ADVANCED MACHINE LEARNING

KAGGLE CHALLENGE: SPACESHIP TITANIC

MODULE: MOD006566

Table of Contents

Background

Problem Description

(Addison Howard, 2022) We are in the year 2912 and there is a mystery to solve in the cosmic. We got a communication from almost 4 light years elsewhere in the cosmic and something is looking not correct. There is an interstellar passenger liner called spaceship Titanic with capacity of 13,000 passengers launched one month before. This spaceship is on its first trip and serves the purpose for taking passengers from our solar system to other three exoplanets which are newly habitable that are orbiting near to the stars. During this trip , while the spaceship is rounding Alpha Centauri and heading towards the first exoplanet called The Torrid 55Cancri E, the spaceship hit the spacetime anomaly and went inside the dust cloud and hidden there. Unfortunately, this spaceship titanic also got to met with same fate that happened 1000 years ago. Due to this collision almost 50% of the spaceship passengers are transported to another dimension than to their respective destinations.

The challenge in hand is to find which passengers are transported by this anomaly. To help with to achieve this task, we have the data of passengers that is retrieved from this spaceship damaged computer systems. From above, it is a classification problem that is to find whether the passenger is transported to another dimension or not.

Literature Survey

(Ekinci, 2018) applied fourteen different machine learning algorithms including artificial neural networks on publicly available famous titanic dataset to analyse the chances of passenger survival. Also, defined the features that are having correlation with passenger survival and other crew. (Whitley, 2015) used tree algorithms to classify the passenger survivals and implemented pruning to find the optimal features that affect the prediction. (Tan, 2022) provided a detailed exploratory data analysis on this new futuristic spaceship dataset with proper visualizations and some data pre-processing to find the optimal features increase the prediction quality. (imarranz, 2022) performed data imputations to fill the null values in the dataset and also implemented ensemble methods to classify the passengers.

In this report, the tasks performed includes detailed data analysis and explaining the results on the spaceship titanic dataset, filling the null values with respect to that column distribution pattern, visualizing each and every feature in the dataset using different visualizations, performing feature selection by removing unnecessary features from the dataset, finding the correlations among features, one hot encoding on the categorical features, scaling the values to get them down to same scale, dividing the dataset into train and testing, implementing basic machine learning algorithms first and analysing the results using metrics, and also implementing advanced machine learning techniques such as ensemble methods, Ada boost, XgBoost , Random Forest and bagging classifiers. Every approach taken above is precisely explained with results. At the end, comparing all the performances to find the best algorithm to predict the test dataset.

Exploratory Data Analysis

Spaceship Titanic dataset is publicly available dataset on Kaggle (Addison Howard, 2022). The train dataset contains 8693 rows and 14 columns in total. These columns are the passenger's information such as Id, Home planet, cryosleep (Boolean variable says whether the passenger is in cryosleep or not), cabin (takes the form of deck/num/side and side again can be port or starboard) , Destination, Age, VIP (Boolean variable says whether the passenger is having VIP status or not), amenities used by passenger such as Room service, Food court, Shopping Mall, Spa and VR Deck, Name and Transported column (prediction variable). This dataset has null values in every column except Id and Transported column. The test dataset contains 4277 rows and 13 columns (same as train dataset) with missing prediction column.

Each and every columns from the dataset are visualised to analyse the patterns in the dataset. From the train dataset, it is found that 49.6% of passengers are transported to alternate dimensions and rest are not transported, 52.9% of passengers are from planet Earth, 24.5% from planet Europa and 20.2% from planet Mars, 62.5% passengers are in cryosleep and rest are not, 68% of passengers are going to planet TRAPPIST-1e , 20.7% to planet 55 Cancri e and 0.9% to PSO J318.5-22. These findings are done using value_counts functions from pandas. The age distribution for the passengers with respect to transported or not is visualised using histplot as shown in the below figure (fig 1.1).

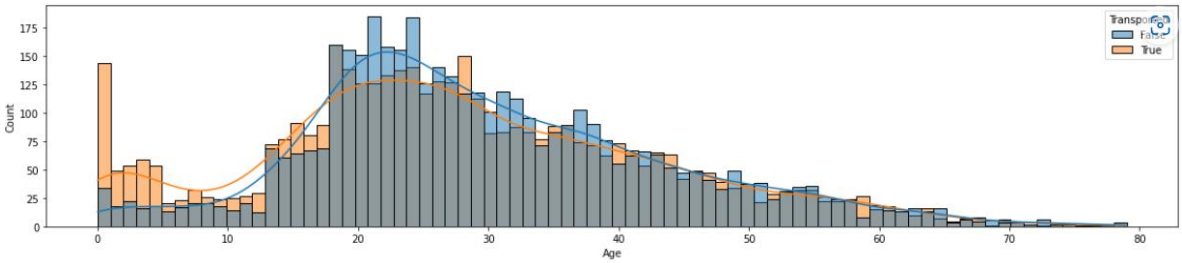


Figure 1.1: Age distribution with hue as Transported

From the above histplot, the passengers more transported are in the age group 0-5 than others and passengers from age group 18-27 are not transported.

Features VIP, cryosleep and home planet with respect to Transported column is shown below in figure (fig 1.2). From this we see that passenger with or without VIP status is transported in equal proportion, passengers in cryosleep are less transported than the passengers not in cryosleep and passengers from planets Europa and Mars are more transported than planet Earth, destination is not having much affect on whether the passenger is transported or not.

