一張含有 圖形, 圓形, 平面設計, 藝術 的圖片

自動產生的描述

Group Name: NHK

Student Name: Wan Hoi Nam 23021285D

Fung Pui Kiu 23034284D

Lai Chun Ho 23020364D

COMP 4433 - Data Mining and Data Warehousing

Competition – Spaceship Titanic

Content

[Exploratory Data Analysis (EDA) 3](#_Toc150474618)

[Handle Missing Value 6](#_Toc150474631)

[Feature Engineering (FE) 7](#_Toc150474632)

[Preprocessing 11](#_Toc150474637)

[Model Analysis 11](#_Toc150474640)

[Result and Conclusion 13](#_Toc150474644)

[Link of Kaggle 14](#_Toc150474645)

# Objective of project

Our project is to analyze the data of the spaceship titanic which aims to predict the transportation rate of passengers to an alternative dimension. The deliverables of this project will be trained data and provide predictions through model analysis.

# Exploratory Data Analysis (EDA)

EDA is used to investigate dataset and understand the characteristics and relationship of data, often employ the analyzation visually. It helps to determine the pattern of dataset, spot abnormal data. It offers a direction for further analysis such as Feature Engineering.

## Age Distribution

|  |
| --- |
| sns.histplot(data=train, x='Age', hue='Transported', binwidth=1, kde=True)  plt.title('Age distribution') |

Through hist graph, it shows the highest frequency of age group is 20 – 29 years old which is **less likely** to be transported. Besides, the group of 0 – 9 years old is more likely to be transported, especially group of 0 years old. Moreover, passenger who over 29 years old is equally likely to be transported. 一張含有 螢幕擷取畫面, 繪圖, 文字, 圖表 的圖片

自動產生的描述

#### Findings

It can be found that the distribution of age group can be separated into a various of groups and have further analysis with feature engineering. For instance, separate the age group into 0 - 9 years old, 10 - 19 years old so on.

## Expenditure

|  |
| --- |
| # Expenditure features  exp\_feats=['RoomService', 'FoodCourt', 'ShoppingMall', 'Spa', 'VRDeck']  # Plot expenditure features  fig=plt.figure(figsize=(10,20))  for i, var\_name in enumerate(exp\_feats):  # Left plot  ax=fig.add\_subplot(5,2,2\*i+1)  sns.histplot(data=train, x=var\_name, axes=ax, bins=30, kde=False, hue='Transported')  ax.set\_title(var\_name)  # Right plot(truncated)  ax=fig.add\_subplot(5,2,2\*i+2)  sns.histplot(data=train, x=var\_name, axes=ax, bins=30, kde=True, hue='Transported')  plt.ylim([0,100])  ax.set\_title(var\_name)  fig.tight\_layout() # Improves appearance a bit  plt.show() |

Besides, passenger can purchase some facilities provided in Spaceship. Through analyzation, the customer behavior can be affected with the expenditure of spaceship service.

### 一張含有 圖表, 繪圖, 文字, 螢幕擷取畫面 的圖片 自動產生的描述Room Services

Passengers who **spend less money** are more likely to be transported while passengers who **spend more money** may not be able to be transported.

### Food court

一張含有 繪圖, 行, 圖表, 螢幕擷取畫面 的圖片

自動產生的描述

Passengers who **spend less money** are less likely to be transported while passenger who **spends more money** may likely to be transported.

一張含有 繪圖, 圖表, 行, 螢幕擷取畫面 的圖片

自動產生的描述

### Shopping Mall

Passengers who **spend less money** are less likely to be transported while passenger who **spends more money** may likely to be transported.

### 一張含有 行, 繪圖, 圖表, 螢幕擷取畫面 的圖片 自動產生的描述Spa

Passengers who **spend less money** are more likely to be transported while passengers who **spend more money** may not be able to be transported.

一張含有 繪圖, 圖表, 行, 螢幕擷取畫面 的圖片

自動產生的描述

### VR Deck

Passengers who **spend less money** are more likely to be transported while passengers who **spend more money** may not be able to be transported.

#### Finding

The amount of spending on passengers has rapidly decreased with the high number of expenditures. The distribution of Room service, Spa, VR deck is different from the distribution of Food court, Shopping mall. Most transported passengers spend less money on facilities, but non-transported passengers spend more money on facilities. To have further analysis, it can extract the total expenditure on those facilities and classify whether passengers are transported or non – transported.

