Data Preprocessing

Now, we are done with exploratory data analysis of both training and testing datasets. Now, we should get into preprocessing for both the datasets as some of the features are not numerical.

Importing all packages

In [320…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import *
from sklearn.linear_model import *
from math import *
from sklearn.ensemble import *
from sklearn.feature_selection import *
from sklearn.feature_extraction import *
from sklearn.naive_bayes import *
from sklearn.discriminant_analysis import *
from sklearn.preprocessing import *
from sklearn.metrics import *
from sklearn.neighbors import *
from sklearn.cluster import *
```

Importing all datasets

In [321…

```
df_train = pd.read_csv("train_eda.csv")
df_test = pd.read_csv("test_eda.csv")
```

Displaying first 5 elements of training dataset

In [322…

```
df_train.head()
```

Out[322]:

	index	PassengerId	HomePlanet	Cabin Deck	Cabin Number	Cabin Side	CryoSleep	Destination	Age	VIP
0	0	0001_01	Europa	B	0	P	False	TRAPPIST-1e	27.0	False
1	1	0002_01	Earth	F	0	S	False	TRAPPIST-1e	27.0	False
2	2	0003_01	Europa	A	0	S	False	TRAPPIST-1e	27.0	True
3	3	0003_02	Europa	A	0	S	False	TRAPPIST-1e	27.0	False
4	4	0004_01	Earth	F	1	S	False	TRAPPIST-1e	27.0	False

Displaying first 5 elements of testing dataset

In [323…

```
df_test.head()
```

Out[323]:

	index	PassengerId	HomePlanet	Cabin Deck	Cabin Number	Cabin Side	CryoSleep	Destination	Age	VII
0	0	0013_01	Earth	G	3	S	True	TRAPPIST-1e	26.0	Fals
1	1	0018_01	Earth	F	4	S	False	TRAPPIST-1e	26.0	Fals
2	2	0019_01	Europa	C	0	S	True	55 Cancri e	26.0	Fals
3	3	0021_01	Europa	C	1	S	False	TRAPPIST-1e	26.0	Fals
4	4	0023_01	Earth	F	5	S	False	TRAPPIST-1e	26.0	Fals

Removal of dummy column "index" in both the datasets

In [324…

```
train_1 = df_train.drop("index",axis=1,inplace=False)
test_1 = df_test.drop("index",axis=1,inplace=False)
```

In [325…

```
train_1.head()
```

Out[325]:

	PassengerId	HomePlanet	Cabin Deck	Cabin Number	Cabin Side	CryoSleep	Destination	Age	VIP	Roon
0	0001_01	Europa	B	0	P	False	TRAPPIST-1e	27.0	False	
1	0002_01	Earth	F	0	S	False	TRAPPIST-1e	27.0	False	
2	0003_01	Europa	A	0	S	False	TRAPPIST-1e	27.0	True	
3	0003_02	Europa	A	0	S	False	TRAPPIST-1e	27.0	False	
4	0004_01	Earth	F	1	S	False	TRAPPIST-1e	27.0	False	

In [326…

```
test_1.head()
```

Out[326]:

	PassengerId	HomePlanet	Cabin Deck	Cabin Number	Cabin Side	CryoSleep	Destination	Age	VIP	Roor
0	0013_01	Earth	G	3	S	True	TRAPPIST-1e	26.0	False	
1	0018_01	Earth	F	4	S	False	TRAPPIST-1e	26.0	False	
2	0019_01	Europa	C	0	S	True	55 Cancri e	26.0	False	
3	0021_01	Europa	C	1	S	False	TRAPPIST-1e	26.0	False	
4	0023_01	Earth	F	5	S	False	TRAPPIST-1e	26.0	False	

Checking for unique values in "HomePlanet" feature

In [327…

```
hp_train = train_1["HomePlanet"].unique()
hp_test = test_1["HomePlanet"].unique()
hp_train.sort()
hp_test.sort()
print("Training : ",hp_train)
print("Testing  : ",hp_test)
```

```
Training :  ['Earth' 'Europa' 'Mars']
Testing  :  ['Earth' 'Europa' 'Mars']
```

Performing one-hot encoding for "HomePlanet" feature

In [328…