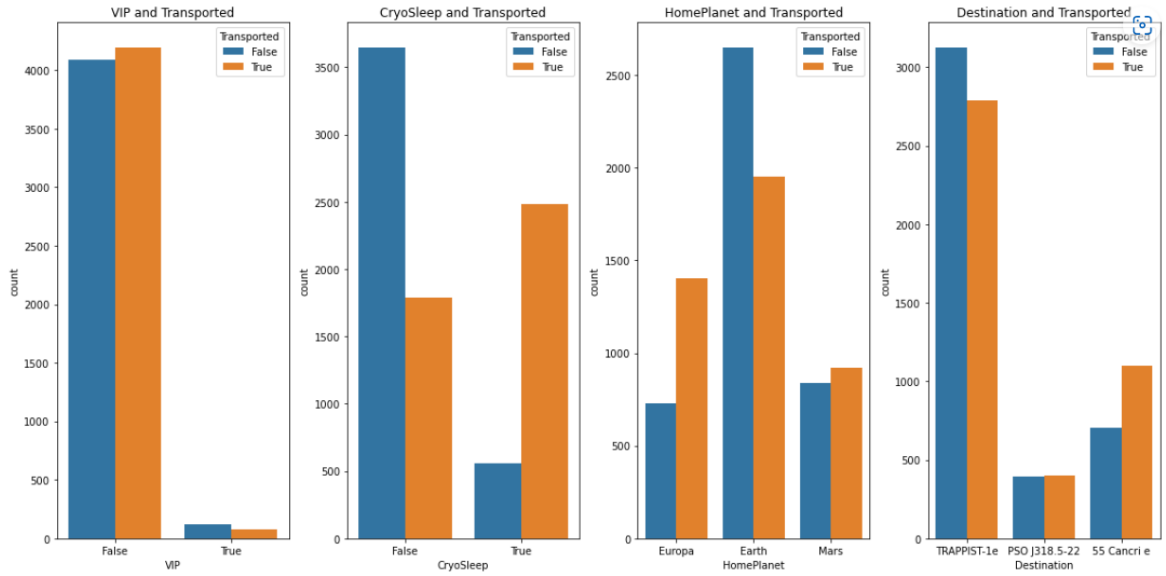
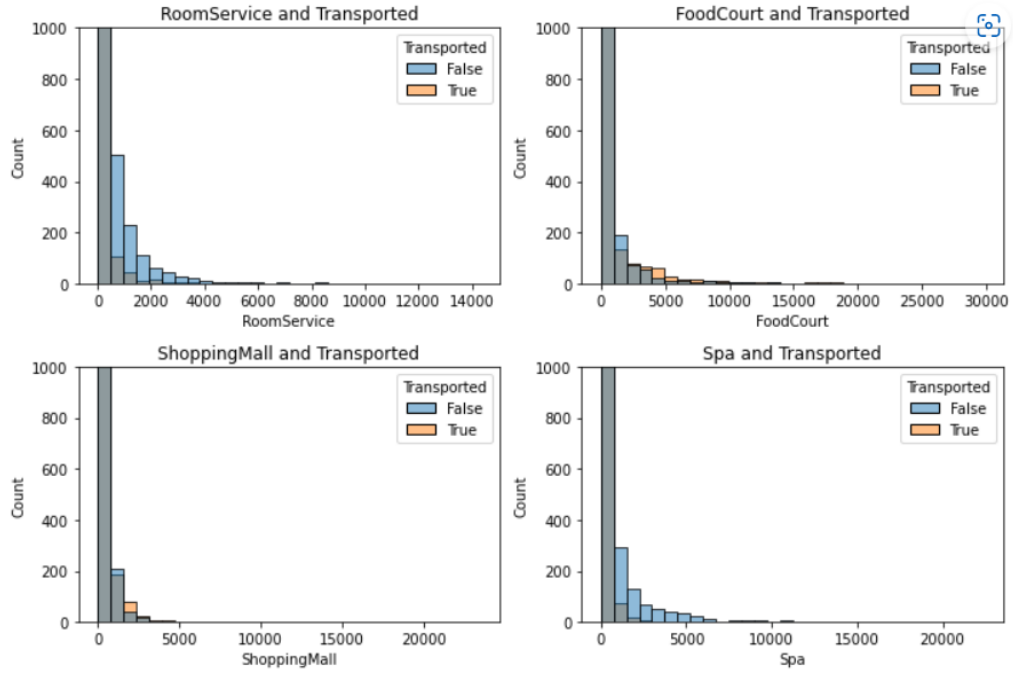


Figure 1.3: VIP, Cryosleep, Home planet, Destination with Transported column

The below figure (fig 1.4) shows the effect of taking amenities and being transported.



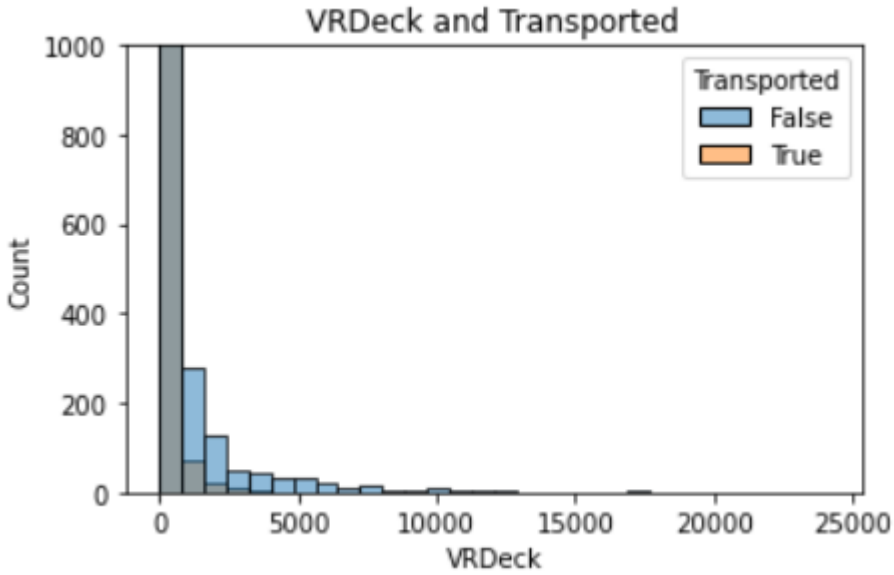


Figure 1.5: Affect of taking amenities and being transported

The above graphs show that, passengers who took room service, spa and VRdeck are not transported than the other two services food court and shopping mall.

These features are also analysed independently to see the distribution of amount spent by the passengers for each service. We can see that most of the passengers didn't spend any money on the services. This is shown in below figure (fig 1.6).