## Categorical Feature

|  |
| --- |
| # Categorical features  cat\_feats=['HomePlanet', 'CryoSleep', 'Destination', 'VIP']  # Plot categorical features  fig=plt.figure(figsize=(10,16))  for i, var\_name in enumerate(cat\_feats):  ax=fig.add\_subplot(4,1,i+1)  sns.countplot(data=train, x=var\_name, axes=ax, hue='Transported')  ax.set\_title(var\_name)  fig.tight\_layout() # Improves appearance a bit  plt.show() |

### HomePlanet

一張含有 文字, 圖表, 數字, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片

自動產生的描述Most of passenger who lives in Earth are not transported while most of passenger who lives in Europa or Mars are more likely to transported.

### Cryosleep

More people who have cryosleep are more likely to transport while less passenger who have cryosleep are less likely to be transported.

### 一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片 自動產生的描述Destination

Most passenger who targets TRAPPIST - 1e are likely to be transported while more passenger who target 55 Cancri e are likely to be transported. Moreover, the transported rate of passenger who target PSO J318.522 are equally.

### 一張含有 文字, 螢幕擷取畫面, 繪圖, 數字 的圖片 自動產生的描述VIP

Most of passenger are not as VIP of spaceship titanic while the transport rate is equally same. It is not useful for classifying data with VIP.

#### Finding

Cryosleep has a large contrast on the transported rate of passenger while the difference of VIP has a smaller contrast, the transported rate of passenger is like each other. However, the transported rate of Destination & Homeplanet has larger contrast. Thus, VIP is less useful for further analysis, and it is estimated to delete this attribute.

# Handle Missing Value

Name, Cabin, PassengerId has dropped before handling missing data because they are neither categorical data nor numerical data.

|  |
| --- |
| # Columns with missing values  na\_cols=data.columns[data.isna().any()].tolist()  # Missing values summary  mv=pd.DataFrame(data[na\_cols].isna().sum(), columns=['Number\_missing'])  mv['Percentage\_missing']=np.round(100\*mv['Number\_missing']/len(data),2)  mv |

|  |
| --- |
| categorical\_cols=['HomePlanet', 'CryoSleep', 'Destination','VIP','Cabin']  numerical\_cols=['Age','RoomService','FoodCourt', 'ShoppingMall', 'Spa', 'VRDeck']  # fill missng data  def fill\_missing(df):  for ft in categorical\_cols:  df[ft].fillna(df[ft].mode()[0],inplace=True)  for ft in numerical\_cols:  df[ft].fillna(df[ft].median(),inplace=True)  return df  fill\_missing(data)  data.isna().any() |

#### 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片 自動產生的描述Findings

The numbers of missing data in column are counted and the percentage of missing data is calculated. Besides, the percentage of missing data in each column is less than 2.5% except the predicted column – “Transported”. It means the percentage of all columns of missing data is also less than 2.5% which is relatively small.

Filling missing value could help to understand the scope, identify problem, and provides better solution for handing them. To handle missing values, mode and median is used to filling the categorical attributes and numerical attributes respectively. It increases the accuracy of prediction by keeping the model analysis.

# Feature Engineering (FE)

What have fixed:

* Size of train set: 0.2
* Handle missing data before feature engineering.

## Age group distribution

|  |
| --- |
| # New features  x = [train, test]  for i in x:  i['Age\_group']=np.nan  i.loc[i['Age']<=9,'Age\_group']='Age\_0-9'  i.loc[(i['Age']>=10) & (i['Age']<=19),'Age\_group']='Age\_10-19'  i.loc[(i['Age']>=20) & (i['Age']<=29),'Age\_group']='Age\_20-29'  i.loc[(i['Age']>=30) & (i['Age']<=39),'Age\_group']='Age\_30-39'  i.loc[(i['Age']>=40) & (i['Age']<=49),'Age\_group']='Age\_40-49'  i.loc[i['Age']>=50 & (i['Age']<=59),'Age\_group']='Age\_50-59'  i.loc[i['Age']>=60 ,'Age\_group']='Age\_60+'  # Plot distribution of new features  plt.figure(figsize=(10,4))  g=sns.countplot(data=n\_train, x='Age\_group', hue='Transported', order=['Age\_0-9','Age\_10-19','Age\_20-29','Age\_30-39','Age\_40-49','Age\_50-59','Age\_60+'])  plt.title('Age group distribution') |

一張含有 文字, 螢幕擷取畫面, 字型, 繪圖 的圖片

自動產生的描述

It separated age group and tried to find out the relationship between the predicted value “transported” and the separated age groups.

#### Findings

The ratio of transported in age group 0 – 9 is higher than rate of non – transported.