```
ohe = OneHotEncoder(drop=[["Earth"]])
train_ohe = ohe.fit_transform(train_1["HomePlanet"].to_numpy().reshape(-1,1)).t
test_ohe = ohe.fit_transform(test_1["HomePlanet"].to_numpy().reshape(-1,1)).toa
home_planet_train = pd.DataFrame(train_ohe,columns=["HomePlanet_Europa","HomePl
home_planet_test = pd.DataFrame(test_ohe,columns=["HomePlanet_Europa","HomePlar
```

In [329…

```
train_2 = train_1.copy()
train_2.drop(columns=["HomePlanet"],axis=1,inplace=True)
ctr = 1
for i in home_planet_train:
    train_2.insert(loc=ctr,column=i,value=home_planet_train[i])
    ctr += 1
train_2.head()
```

Out[329]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Deck	Cabin Number	Cabin Side	CryoSleep	De
0	0001_01	1.0	0.0	B	0	P	False	T
1	0002_01	0.0	0.0	F	0	S	False	T
2	0003_01	1.0	0.0	A	0	S	False	T
3	0003_02	1.0	0.0	A	0	S	False	T
4	0004_01	0.0	0.0	F	1	S	False	T

In [330…

```
test_2 = test_1.copy()
test_2.drop(columns=["HomePlanet"],axis=1,inplace=True)
ctr = 1
for i in home_planet_test:
    test_2.insert(loc=ctr,column=i,value=home_planet_test[i])
    ctr += 1
test_2.head()
```

Out[330]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Deck	Cabin Number	Cabin Side	CryoSleep	De
0	0013_01	0.0	0.0	G	3	S	True	T
1	0018_01	0.0	0.0	F	4	S	False	T
2	0019_01	1.0	0.0	C	0	S	True	5
3	0021_01	1.0	0.0	C	1	S	False	T
4	0023_01	0.0	0.0	F	5	S	False	T

Checking for unique values in "Cabin Deck" feature

In [331…

```
cd_train = train_2["Cabin Deck"].unique()
cd_test = test_2["Cabin Deck"].unique()
cd_train.sort()
cd_test.sort()
print("Training : ",cd_train)
print("Testing  : ",cd_test)
```

```
Training :  ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'T']
Testing  :  ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'T']
```

Performing One-Hot Encoding for "Cabin Deck" feature

In [332…

```
ohe = OneHotEncoder(drop=[["A"]])
train_ohe = ohe.fit_transform(train_2["Cabin Deck"].to_numpy().reshape(-1,1)).t
test_ohe = ohe.fit_transform(test_2["Cabin Deck"].to_numpy().reshape(-1,1)).toa
```

```
cabin_deck_train = pd.DataFrame(train_ohe,columns=["Cabin Desk B","Cabin Desk C
cabin_deck_test = pd.DataFrame(test_ohe,columns=["Cabin Desk B","Cabin Desk C",
```

In [333…

```
train_3 = train_2.copy()
train_3.drop("Cabin Deck",axis=1,inplace=True)
ctr = 3
for i in cabin_deck_train:
    train_3.insert(loc=ctr,column=i,value=cabin_deck_train[i])
    ctr += 1
train_3.head()
```

Out[333]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Desk B	Cabin Desk C	Cabin Desk D	Cabin Desk E	Cabin Desk F	C D
0	0001_01	1.0	0.0	1.0	0.0	0.0	0.0	0.0	
1	0002_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
2	0003_01	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0003_02	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0004_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

5 rows × 22 columns

In [334…

```
test_3 = test_2.copy()
test_3.drop("Cabin Deck",axis=1,inplace=True)
ctr = 3
for i in cabin_deck_test:
    test_3.insert(loc=ctr,column=i,value=cabin_deck_test[i])
    ctr += 1
test_3.head()
```

Out[334]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Desk B	Cabin Desk C	Cabin Desk D	Cabin Desk E	Cabin Desk F	C D
0	0013_01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0018_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
2	0019_01	1.0	0.0	0.0	1.0	0.0	0.0	0.0	
3	0021_01	1.0	0.0	0.0	1.0	0.0	0.0	0.0	
4	0023_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

5 rows × 21 columns

Performing One-Hot Encoding for "Cabin Side", "CryoSleep", "VIP", "Transported" features

In [335…