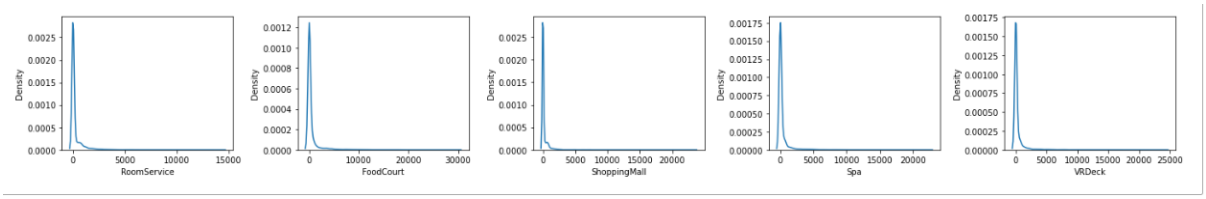


Figure 1.6: Distribution of money spent on services

To see the affect of feature "side" with respect to transported column is shown below (fig 1.7). People on one side are transported more compared to other.

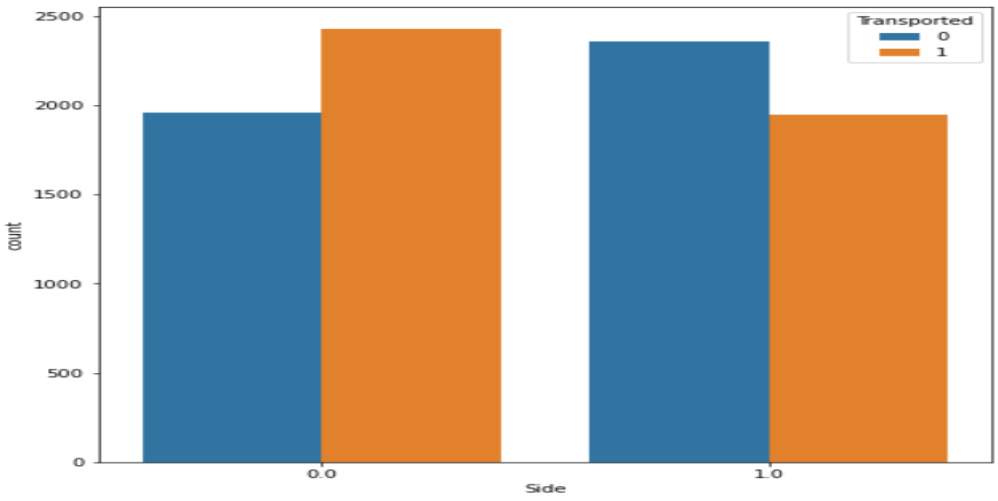


Figure 1.7: Feature side with respect to Transported column

From the below figure (fig 1.8) Passengers who are in cryosleep did not spend any money on the services, this indicates the null values in cryosleep can be determined based on the passenger spent any money the services or not.

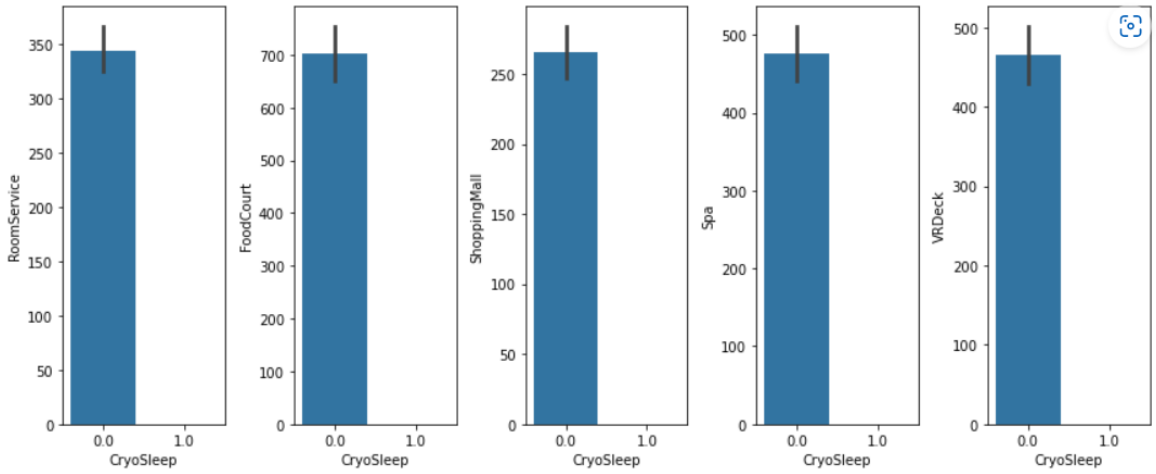


Figure 1.8: Amenities with respect to Cryosleep

Data Pre-processing

In data pre-processing, the first step taken is filling the null values. Firstly, separating the cabin features into Deck and side by eliminating "num" in cabin feature because it is not having any affect on prediction. We can determine that if the values in amenities are null then the passenger didn't spend anything on them, and majority of the passengers didn't spend anything (fig 1.6). So, we can fill all the null values with 0. Null values in VIP feature are filled with False because 95% of passengers are not having VIP status. Null values in Age column are filled with median of Age. Number is extracted from the passenger Id column as it takes the form "group_number".

All the Boolean features such as VIP, Cryosleep, transported are changed to 1 (True) and 0 (False). Null values in cryosleep feature are determined as if the value is NaN and passenger didn't spend any money on amnesties then they are in cryosleep else they are not in cryosleep. This is achieved by defining a function "cryosleepNa". Null values for columns Deck, Destination, Home planet and side are filled according to the proportions of them. For example, null values for home planet column are determined by taking percentage of each value such as, 54.1% of passengers are from Earth, 25% from Europa and 20.7% from Mars. The null values are filled according to this proportions randomly. Then Dummies are created for columns home planet, Destination and Deck using pandas get_dummies method. We drop the cabin features as we extracted the deck and side from it and also name, passenger Id columns from the dataset. After all these steps, the dataset that is ready to for fit into model is shown below (fig 1.10).

	CryoSleep	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck	Transported	Side	...	Mars	PSO J318.5-22	TRAPPIST-1e	B	C	D	E	F	G	T
0	0.0	39.0	0	0.0	0.0	0.0	0.0	0.0	0	1.0	...	0	0	1	1	0	0	0	0	0	0
1	0.0	24.0	0	109.0	9.0	25.0	549.0	44.0	1	0.0	...	0	0	1	0	0	0	0	1	0	0
2	0.0	58.0	1	43.0	3576.0	0.0	6715.0	49.0	0	0.0	...	0	0	1	0	0	0	0	0	0	0
3	0.0	33.0	0	0.0	1283.0	371.0	3329.0	193.0	0	0.0	...	0	0	1	0	0	0	0	0	0	0
4	0.0	16.0	0	303.0	70.0	151.0	565.0	2.0	1	0.0	...	0	0	1	0	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8688	0.0	41.0	1	0.0	6819.0	0.0	1643.0	74.0	0	1.0	...	0	0	0	0	0	0	0	0	0	0
8689	1.0	18.0	0	0.0	0.0	0.0	0.0	0.0	0	0.0	...	0	1	0	0	0	0	0	0	1	0
8690	0.0	26.0	0	0.0	0.0	1872.0	1.0	0.0	1	0.0	...	0	0	1	0	0	0	0	0	1	0
8691	0.0	32.0	0	0.0	1049.0	0.0	353.0	3235.0	0	0.0	...	0	0	0	0	0	0	1	0	0	0
8692	0.0	44.0	0	126.0	4688.0	0.0	0.0	12.0	1	0.0	...	0	0	1	0	0	0	1	0	0	0

8693 rows × 22 columns

Figure 1.10: Dataset after pre-processing

To find the correlations among the features, we plot a heat map that shows all the correlations with respect to every other features. The below figure (fig 1.11) shows the correlation matrix.

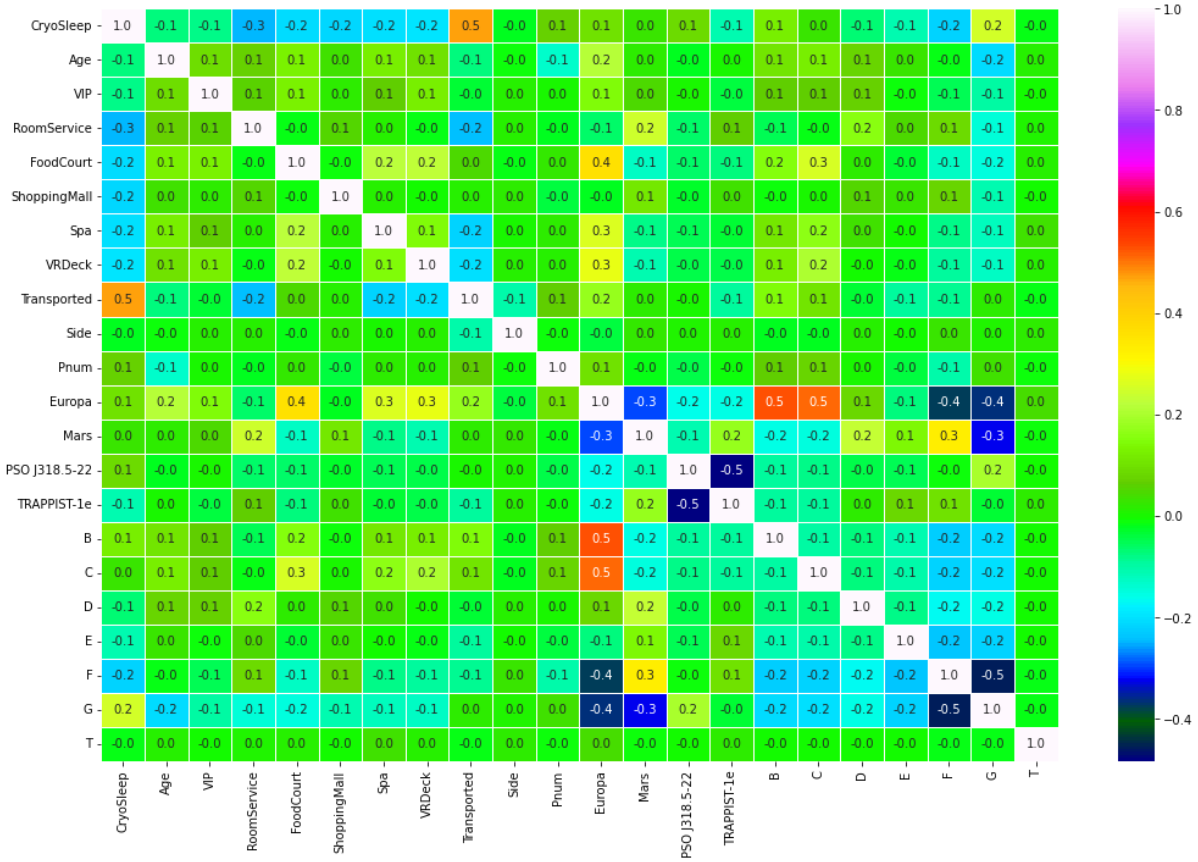


Figure 1.11: Correlation among features

We can see that cryosleep feature has high positive corelation with Transported column and also Deck B, C are highly correlated with Europa planet. And Deck F, G are highly negatively corelated with each other and with Europa planet.

Legal, Ethical and Privacy

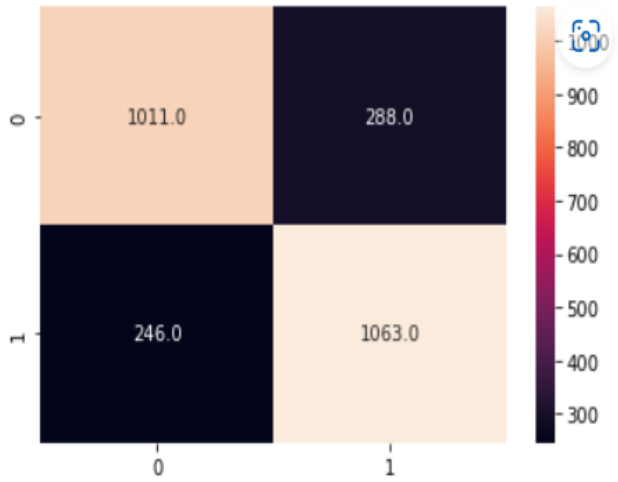
The dataset used in this report is publicly available on Kaggle platform. Before downloading the dataset from platform, we must accept all the compliance rules defined. Data ethics are followed while dealing with this dataset without any data leakage. The data in this dataset is sensitive even though it was not the incident happened in real but an imaginary thing for now. There are privacy concerns included in this dataset. If the incident happened in real, then GDPR compliance is must before proceeding with this dataset.

Machine Learning Approaches

In this report we will discuss varied basic and some advanced machine learning techniques applied on this spaceship titanic dataset and will analyse the performances with each other. The task in hand is a classification task, so we will be using supervised classification algorithms. In every approach I have K Fold validation technique because there are no labels for test data, so that if cross validation is performed then the resulting model will be robust to make predictions. The train dataset is divided into train and validation (test) set for model building in all techniques. In every technique we focus on reducing the False positives and negatives and we will find the best model that has least of them.

Logistic Regression

In this technique, the model is built with parameters such as “liblinear” which is optimization algorithm chosen for small datasets, max iterations being 1000 and random state with some random integer. Here we used grid search that includes grid params such as penalty, cross validation and using all the processor cores to run the model. This model fits 10 folds for each of 20 candidates, totalling 200 fits. After model fitting, the observed train and validation accuracy 78.7% and 79.5%. There is not much difference in the metrics, all are close to each other.



	precision	recall	f1-score	support
0	0.80	0.79	0.79	1299
1	0.79	0.80	0.80	1309
accuracy			0.80	2608
macro avg	0.80	0.80	0.80	2608
weighted avg	0.80	0.80	0.80	2608

Figure 1.12: confusion matrix and classification report for Logistic regression

NaiveBaye's

This algorithm takes the default parameters and fits the model with train dataset. The train and validation accuracy observed are 70.9% and 71.62%. From all the models performed in this report, this model is under performed of all. The recall and f1 scores are very low when compared to other model's metrics.

	precision	recall	f1-score	support
0	0.80	0.81	0.80	1299
1	0.81	0.80	0.80	1309
accuracy			0.80	2608
macro avg	0.80	0.80	0.80	2608
weighted avg	0.80	0.80	0.80	2608

Figure 1.13: Classification report for Naïve Baye's

Support Vector Machine

In this technique, we use grid search with parameters estimator as svm and grid parameters (poly and sigmoid as kernels, and some gamma and C values, cross validation (kfold) and all processor cores). This model fits 10 folds for each of 54 candidates, totalling to 540 fits. The train and validation accuracy achieved are 78.3% and 80.2%. The best estimator is SVC (C=10, degree=2, gamma=1, kernel='poly') and best params are {'C': 10, 'degree': 2, 'gamma': 1, 'kernel': 'poly'}.

	precision	recall	f1-score	support
0	0.80	0.81	0.80	1299
1	0.81	0.80	0.80	1309
accuracy			0.80	2608
macro avg	0.80	0.80	0.80	2608
weighted avg	0.80	0.80	0.80	2608

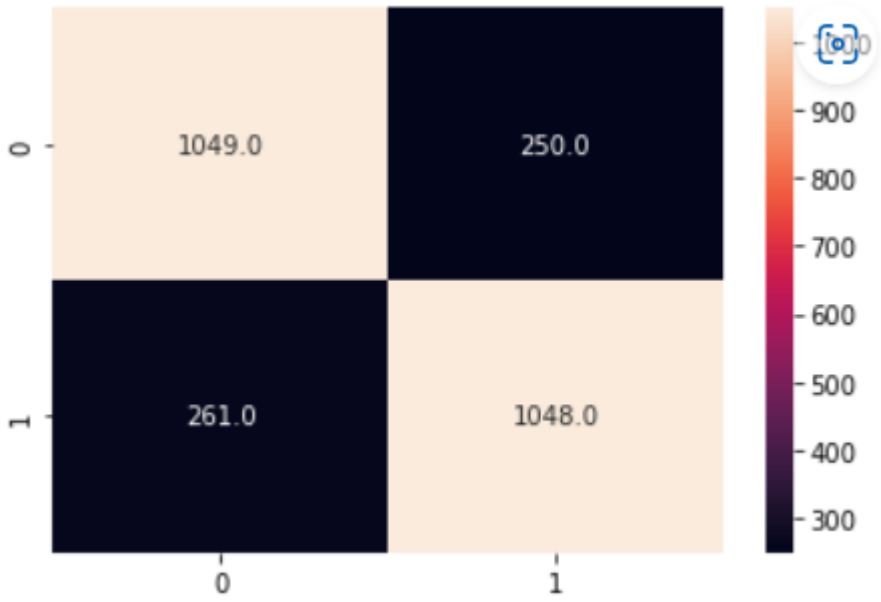


Figure 1.13: confusion matrix and classification report for SVM

K-Nearest-Neighbour

In this technique, the grid search parameters are knn as estimator, weights (uniform and distance), algorithms (auto, ball_tree, kd_tree, brute), n_neighbours from 1 to 50, cross validation (kfold). The best estimators are (algorithm='ball_tree', n_neighbors=16) and best params are {'algorithm': 'ball_tree', 'n_neighbors': 16, 'weights': 'uniform'}. The train and validation accuracy achieved are 75.1% and 76.2%.

	precision	recall	f1-score	support
0	0.72	0.86	0.78	1299
1	0.82	0.67	0.74	1309
accuracy			0.76	2608
macro avg	0.77	0.76	0.76	2608
weighted avg	0.77	0.76	0.76	2608

