is complete, the next step is to select appropriate machine learning algorithms for modeling price prediction.

preparing training and testing data

```
22]: #importing train test split from sklearn.model_selection
     from sklearn.model_selection import train_test_split
     x=data.drop(["satisfaction"],axis=1) #input data to model except result "satisfaction"
     y=data["satisfaction"]   #satisfaction is prediction
     x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20)   #defining training and test data, test size=0.2

23]: x_train,x_val,y_train,y_val=train_test_split(x_train,y_train,test_size=0.20)   #defining training and val data, test
```

F. Model Implementation

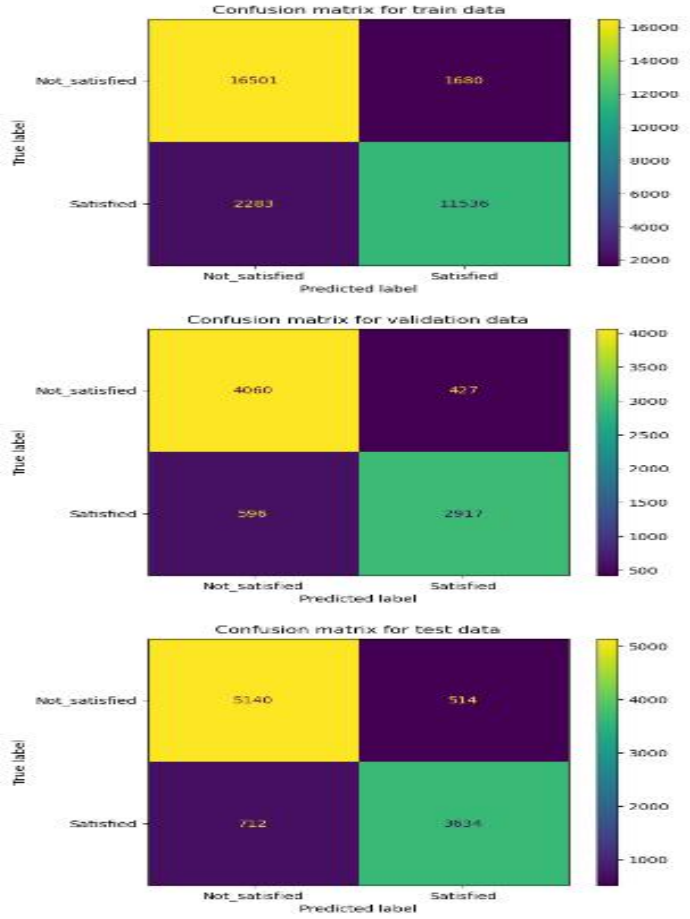
Model implementation refers to the process of deploying the trained machine learning model into a production environment, enabling it to make predictions on new data.

Model Building and Evaluation

In this section, we will construct our prediction model by selecting algorithms for each of the techniques outlined in the preceding section. Following model construction, we will assess its performance and outcomes. Let's summarize the machine learning building techniques:

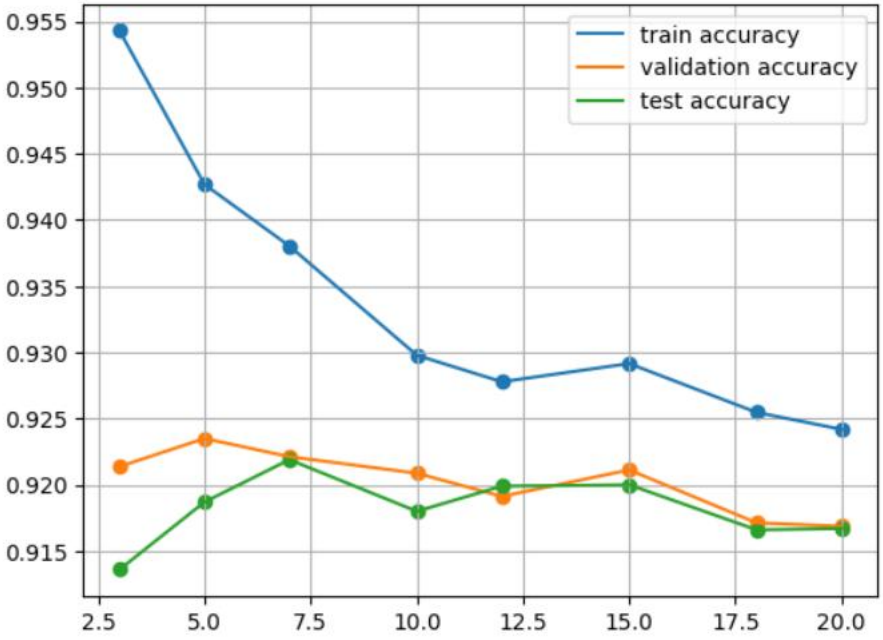
Logistic Regression:

In the project, logistic regression is utilized as a binary classification algorithm to predict passenger satisfaction. It models the probability of a binary outcome based on features such as gender, age, type of travel, flight class, and various satisfaction ratings.



K-Nearest Neighbors (KNN) Classifier:

KNN, or K-Nearest Neighbors, is a non-parametric, instance-based algorithm primarily employed for classification tasks.



In the project, KNN is utilized to classify passengers' satisfaction levels based on the similarity of their features to those of other passengers in the dataset.

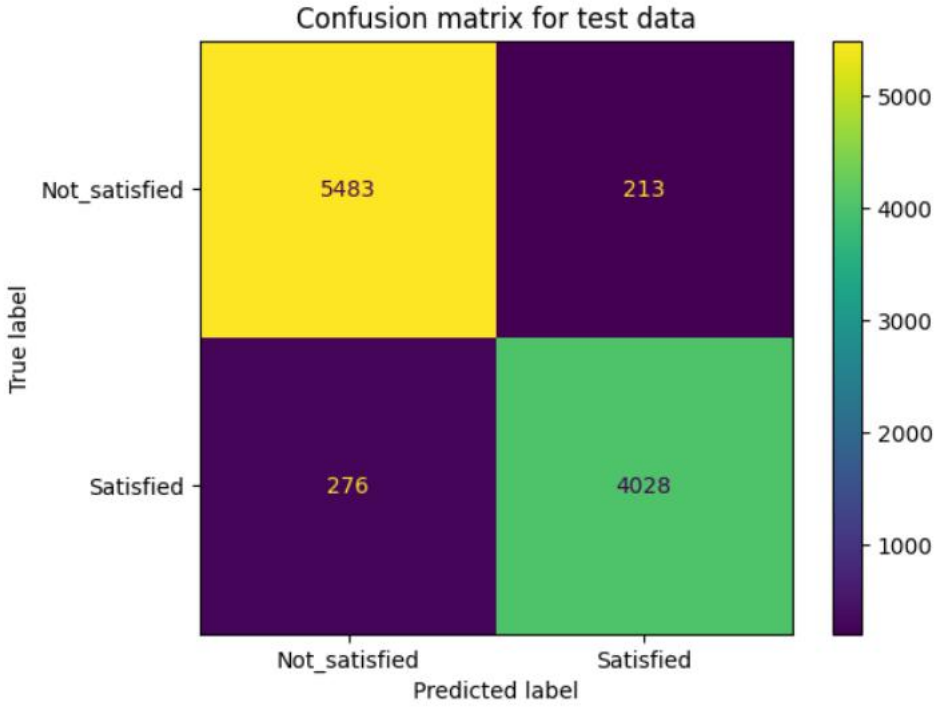
	k	accuracy_score_train	accuracy_score_val	accuracy_score_test
0	3	0.954281	0.921375	0.9136
1	5	0.942688	0.923500	0.9187
2	7	0.938031	0.922125	0.9219
3	10	0.929781	0.920875	0.9180
4	12	0.927781	0.919125	0.9199
5	15	0.929156	0.921125	0.9200
6	18	0.925469	0.917125	0.9166
7	20	0.924188	0.916875	0.9167

Support Vector Machine (SVM) Classifier:

SVM, or Support Vector Machine, is a versatile supervised learning algorithm capable of handling both classification and regression tasks.

	precision	recall	f1-score	support
0	0.95	0.96	0.96	5696
1	0.95	0.94	0.94	4304
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

In this project, SVM is utilized to classify passenger satisfaction by identifying the hyperplane that optimally separates the classes within the feature space.