## Cabin Distribution

|  |
| --- |
| x = [train, test]  for i in x:  # Cabin - The cabin number where the passenger is staying. Takes the form deck/num/side, where side can be either P for Port or S for Starboard.  i["Cabin"].fillna("np.nan/np.nan/np.nan",inplace=True)  i["Deck"] = i["Cabin"].apply(lambda x: x.split("/")[0])  i["Num"] = i["Cabin"].apply(lambda x: x.split("/")[1])  i["Side"] = i["Cabin"].apply(lambda x: x.split("/")[2])  #missing value to nan  cols = ["Deck","Num","Side"]  i[cols]=i[cols].replace("np.nan",np.nan)  #filling missing value to median  i["Num"].fillna(i["Num"].median(),inplace=True)  #convert to int  i['Num']=i['Num'].astype(int)  sns.countplot(data = n\_train, x = 'Deck', hue = 'Transported', order=['A','B','C','D','E','F','G','T'])  plt.title('Deck group distribution') |

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

It extracted the distribution of decks in the spaceship **individually**.

#### Findings

The deck of “T” might be an outlier because deck “T” has only 5 samples.

## Total Expenditure

|  |
| --- |
| # New features - training set  n\_train['Total Expenditure'] =n\_train['RoomService']+n\_train['FoodCourt']+ n\_train['ShoppingMall']+ n\_train['Spa']+ n\_train['VRDeck']#different between max and min are too big  # New features - test set  n\_test['Total Expenditure'] = n\_test['RoomService']+n\_test['FoodCourt']+ n\_test['ShoppingMall']+ n\_test['Spa']+ n\_test['VRDeck']  # Plot distribution of new features  plt.figure(figsize=(15,6))  sns.histplot(x="Total Expenditure", hue="Transported", data=n\_train, kde=True, palette="Set2",bins=200)  plt.ylim(0,3000)  plt.xlim(0,10000)  plt.title("Total Expenditure Distribution"); |

一張含有 文字, 螢幕擷取畫面, 繪圖, 行 的圖片

自動產生的描述

Refer to EDA, the total expenditure has included the expenditure of room service, food court, shopping mall, Spa, VR deck.

#### 一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片 自動產生的描述Findings

It found that passengers who do not spend money are transported successfully while passengers who spend money has less potentially been transported.

## Group Size

|  |
| --- |
| # New feature - Group  n\_train['Group'] = n\_train['PassengerId'].apply(lambda x: x.split('\_')[0]).astype(int)  n\_test['Group'] = n\_test['PassengerId'].apply(lambda x: x.split('\_')[0]).astype(int)  # New feature - Group size  n\_train['Group\_size']=n\_train['Group'].map(lambda x: pd.concat([n\_train['Group'], n\_test['Group']]).value\_counts()[x])  n\_test['Group\_size']=n\_test['Group'].map(lambda x: pd.concat([n\_train['Group'], n\_test['Group']]).value\_counts()[x])  # Plot distribution of new features  plt.figure(figsize=(20,4))  plt.subplot(1,2,1)  sns.histplot(data=n\_train, x='Group', hue='Transported', binwidth=1)  plt.title('Group')  plt.subplot(1,2,2)  sns.countplot(data=n\_train, x='Group\_size', hue='Transported')  plt.title('Group size')  fig.tight\_layout() |

|  |
| --- |
| # New feature  n\_train['Solo']=(n\_train['Group\_size']==1).astype(int)  n\_test['Solo']=(n\_test['Group\_size']==1).astype(int)  # New feature distribution  plt.figure(figsize=(10,4))  sns.countplot(data=n\_train, x='Solo', hue='Transported')  plt.title('Passenger travelling solo or not')  plt.ylim([0,3000]) |

一張含有 文字, 螢幕擷取畫面, 行, 字型 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

#### 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片 自動產生的描述Findings

Group would not be used due to the reflection of passenger group size. It might have duplicate records because different passenger may belong to same group. From graph of group size, most passenger are travelling alone which allow to have further analyzing the transported value of passenger traveling solo or non-solo. Besides, the number of passengers who travelling solo is larger than the number of passengers who travelling non – solo while the value of transported solo is lower than the value of transported non – solo.

#### Conclusion

After comparing the relation between the transported column and relative column, Age group distribution, Cabin distribution, Total expenditure, Group size and transport group or solo will be selected as feature.

# Preprocessing

## Label Encoding vs One - hot Encoding

|  |
| --- |
| PASSENGER\_ID = n\_test[['PassengerId']]  n\_train.drop(['PassengerId',"Cabin","Group"], axis=1, inplace=True)  n\_test.drop(['PassengerId',"Cabin","Group"], axis=1, inplace=True)  # Here we are applying LabelEncoding in all our categorical cols (Turning categorical into numerical)  categorical\_cols= ['HomePlanet','CryoSleep','Destination','VIP','Deck','Side','Age\_group']  for i in categorical\_cols:  le=LabelEncoder()  arr=np.concatenate((n\_train[i], n\_test[i])).astype(str)  le.fit(arr)  n\_train[i]=le.transform(n\_train[i].astype(str))  n\_test[i]=le.transform(n\_test[i].astype(str)) |

To implement the values into Model, label encoding is used to provide a range value of 0 - (N - 1) and convert categorical feature into numerical. To take an illustration, Homeplanet has three attributes, convert Earth to “0”, Europa to “1”, Mars to “2”. It reduced various features generated when using One hot encoding to classify the categorical feature. It separated the categorical feature into smaller feature and involve 0 / 1 for indicating the value of “Yes / No” which makes the analysis become slower and clumsy. Thus, Label Encoding is used to preprocessing data.