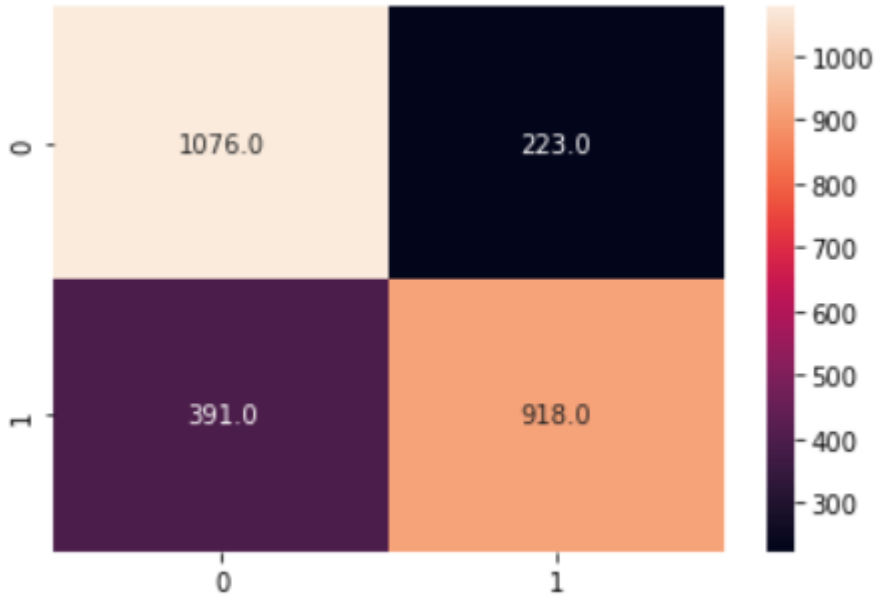


Figure 1.14 confusion matrix and classification report for KNN

XgBoost

This algorithm is built with parameters such as gamma=0, learning_rate= 0.06, max_depth= 10, n_estimators= 300, cross validation. The train and validation accuracy achieve are 79.1% and 80.06%. All the metrics shows good values from the below classification report.

	precision	recall	f1-score	support
0	0.80	0.81	0.80	1299
1	0.81	0.80	0.80	1309
accuracy			0.80	2608
macro avg	0.80	0.80	0.80	2608
weighted avg	0.80	0.80	0.80	2608

Figure 1.15: classification report for xgboost

Gradient Boosting Classifier

This algorithm is built with grid search and parameters including different estimators and learning rates, cross validation, and all processor cores. It fits 10 folds for each of 64 candidates, totalling 640 fits. The best estimators are (max_depth=4, random_state=42) and best params are {'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 100}. The train and validation accuracy are 79.8% and 80%. Both xgboost and gradient boost are having same performance.

	precision	recall	f1-score	support
0	0.83	0.77	0.80	1299
1	0.79	0.85	0.82	1309
accuracy			0.81	2608
macro avg	0.81	0.81	0.81	2608
weighted avg	0.81	0.81	0.81	2608

Figure 1.16: classification report for Gradient Boosting classifier

Decision Tree

This algorithm is built with grid search and parameters including, criterion as ['Gini', 'entropy'] splitter as ['best', 'random'], cross validation, max depth from 1 to 10 and max_features as [0.5,0.7,1]. It fits 10 folds for each of 120 candidates, totalling 1200 fits. The best parameters are {'criterion': 'entropy', 'max_depth': 9, 'max_features': 0.5,'splitter': 'best'} and best estimator is (criterion='entropy', max_depth=9, max_features=0.7, random_state=41). The train and validation accuracy achieved are 78.1% and 78.6%.

	precision	recall	f1-score	support
0	0.78	0.78	0.78	1299
1	0.78	0.79	0.79	1309
accuracy			0.78	2608
macro avg	0.78	0.78	0.78	2608
weighted avg	0.78	0.78	0.78	2608

Figure 1.17: classification report for Decision Tree

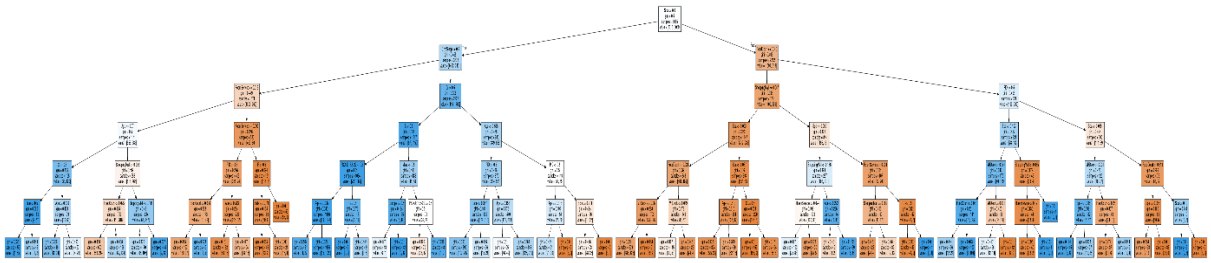


Figure 1.18: Decision Tree obtained from model.

Bagging Classifier

This approach used grid search with parameters including n_estimators as [100, 200, 400, 600] max_samples as [1, 2, 4, 8, 16, 32] and cross validation. It fits Fitting 10 folds for each of 24 candidates, totalling 240 fits. The best estimator is observed with max_samples=32, n_estimators=100 and best parameters are max_samples=32, n_estimators=400. The train and validation accuracy achieved are 77.2% and 76.6%.

	precision	recall	f1-score	support
0	0.78	0.80	0.79	1299
1	0.79	0.77	0.78	1309
accuracy			0.78	2608
macro avg	0.79	0.78	0.78	2608
weighted avg	0.79	0.78	0.78	2608

Figure 1.19: classification report for bagging classifier

AdaBoost classifier

This approach with grid search takes the parameters such as n_estimators as [100, 200, 400, 600], learning_rate as [0.001, 0.01, 0.1, 1], algorithm as ['SAMME', 'SAMME.R'] and cross validation. It fits Fitting 10 folds for each of 32 candidates, totalling 320 fits.