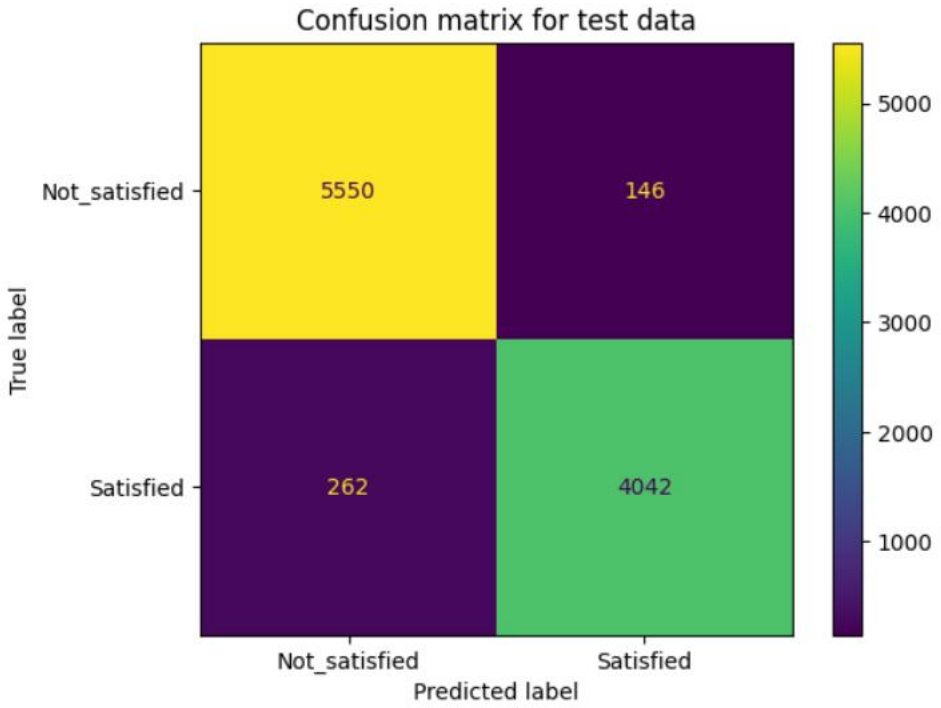


Random Forest Classifier:

Random forest is an ensemble learning technique that builds multiple decision trees during training and outputs the mode of the classes for classification.

	precision	recall	f1-score	support
0	0.95	0.97	0.96	5696
1	0.97	0.94	0.95	4304
accuracy			0.96	10000
macro avg	0.96	0.96	0.96	10000
weighted avg	0.96	0.96	0.96	10000

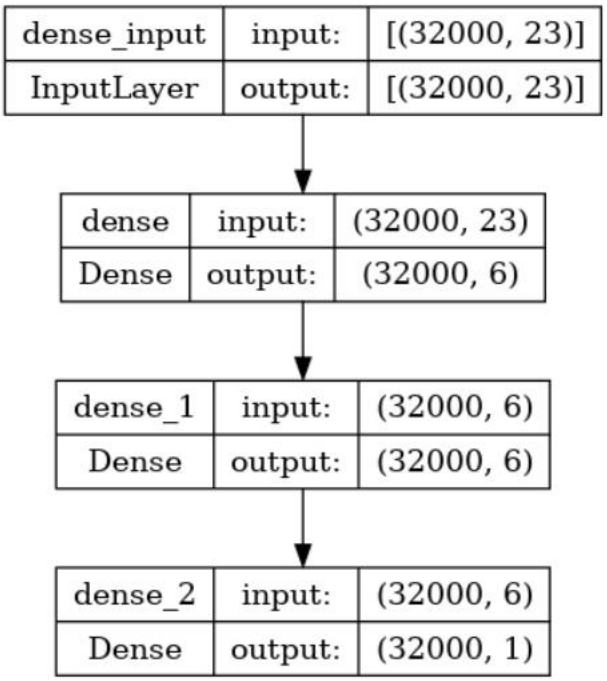
In the project, a random forest classifier is employed to predict passenger satisfaction by combining the predictions of multiple decision trees.



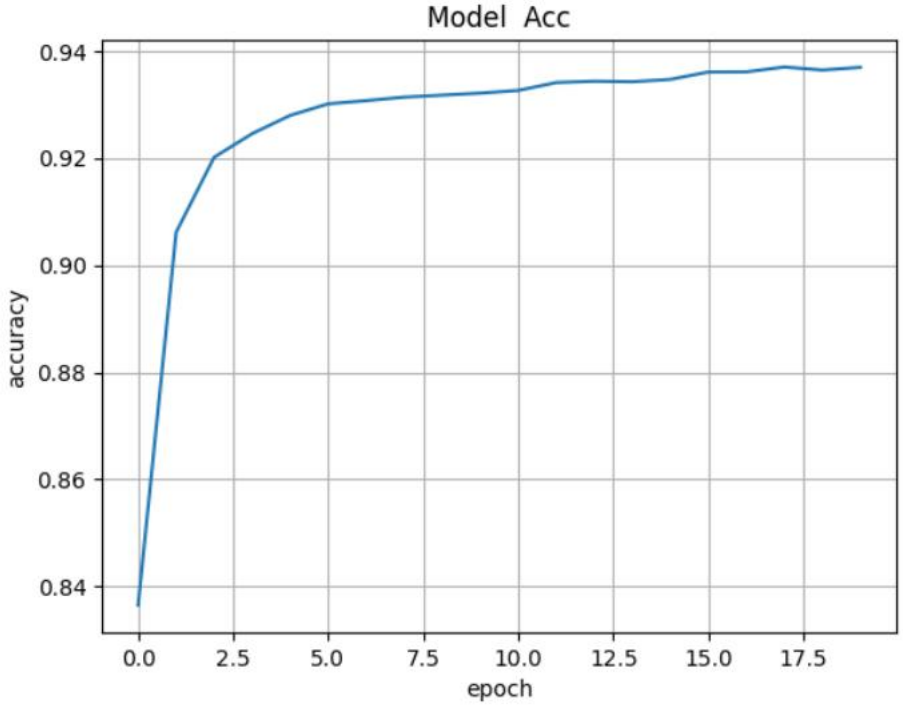
Deep Learning Model:

Deep learning entails training neural networks with multiple hidden layers to recognize intricate patterns and relationships within the data.

```
from tensorflow.keras.utils import plot_model
plot_model(dl_classifier, to_file = "model_1.png", show_shapes=True)
```



In the project, a deep learning model is constructed using the Keras library to predict passenger satisfaction based on the provided features.



Conclusion

In summary, our study aimed to forecast passenger satisfaction with Ratra Airlines utilizing various machine learning methodologies. Through comprehensive preprocessing, exploratory data analysis, feature engineering, and model construction, we obtained crucial insights into the factors influencing satisfaction and developed precise prediction models.

The key findings revealed that the Random Forest model exhibited the highest accuracy, closely trailed by SVM. Factors such as inflight entertainment, seat comfort, onboard service, and cleanliness emerged as notable determinants of satisfaction. To tackle class imbalance, we employed oversampling and undersampling techniques, while model performance was assessed using metrics like accuracy, precision, recall, and F1-score.

The implications of our study include potential operational enhancements grounded on identified factors, personalized services, and continuous monitoring for adaptation. Nonetheless, limitations such as data quality and model interpretability were noted, emphasizing the necessity for broader datasets and transparent models. Furthermore, external factors like economic conditions may also impact satisfaction levels, underscoring the importance of integrating external data sources.

In essence, our study underscores the potential of machine learning in comprehending and predicting passenger satisfaction, providing airlines with invaluable insights for operational streamlining and customer satisfaction augmentation.

References

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Springer.
- Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing.

- Brownlee, J. (2016). Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End. Machine Learning Mastery.
- McKinney, W., & others. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.
- Chollet, F. (2017). Deep Learning with Python. Manning Publications.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798-1828.
- McKinney, W., & others. (2018). Python for Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.
- VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.