## Validation Dataset

|  |
| --- |
| X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=101, stratify=y) |

Validation dataset is used to estimate whether there is a problem in the train set. It helps to have adjustment on model analysis. Besides, using test sets will affect the accuracy of model analysis.

# Model Analysis

Confusion matrix for analysis will show the performance of the model by using the predicted value and actual value. Four labels of model prediction will be displayed in matrix format, which are true positives, false positives, true negatives, false negatives. True label will present as y-axis and predicted label as the x-axis. True labels represent the actual value, and the predicted labels represent the predicted value.

|  |
| --- |
| classifier = [KNeighborsClassifier(), DecisionTreeClassifier(random\_state=101), RandomForestClassifier(random\_state=101)]  title = ['Kneighbors classifier', 'Decision Tree Classifier', 'Random Forest Classifier']  for i in classifier:  print("\033[1m" + title[classifier.index(i)] + "\033[0;0m")  predictor = i.fit(X\_train, y\_train)  y\_pred = predictor.predict(X\_test)  accuracy\_knn = accuracy\_score(y\_test, y\_pred)  model\_dict[title[classifier.index(i)]] = accuracy\_knn  print(accuracy\_knn)  confusion\_matrix = metrics.confusion\_matrix(y\_test, y\_pred)  cm\_display = metrics.ConfusionMatrixDisplay(confusion\_matrix = confusion\_matrix, display\_labels = [False, True])  cm\_display.plot()  plt.show() |

|  |  |
| --- | --- |
| KNN → Accuracy (0.776308) 一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片  自動產生的描述 | Decision Tree → Accuracy (0.752156) 一張含有 螢幕擷取畫面, 文字, 圖表, 行 的圖片  自動產生的描述 |
| Random forest → Accuracy (0.805635) 一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片  自動產生的描述 | |

#### Conclusion

After scaling a confusion matrix, it would be found random forest has the lowest number of true negative and false positive values. Therefore, random forest would be decided as the most accurate model, and it will be used for prediction the result.

# Result and Conclusion

|  |
| --- |
| # Load the submission file  submission\_df=pd.read\_csv('/content/sample\_submission.csv')  classifier = RandomForestClassifier(random\_state=101)  predictor = classifier.fit(X\_train, y\_train)  y\_pred = predictor.predict(X\_test)  accuracy\_rfc = accuracy\_score(y\_test, y\_pred)  print(accuracy\_rfc)  classifier = predictor  submission = classifier.predict(n\_test)  data = {  'PassengerId': np.array(PASSENGER\_ID).reshape(len(PASSENGER\_ID)),  'Transported': np.array(submission).astype('bool')  }  # Add predictions  submission\_df = pd.DataFrame(data=data).reset\_index()  submission\_df.drop(columns=['index'], inplace=True, axis=1)  submission\_df.head()  plt.pie(submission\_df.Transported.value\_counts(), shadow=True, explode=[.1,.1], autopct='%.1f%%')  plt.title('Transported ', size=18)  plt.legend(['False', 'True'], loc='best', fontsize=12)  plt.show()  # Output to csv  submission\_df.to\_csv("submission\_M1F2.csv", index=False) |

一張含有 文字, 螢幕擷取畫面, 標誌, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 標誌, 字型 的圖片

自動產生的描述After model analysis, random forest is selected as the most accurate model. It is used to predict the result of the distribution of “Transported” in submission dataset:

Besides, it is interested to think about if handling feature engineering before handling missing data. Will there any change on the performance of prediction? To demonstrate the change, a version is developed and stated prediction:

Both csv file is uploaded to Kaggle and result:

一張含有 文字, 螢幕擷取畫面, 字型, 白色 的圖片

自動產生的描述

It can be found the accuracy in handling missing data before handling feature engineering has a better performance on handling feature engineering before handles missing data. The method 2 (handling feature engineering before handles missing data) causes a negative impact on the accuracy of dataset which missing value will affect the actual value of feature engineering. Thus, it decides to use method 1(handling missing data before handling feature engineering) as the prediction.

# Link of Kaggle

A clear workflow on analysis of transported rate in spaceship titanic is stated:

<https://www.kaggle.com/nataouo/space-nhk>