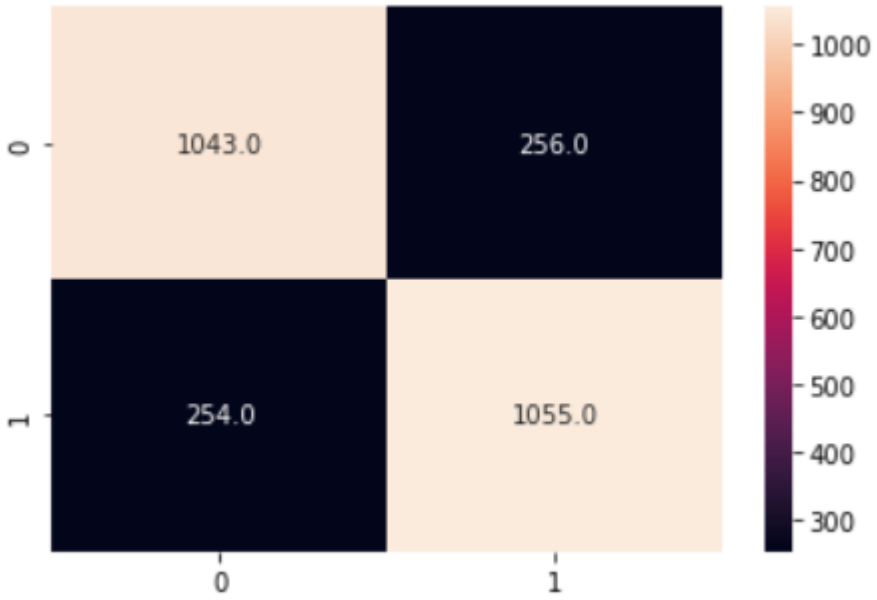
The best estimators are learning_rate=1, n_estimators=400, random_state=42 and best params are 'algorithm': 'SAMME.R', 'learning_rate': 1, 'n_estimators': 400. The train and validation accuracy achieved are 78.70% and 79.44%.

	precision	recall	f1-score	support
0	0.82	0.76	0.79	1299
1	0.78	0.83	0.80	1309
accuracy			0.79	2608
macro avg	0.80	0.79	0.79	2608
weighted avg	0.80	0.79	0.79	2608

Figure 1.20: classification report for AdaBoost.

Random Forest

This algorithm with grid search takes the parameters such as n_estimators as [100, 200, 400, 600], max depth from 1 to 8 and cross validation. It fits Fitting 10 folds for each of 32 candidates, totalling 320 fits. The best estimators are (max_depth=8, n_estimators=200, random_state=42) and params are (max_depth=8, n_estimators=200, random_state=42). The train and validation accuracy achieved are 79.2% and 80.44%. From the above all techniques random forest and gradient boost has best performance on this dataset. This is show in below figure fig 1.22. False positives and negatives in these models are less compared all other models.



	precision	recall	f1-score	support
0	0.80	0.80	0.80	1299
1	0.80	0.81	0.81	1309
accuracy			0.80	2608
macro avg	0.80	0.80	0.80	2608
weighted avg	0.80	0.80	0.80	2608

Figure 1.21: confusion matrix and classification report for Random Forest

]:

	Train_Accuracy	Validation_Accuracy
Logistic Regression	78.685426	79.524540
Naive Bayes	70.913086	71.625767
Support vector machine	78.406010	80.406442
Decision Tree	78.142690	78.642638
EXtra Gradient Boost	78.619177	80.214724
Gradient Boosting Classifier	79.687500	81.096626
Bagging Classifier	77.255180	76.687117
RandomForest	79.244015	80.444785
K-nearest neighbour	75.069895	76.457055
ada boost classifier	78.701603	79.447853

Figure 1.22: performances of all the models performed on this dataset.

Kaggle challenge submission proof

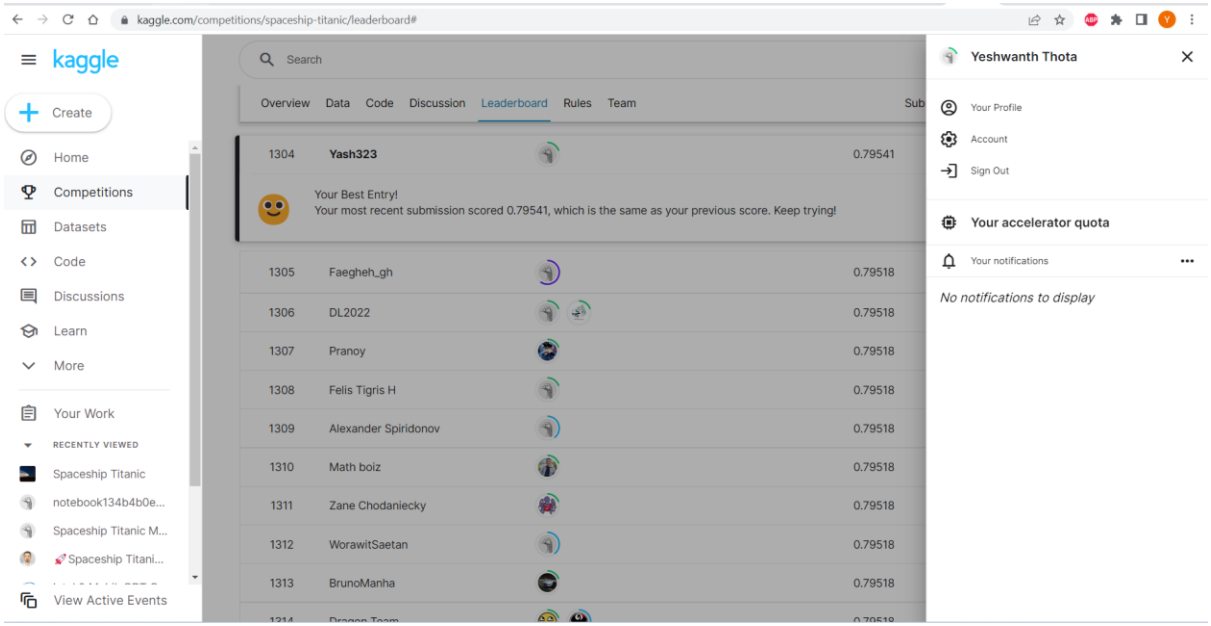


Figure: Leader board in Kaggle

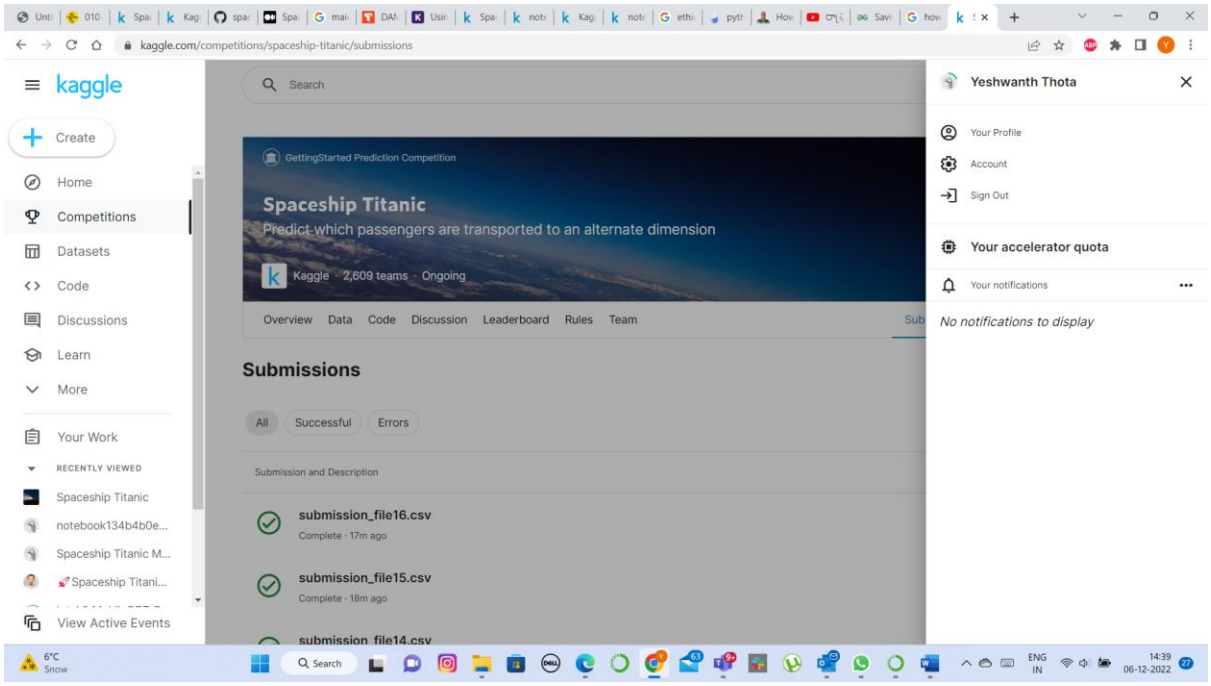


Figure: Submission into Kaggle

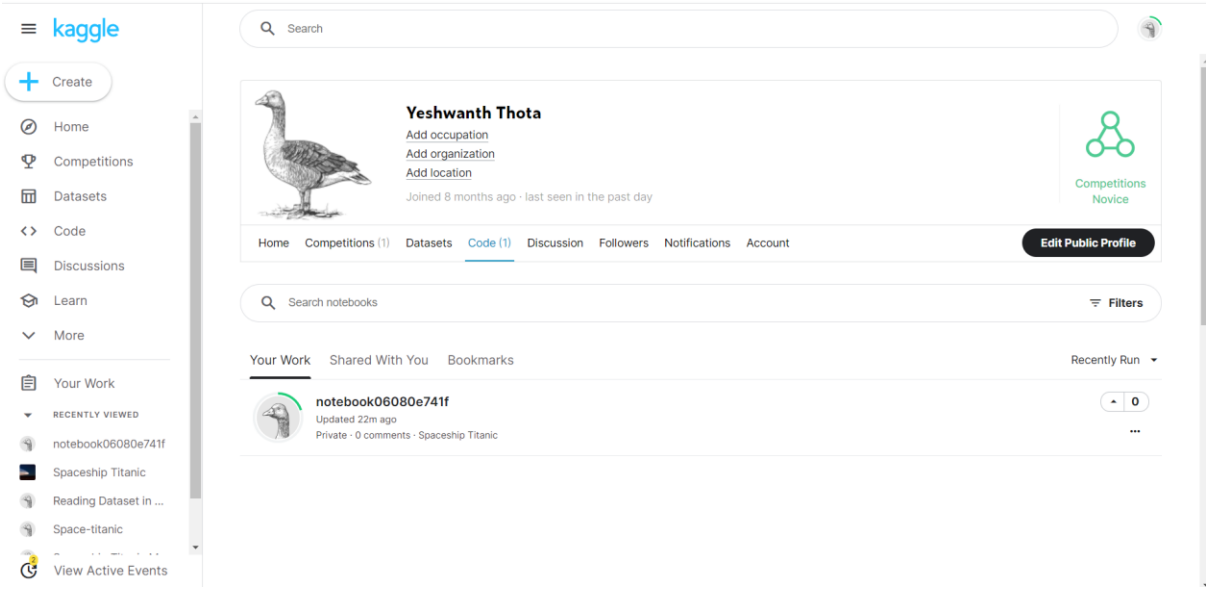


Figure: code uploaded to Kaggle

Result

In this report, detailed data analysis, data pre-processing and used different machine learning models with hyper parameter tuning on spaceship titanic dataset. Random forest and gradient boost classifier models are having best performance among all the other models as shown in table (fig 1.22). Due to this reason, these two models to predict the test dataset and submitted the prediction file to Kaggle.

Future work

The future work in this task includes implementing further advanced techniques and implementing artificial neural networks on this dataset to analyse how neural networks deals the problem and getting insights from it. With this we can also try to improve the model performance and reduce the type1 and type2 errors to get the optimal results.

References

Addison Howard, A. C.-k. (2022, December 05). *Spaceship Titanic*. Retrieved from Kaggle: https://kaggle.com/competitions/spaceship-titanic

Ekinci, E. a. (2018). *A comparative study on machine learning techniques using Titanic dataset.* 7th international conference on advanced technologies.

imarranz. (2022, december 05). *spaceship-titanic*. Retrieved from github: https://github.com/imarranz/spaceship-titanic/tree/master/notebooks

Tan, E. (2022, December 05). *mlearning-ai*. Retrieved from Medium: https://medium.com/mlearning-ai/spaceship-titanic-an-alternative-to-the-plain-old-titanic-dataset-aa98924c606c

Whitley, M. A. (2015). Using statistical learning to predict survival of passengers on the RMS Titanic. *Kansas State University*.