“Cruise Ship Purchase” Exercise

This project is about predicting the crew size of cruise ships based on multiple variables.

1. Exploring the dataset (EDA)

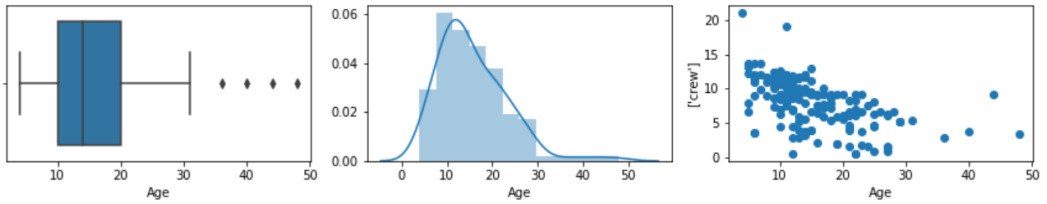
The dataset contained 158 samples, 8 features, and one target variable which was the crew size. Displaying the dataset values showed that among the provided variables, two of them were categorical (i.e., “*ship_name*” & “*cruise_line*”) and the rest were of numerical format.

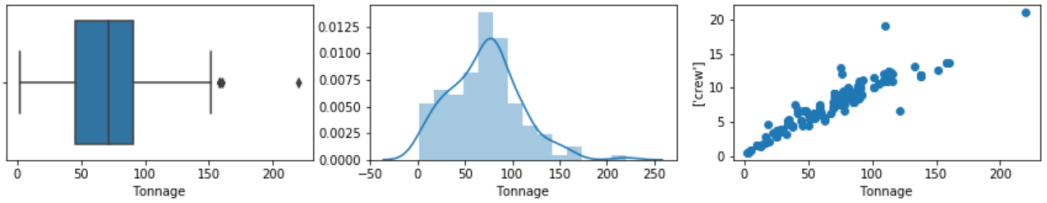
	Ship_name	Cruise_line	Age	Tonnage	passengers	length	cabins	passenger_density	crew
0	Journey	Azamara	6	30.277	6.94	5.94	3.55	42.64	3.55
1	Quest	Azamara	6	30.277	6.94	5.94	3.55	42.64	3.55
2	Celebration	Carnival	26	47.262	14.86	7.22	7.43	31.80	6.70
3	Conquest	Carnival	11	110.000	29.74	9.53	14.88	36.99	19.10
4	Destiny	Carnival	17	101.353	26.42	8.92	13.21	38.36	10.00

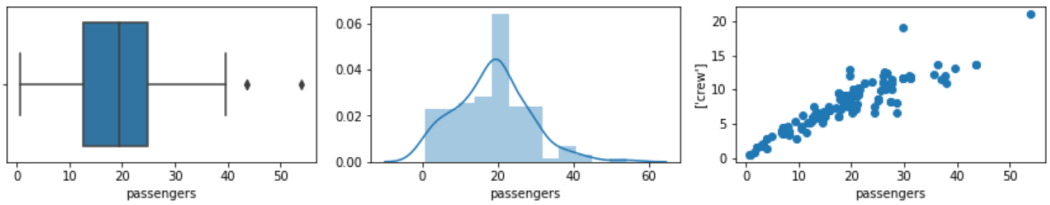
Summarization of the numerical variables illustrated that there was no missing value, while more investigations from their summary stats demonstrated that the provided values were within a reasonable range and no corrupt data was recorded.

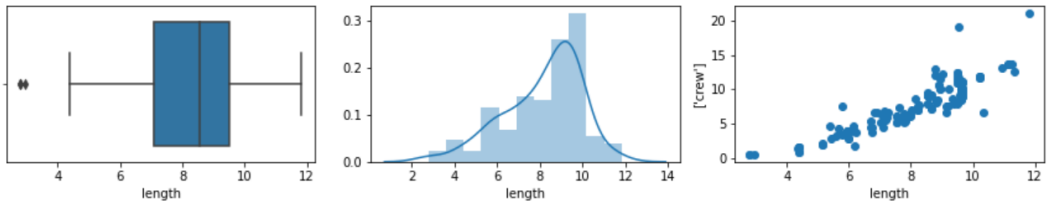
	Age	Tonnage	passengers	length	cabins	passenger_density	crew
count	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000
mean	15.689873	71.284671	18.457405	8.130633	8.830000	39.900949	7.794177
std	7.615691	37.229540	9.677095	1.793474	4.471417	8.639217	3.503487
min	4.000000	2.329000	0.660000	2.790000	0.330000	17.700000	0.590000
25%	10.000000	46.013000	12.535000	7.100000	6.132500	34.570000	5.480000
50%	14.000000	71.899000	19.500000	8.555000	9.570000	39.085000	8.150000
75%	20.000000	90.772500	24.845000	9.510000	10.885000	44.185000	9.990000
max	48.000000	220.000000	54.000000	11.820000	27.000000	71.430000	21.000000

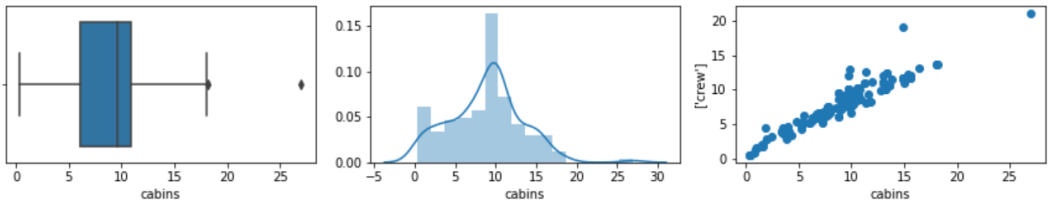
The summary statistics of the numerical features and plots of their distributions indicated that all the variables had roughly normal distribution, while most of them were right-skewed and the “*length*” variable was left-skewed. Except two variables (i.e., “*Age*” and “*passenger_density*”), all other features showed strong linear correlation with the target variable.

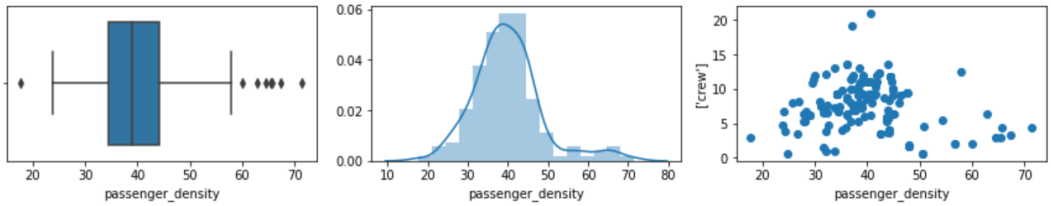

Age
Age
['crew']
Age

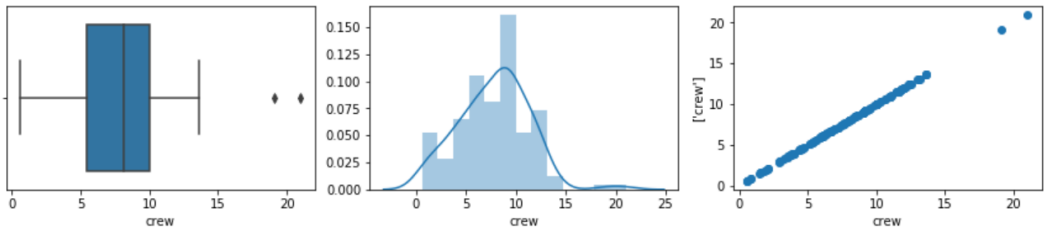

Tonnage
Tonnage
['crew']
Tonnage


passengers
passengers
['crew']
passengers

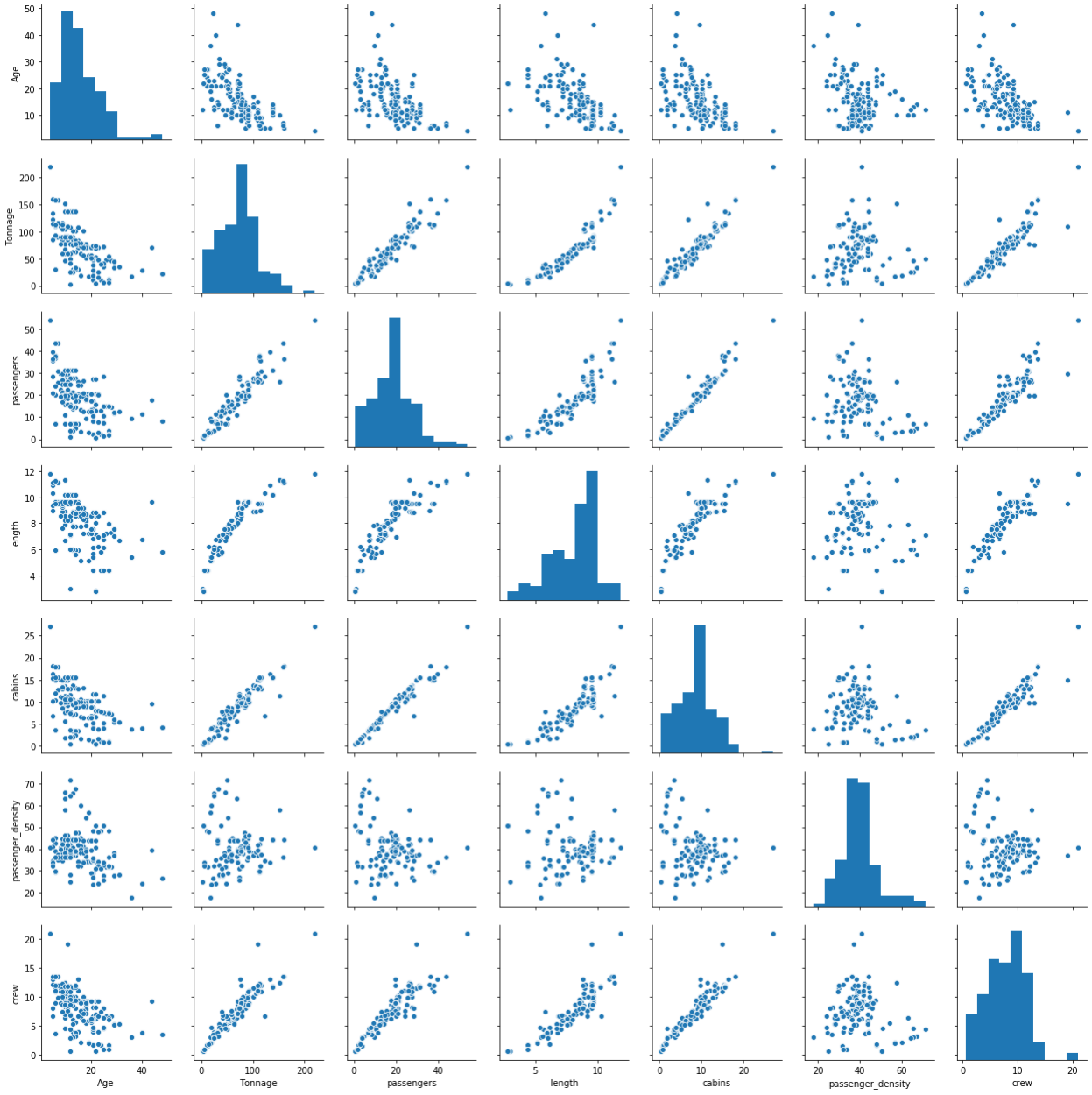

length
length
['crew']
length

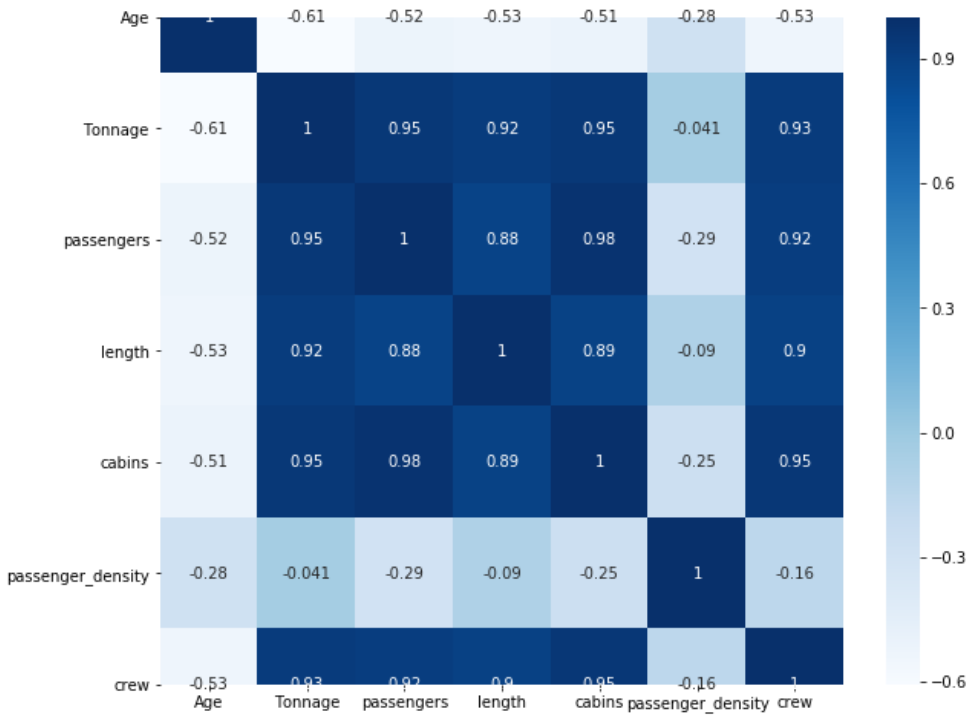

cabins
cabins
['crew']
cabins


passenger_density
passenger_density
['crew']
passenger_density


crew
crew
['crew']
crew

The pair plot and the heatmap of Pearson correlation coefficients from the numerical features also revealed that there were strong linear correlations among each pair of variables (except "*Age*" and "*passenger_density*"). Therefore linear_regression algorithm could not be used for modeling this data due to violation of "no multicollinearity" assumption of this algorithm.





Summarizing the categorical variables also showed that the "*ship_name*" parameter was almost unique for each sample (comparing its number of unique values with the size of dataset) which indicated it would not add useful information to the predictive model. However, the "*cruise_line*" variable was more likely to be useful for modeling.

	Ship_name	Cruise_line
count	158	158
unique	138	20
top	Spirit	Royal_Caribbean
freq	4	23

2. Data preprocessing

The only step for the data preprocessing phase was to convert the "*cruise_line*" variable to a set of dummy variables. This was done using a simple and useful method in pandas library rather than using more complex LabelEncoding & OneHotEncoding methods from sklearn.

3. Feature selection

I decided to remove "*Age*" and "*passenger_density*" variables for modeling due to their low correlations with respect to the target variable. I also removed the "*ship_name*" parameter because of its high number of unique values which most probably could not add useful information to the model.

4. Building predictive models

Train-test split

The dataset was split into training and test sets (with ratios of 0.6/0.4) for proper testing of models. Scaling of the variable values could be performed before modeling; however, I skipped this part for the sake of simplicity. Moreover, based on the range of numerical values and the type of models, this would not mainly affect the models' performances.

Model selection

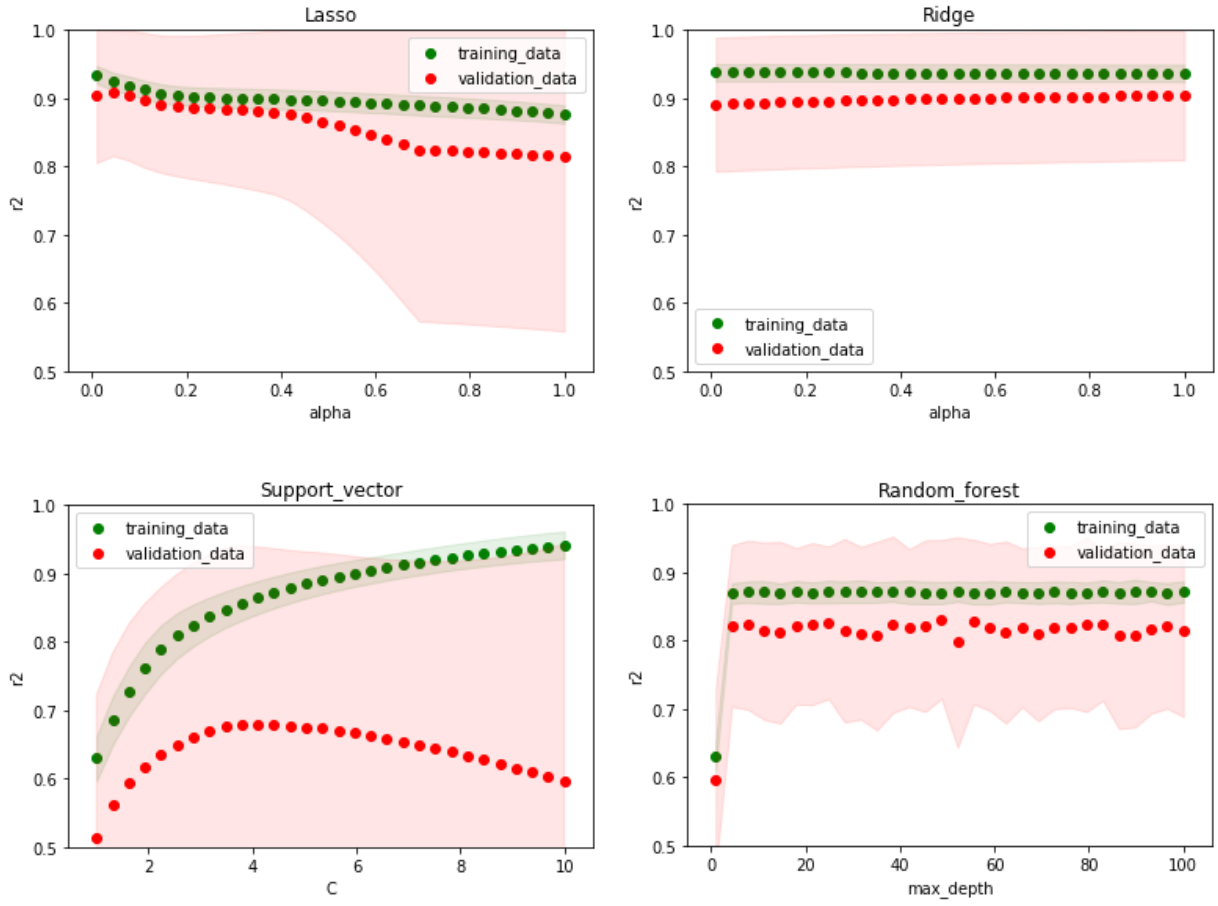
I selected four regression algorithms (i.e., Lasso, Ridge, Support vector regressor, and Random forest) for building a predictor model in this project. One regularization parameter from each model was selected and its changing effect on the model performance was investigated through tracking the changes in the R-squared metric.

There is always a trade-off between bias and variance of predictive models which can result in overfitting or underfitting problems and adversely affect the predictive performances. By increasing the model's complexity, variance usually increases while the bias decreases. As a result, the model performs well in predicting training data, however it can perform very poorly for unseen datasets (this is called overfitting problem due to picking the noises from training data). Too simple models, on the other hand, will have high bias and low variance due to their poor predictability (this is called underfitting problem as the model cannot properly capture the trend from data). Regularization is a process for tuning models to compromise between bias & variance and maximize their predictive performances. Changing the regularization parameter will help in this regard by minimizing both bias and variance.

For "Lasso" and "Ridge" models, the "*alpha*" parameter is the regularization parameter and it can take any value between o and 1. In "Support vector" model, the "*C*" parameter works for regularization and it can take any positive value. In "Random forest" model, there are multiple regularization parameters (i.e., *'max_depth'*, *'min_samples_split'*, *'min_samples_leaf'*, *'max_features'*) from which I selected the "*max_depth*" and it could take any positive value.

I investigated the bias_variance trade_off within the models through plotting the changes in their regularization parameter versus the variations in the "*R_squared*" metric. The bias_variance plots showed that "SVR" model strongly suffered from both high bias & variance, while the "Lasso" model had high variance problem. The remained models performed better with very low bias and some variance levels.

It should be noted that using cross_validation for all the models was helpful in reducing the overfitting (or high variance) problem. I selected the "Random_forest" model as best performing model for this problem due to its robustness for overfitting. As an ensemble type of algorithm, using this model reduces the chance of overfitting.

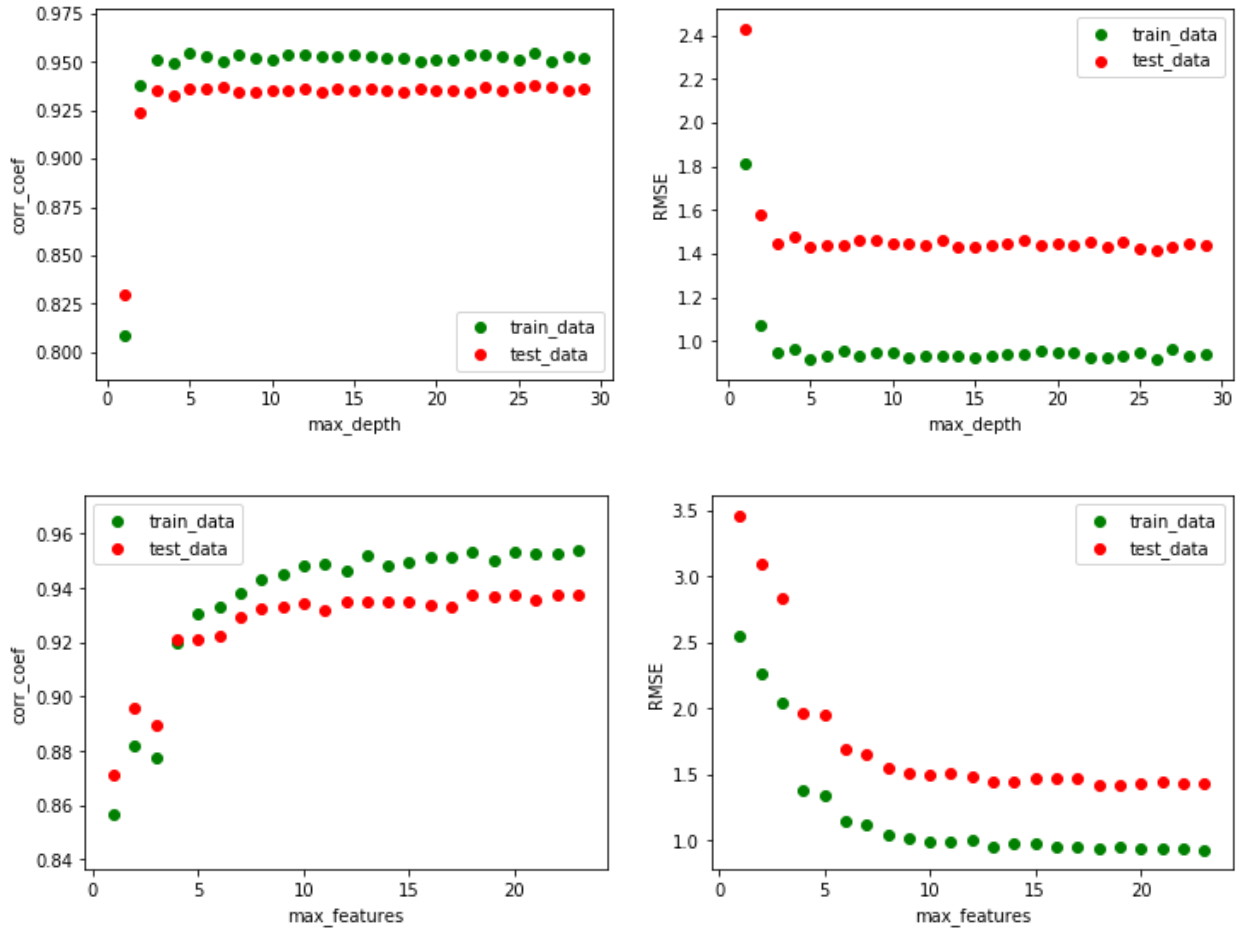


Model tuning

Using grid-search technique, hyperparameter of the selected model were roughly tuned to improve the model performance. *'max_depth'*, *'min_samples_split'*, *'min_samples_leaf'*, and *'max_features'* are the most important hyperparameters of "Random forest" models. The optimized values for the hyperparameters of the selected model are listed below:

max_depth: 50, **max_features**: 10, **min_samples_leaf**: 5, **min_samples_split**: 5 , **n_estimators**: 50

Two of the regularization parameters of this model (i.e., *'max_depth'* & *'max_features'*) were also selected as an example to check their effects on the "*Pearson correlation coefficient*" and the "*Root mean squared error*" as two metrics for measuring models' performances.



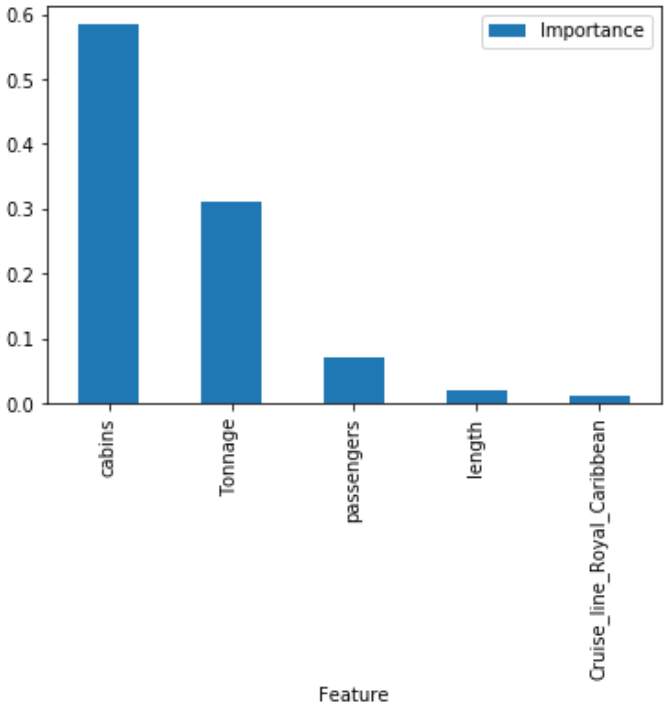
The results showed that the "*max_depth*" and the "max_features" parameters should take the values higher than 5 and 10, respectively in order to achieve best model performance.

Model's efficacy

For measuring the model's efficacy, the selected model was trained on the whole training set using optimized hyperparameters and was tested using "*R_squared*" and "*RMSE*" metrics. Comparing the "*RMSE*" value with the standard deviation and the average value of the target parameter indicated a good model performance.

RMSE_score: 1.42, **r2_score**: 0.87
target_test_mean: 7.62, **target_test_std**: 4.05

Finally, the five most important features on predicting the crew size of cruise ships were determined and their importance values were calculated and plotted as a bar_chart. This plot showed that "*cabins*" and "*Tonnage*" were by far the most important variables in determining the crew size of the ships, while the "*cruise_line*" was the least important one.



Conclusion

In this project a regression model for predicting the crew size of cruise ships based on multiple variables was created and effectively tuned. The predictive features were carefully selected by checking their correlation strength with the target variable. The regularization effects on the models' performances were thoroughly investigated by tracking the changes in "*R_squared*" and "*RMSE*" metrics, and the "*Pearson correlation coefficient*" of the training and the test datasets.