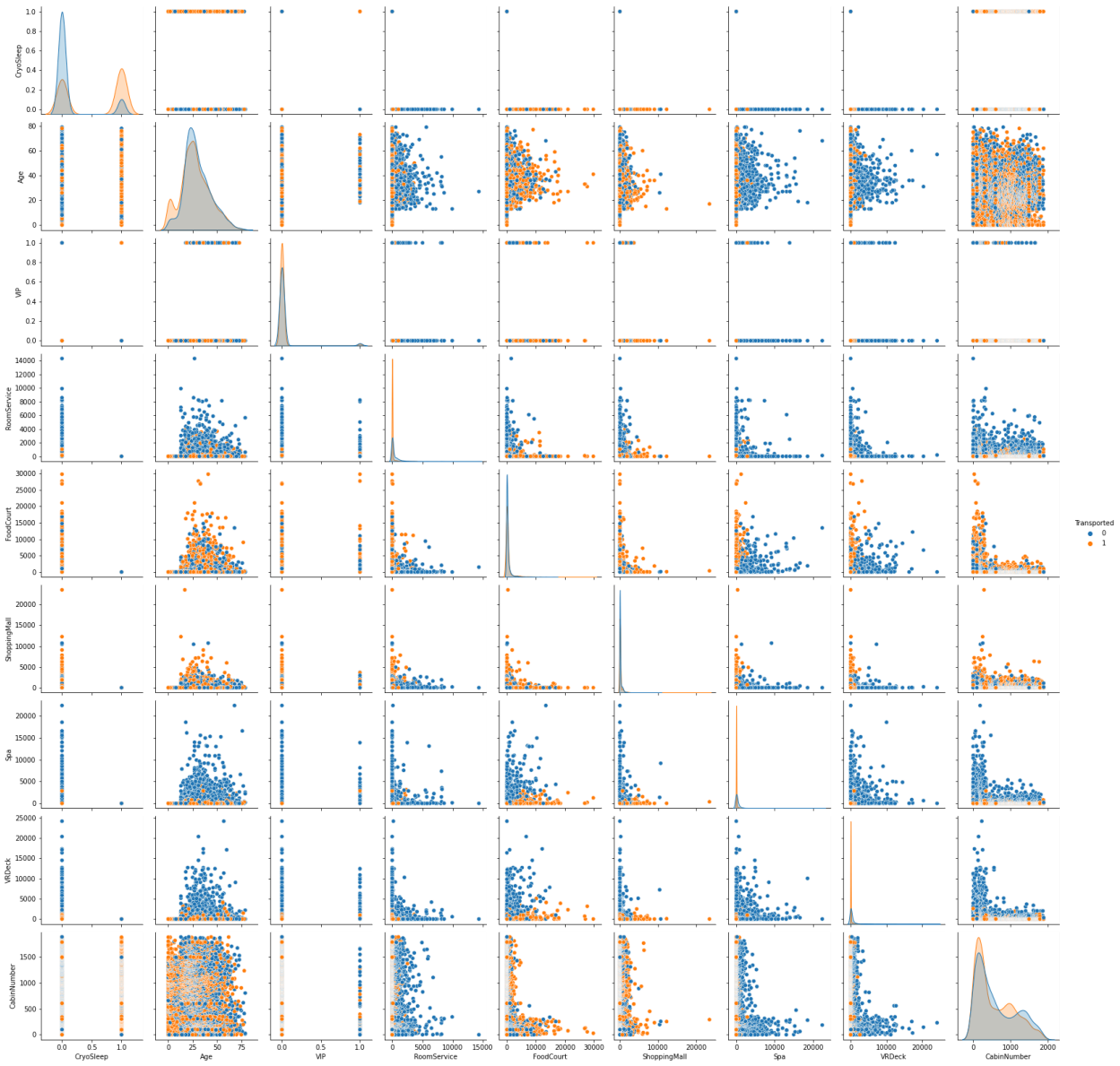
Supervised Machine Learning Course Final Project

The dataset is Kaggle's Spaceship Titanic dataset. The *Spaceship Titanic* is an interstellar passenger liner (analogous to the famous *Titanic* ship, which sank after colliding with an iceberg) which has struck a spacetime anomaly hidden in a dust cloud. The goal of the project is to predict which passengers were transported to an alternate dimension.

Data Exploration/Cleaning:

- Each passenger has a name, and is assigned a Passenger ID. These rows were dropped because they're not useful for prediction.
- Binary data (True/False values) include if the passengers were in Cryosleep, if they were a VIP, and if they were transported (what is being predicted). The binary data was changed from True/False values to integer values 0 and 1.
- The categorical data includes their home planet, what cabin they were in (cabin data was split to have different features for the deck they were on, cabin number, and side: port or starboard), and their destination. This data was one-hot encoded.
- Numerical data includes age, and how much money each passenger spent for room service, the food court, the shopping mall, the spa, and the VR deck. Age is normally distributed, and money spent on each service is highly skewed. For the purpose of two models, this data (along with Cabin Number) was scaled with min-max scaling.
- Missing values were replaced with the mean value (for Age or money spent), or most common value (for true/false data). Missing deck or side values were given their own value, which was unknown. Missing cabin numbers were given the value -1.

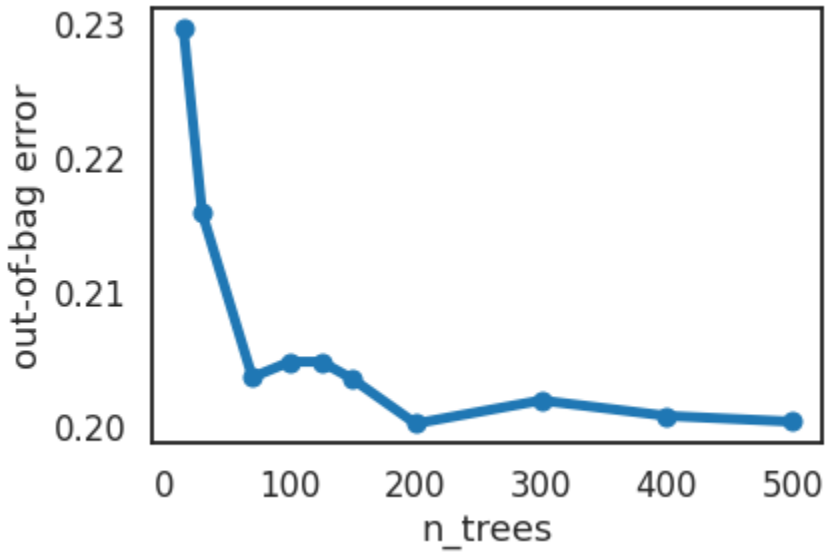
Feature Pairplots:



The labels are small in this image, but none of the features have a clear linear or polynomial relationship to each other. For this reason the three models I chose to train were a Random Forest, K Nearest Neighbors, and a Logistic model. Orange dots correspond to passengers that were transported, and blue dots correspond to passengers that were not transported.

Random Forest:

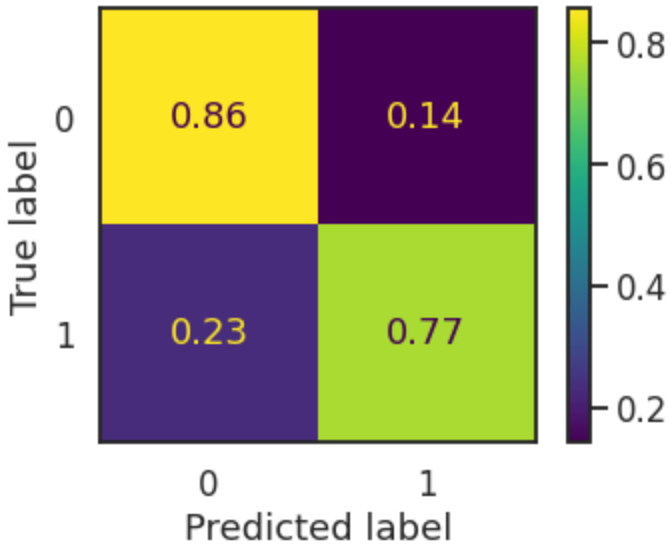
I started with this model, because I thought it would probably perform the best on this type of data. I split the data into 80/20 train/test datasets. A stratified split was used, although the data is split 50/50 between transported/not transported values anyway. The data for this model was not scaled. To see what the optimal number of trees to use were, models with different numbers of trees were fit and their out-of-bag error was compared.



I decided 200 trees would be the optimal number to use as, according to the above graph, that is when the error begins to level out.

	precision	recall	f1-score	support
0	0.78	0.86	0.82	863
1	0.85	0.77	0.80	876
accuracy			0.81	1739
macro avg	0.81	0.81	0.81	1739
weighted avg	0.81	0.81	0.81	1739

	accuracy	precision	recall	f1	auc
0	0.811961	0.845283	0.767123	0.804309	0.812299



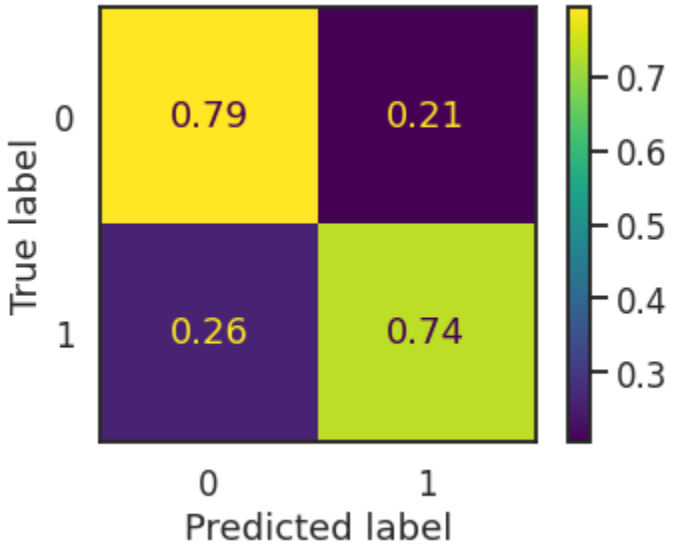
0 values here correspond to a passenger not being transported, and 1 corresponds to a passenger being transported. The model had more false positives for transported passengers, but seeing as values for precision/recall are >75%, I think the model does a decent job of predicting.

K Nearest Neighbors:

The data for this model (and the logistic model) was scaled as I described above. The dataset was also split into 80/20 train/test datasets. The same train/test data was used for this model and the logistic model. Different values for K were tested by training a model for each K value from 1 to 50 and comparing their F1 scores. I decided to use a value of 5 for K, as the F1 score began to drop off after that point.

	precision	recall	f1-score	support
0	0.75	0.79	0.77	863
1	0.78	0.74	0.76	876
accuracy			0.76	1739
macro avg	0.77	0.77	0.76	1739
weighted avg	0.77	0.76	0.76	1739

	accuracy	precision	recall	f1	auc
0	0.764807	0.784409	0.73516	0.758986	0.765031



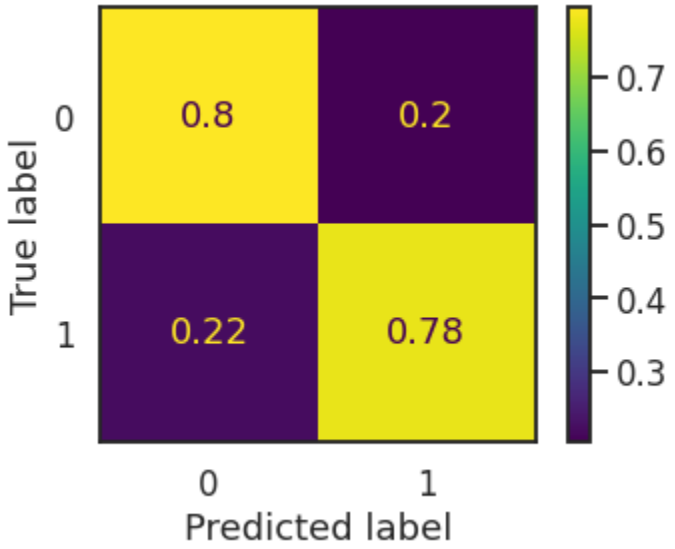
As explained above, 0 values here correspond to a passenger not being transported, and 1 corresponds to a passenger being transported. This model performed worse than the Random Forest model. It may benefit from some feature reduction.

Logistic Model:

Two models, one with an L1 penalty and one with an L2 penalty were tested. The model with the L1 penalty performed better.

	precision	recall	f1-score	support
0	0.78	0.80	0.79	863
1	0.79	0.78	0.79	876
accuracy			0.79	1739
macro avg	0.79	0.79	0.79	1739
weighted avg	0.79	0.79	0.79	1739

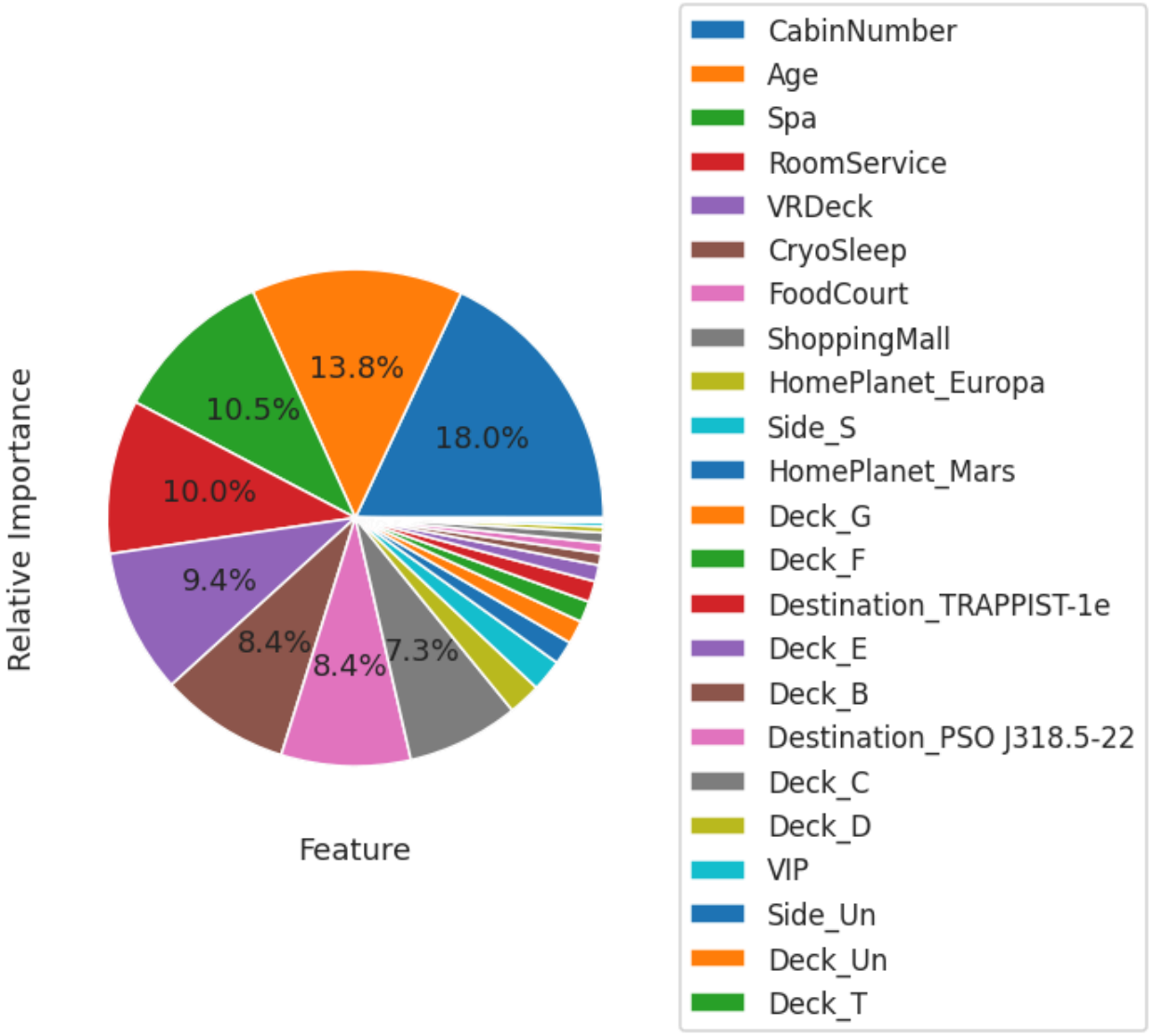
	accuracy	precision	recall	f1	auc
0	0.786084	0.794393	0.776256	0.785219	0.786158



As explained above, 0 values here correspond to a passenger not being transported, and 1 corresponds to a passenger being transported. This model performed better than the K Nearest Neighbors model, and worse than the Random Forest model, though not by much. It likely performed better than KNN because of the L1 penalty which will lessen the contribution of some of the features (i.e there is some feature selection). If I performed further feature selection before training all three models, probaby the KNN and Random Forest models would have performed better.

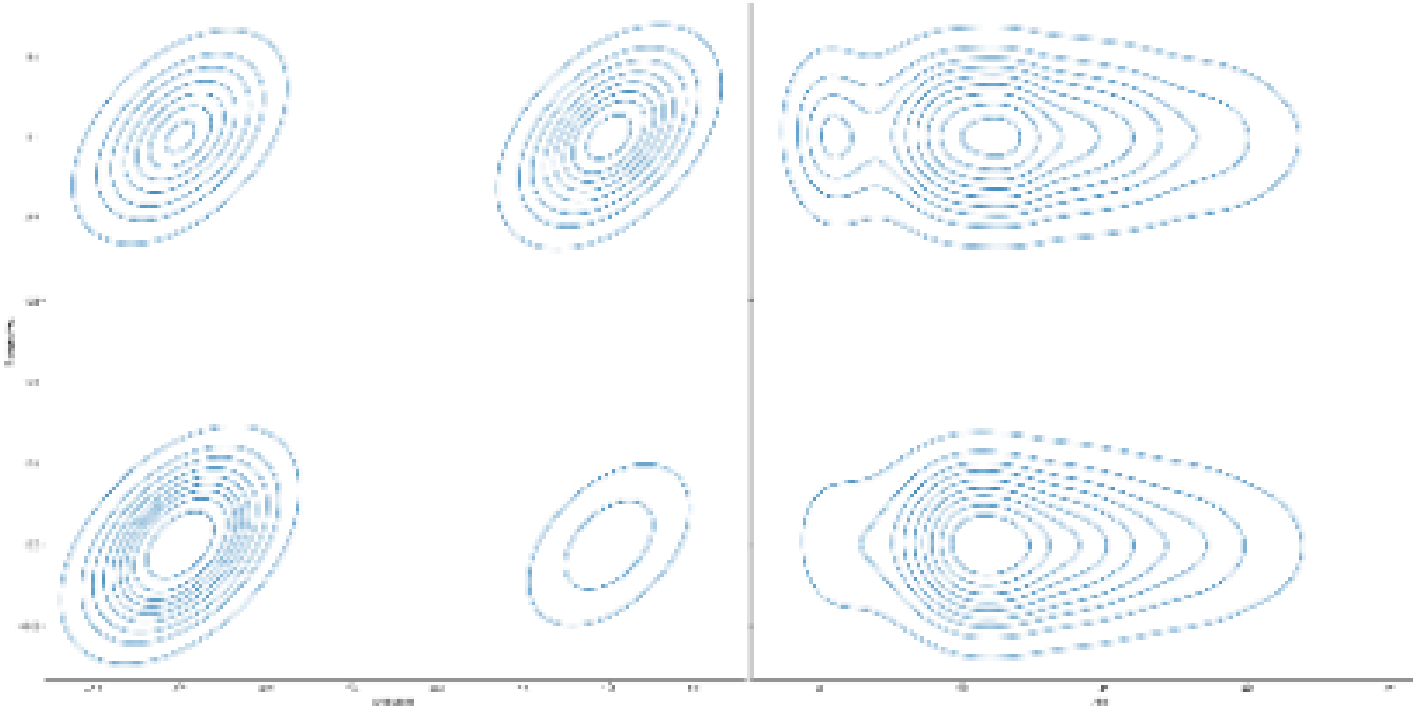
Discussion of Outcomes:

Based on the precision, recall, and F1 scores, the best performing model was the Random Forest model, followed by the logistic model, and then the K Nearest Neighbors model. The logistic model and random forest model both performed similarly, although the logistic model had less of a tradeoff between precision and recall. Likely if I had done better feature selection before training all three models, they would all be improved and there would likely be more of a performance gap between the Random Forest model and the logistic model.



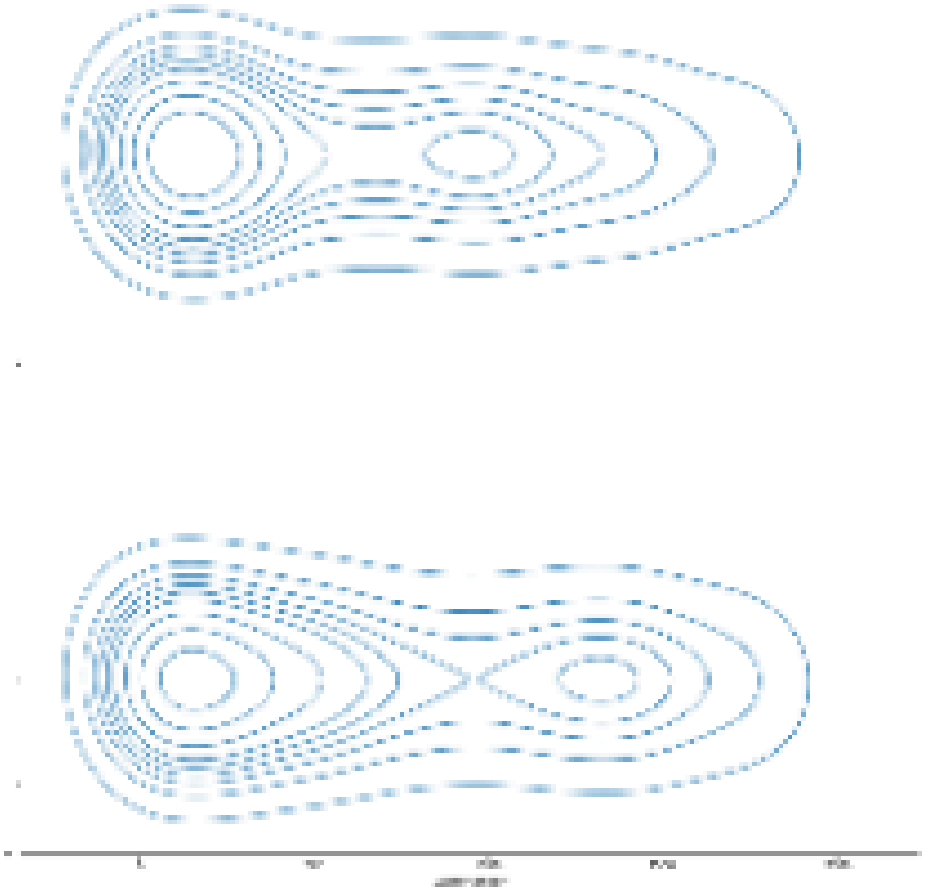
According to the feature importance from the Random Forest model, the most important features were cabin number, age, money spent, and if a passenger was in cryosleep.

Some of these features make sense, such as age and cryosleep. Looking at a pairplot between these two features and whether passengers were transported, it looks like younger passengers were transported more often, and people in cryosleep got transported more often. Money spent on each amenity also had a small correlation with who was transported or not. This might be a more powerful feature if I combined the money spent across all amenities into one feature.



Cryosleep is the left graph, age is on the right. The images refuse to resize correctly. The full image is transported_pairplot_large.png. Transported passengers have a value of 1 on the y axis (the top portion of the graphs). Un-transported passengers have a value of 0 on the y axis (the bottom portion of the graphs).

I question if the cabin number should be as important as the model thinks it is. Theoretically it could have something to do with a passenger's location on the ship, but since the deck and side of the ship weren't important features I think its importance is overblown. It may be due to bias, since more passengers tend to be concentrated in lower cabin numbers.



The distribution of cabin numbers for transported vs not transported passengers seems similar, though it seems like lower cabin numbers are slightly favored for transported passengers. It may be worth removing cabin information from the training data, and fitting the models again to see if there is an increase in performance.

Conclusions:

For predicting whether passengers were transported or not, the best performing model was the Random Forest model, and the most important

features were cabin number, age, amount of money spent on various amenities, and if passengers were in cryosleep.