```
train_4 = train_3.copy()
test_4 = test_3.copy()

train_4["Cabin Side"] = train_4["Cabin Side"].map({"P":0,"S":1})
test_4["Cabin Side"] = test_4["Cabin Side"].map({"P":0,"S":1})

train_4["CryoSleep"] = train_4["CryoSleep"].map({False:0,True:1})
test_4["CryoSleep"] = test_4["CryoSleep"].map({False:0,True:1})

train_4["VIP"] = train_4["VIP"].map({False:0,True:1})
test_4["VIP"] = test_4["VIP"].map({False:0,True:1})

train_4["Transported"] = train_4["Transported"].map({False:0,True:1})
```

In [336…

```
train_4.head()
```

Out[336]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Desk B	Cabin Desk C	Cabin Desk D	Cabin Desk E	Cabin Desk F	C D
0	0001_01	1.0	0.0	1.0	0.0	0.0	0.0	0.0	
1	0002_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
2	0003_01	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0003_02	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0004_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

5 rows × 22 columns

In [337…

```
test_4.head()
```

Out[337]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Desk B	Cabin Desk C	Cabin Desk D	Cabin Desk E	Cabin Desk F
0	0013_01	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0018_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2	0019_01	1.0	0.0	0.0	1.0	0.0	0.0	0.0
3	0021_01	1.0	0.0	0.0	1.0	0.0	0.0	0.0
4	0023_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0

5 rows × 21 columns

In [338…

```
dest_train = train_4["Destination"].unique()
dest_test = test_4["Destination"].unique()

print(dest_train)
print(dest_test)
```

```
['TRAPPIST-1e' 'PSO J318.5-22' '55 Cancri e']
['TRAPPIST-1e' '55 Cancri e' 'PSO J318.5-22']
```

Performing One-Hot Encoding for "Destination" features

In [339…

```
ohe = OneHotEncoder(drop="first")
train_dest = ohe.fit_transform(train_4["Destination"].to_numpy().reshape(-1,1))
test_dest = ohe.fit_transform(test_4["Destination"].to_numpy().reshape(-1,1)).t

destination_train = pd.DataFrame(train_dest,columns=["Destination_PSO J318.5-22
destination_test = pd.DataFrame(test_dest,columns=["Destination_PSO J318.5-22",
```

In [340…

```
destination_train.head()
```

Out[340]:

	Destination_PSO J318.5-22	Destination_TRAPPIST-1e
0	0.0	1.0
1	0.0	1.0
2	0.0	1.0
3	0.0	1.0
4	0.0	1.0

In [341…

```
destination_test.head()
```

Out[341]:

	Destination_PSO J318.5-22	Destination_TRAPPIST-1e
0	0.0	1.0
1	0.0	1.0
2	0.0	0.0
3	0.0	1.0
4	0.0	1.0

In [342…

```
train_5 = train_4.copy()
train_5.drop(columns=["Destination"],axis=1,inplace=True)
ctr = 13
for i in destination_train:
    train_5.insert(loc=ctr,column=i,value=destination_train[i])
    ctr += 1
train_5.head()
```

Out[342]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Desk B	Cabin Desk C	Cabin Desk D	Cabin Desk E	Cabin Desk F	C
0	0001_01	1.0	0.0	1.0	0.0	0.0	0.0	0.0	
1	0002_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
2	0003_01	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0003_02	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0004_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

5 rows × 23 columns

In [343…

```
test_5 = test_4.copy()
test_5.drop(columns=["Destination"],axis=1,inplace=True)
ctr = 13
for i in destination_test:
    test_5.insert(loc=ctr,column=i,value=destination_test[i])
    ctr += 1
test_5.head()
```

Out[343]:

	PassengerId	HomePlanet_Europa	HomePlanet_Mars	Cabin Desk B	Cabin Desk C	Cabin Desk D	Cabin Desk E	Cabin Desk F	C
0	0013_01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0018_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
2	0019_01	1.0	0.0	0.0	1.0	0.0	0.0	0.0	
3	0021_01	1.0	0.0	0.0	1.0	0.0	0.0	0.0	
4	0023_01	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

5 rows × 22 columns

In [344…

```
train_5.to_csv("train_preprocessed.csv",index=False)
test_5.to_csv("test_preprocessed.csv",index=False)
```

In [ ]: