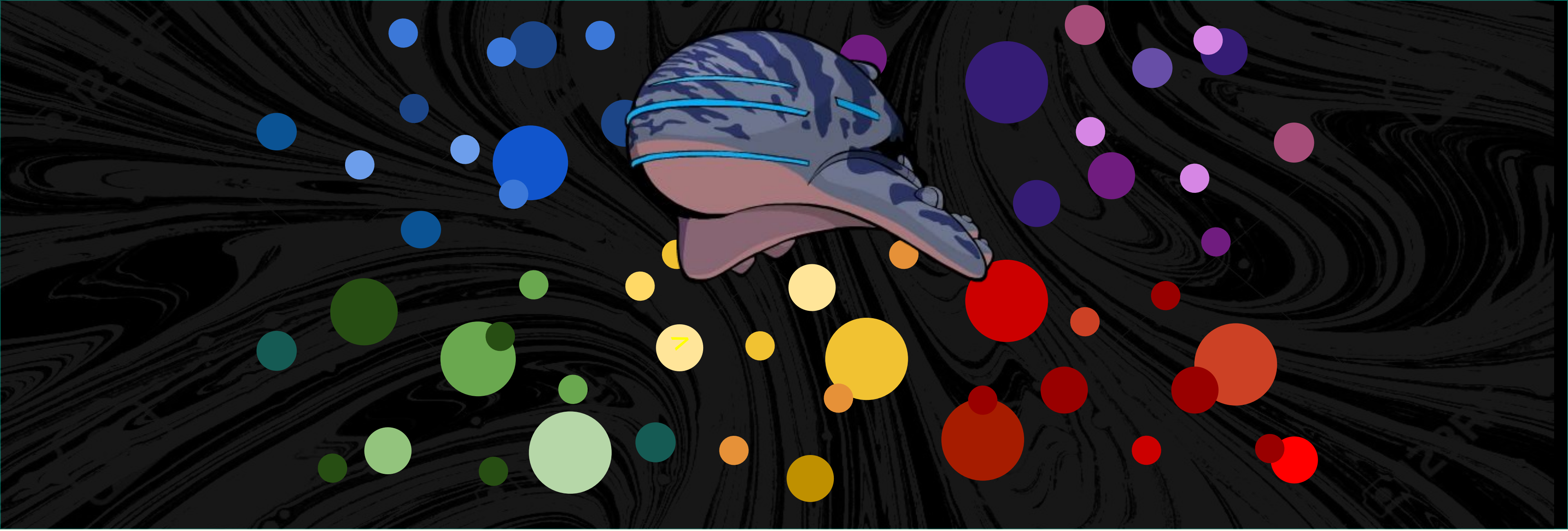


Spaceship Titanic Project

Predicting teleportation among Spaceship Titanic passengers.

Monica Amezquita; Alejandro Hernandez; Hugo Marquez

STA 4990 - Introduction to Data Science, Summer 2022

Welcome to the Year 2912...

- An interstellar spaceship- the *Spaceship Titanic*- was launched a month ago on a voyage to three habitable exoplanets.
- On board were **13,000 passengers** of diverse backgrounds and from three planets home planets.
- While in route to its first destination, the spaceship hit a spacetime anomaly hidden within a space dust cloud.
- Nearly **half the passengers onboard were transported** to an alternate dimension.

To help rescue and retrieve lost passengers, it is now up to three data scientists to put their skills to work and build a model that **accurately predicts which passengers were and were not transported**.

Exploratory Analysis

Given Set of Predictors

1. PassengerId
{gggg_pp}

2. HomePlanet
{Europa; Earth; Mars}

3. CryoSleep
{True; False}

4. Cabin
{Deck/RoomNum/Side}

5. Destination
{55 Cancri e; PSO J318.5-22; TRAPPIST-1e}

6. Age
{0-79}

7. VIP
{True; False}

8. RoomService, FoodCourt, ShoppingMall, Spa, VRDeck
{0 - 29,000}

9. Name
{'First Last'}

10. Transported (Response)
{True; False}

Initial Key Insights

Q. Which passengers had a greater chance of being teleported?

1. (**Expenditures**) Higher spenders
2. (**CryoSleep**) Passengers in cryosleep
3. (**VIP**) VIP passengers
4. (**Age**) High ratio of transported babies
5. (**HomePlanet**) Europa highest, Earth lowest

Values of other predictors seem to not differ so much by **Transported**.

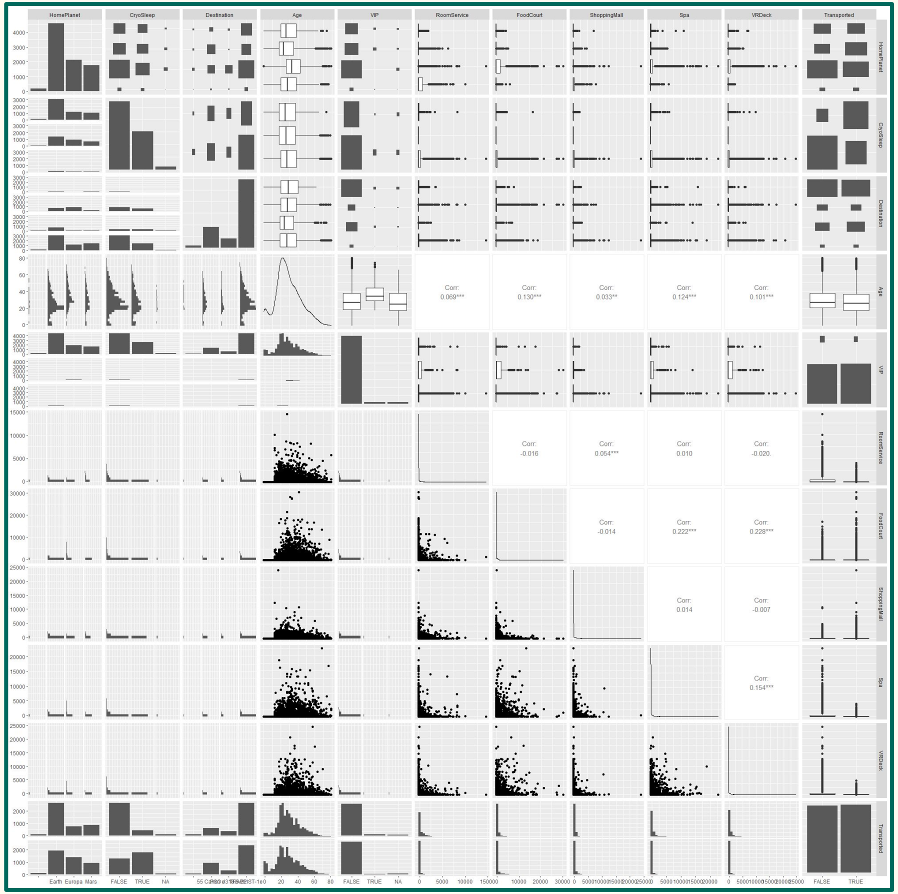
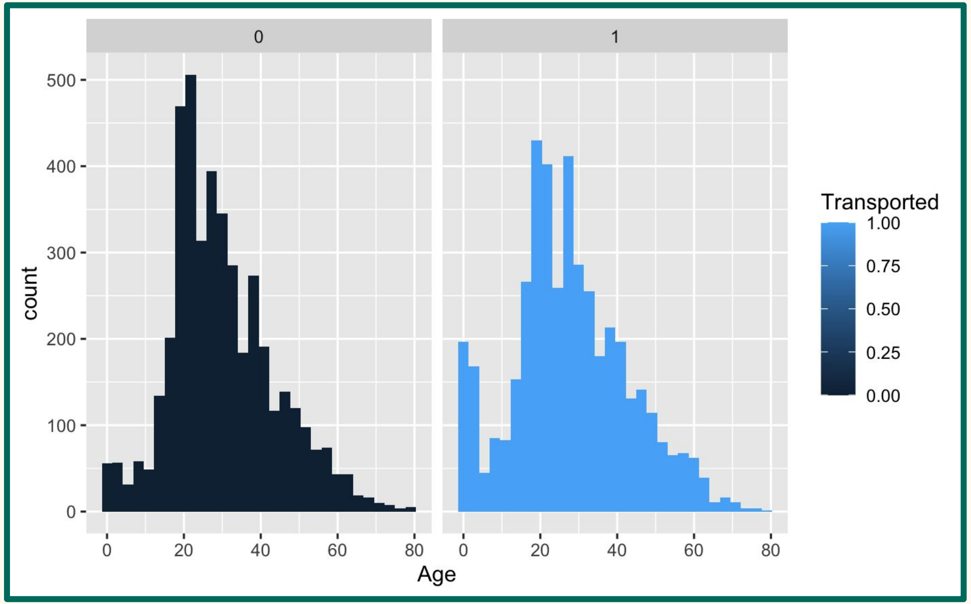


Fig 1: Scatterplot matrix from training dataset

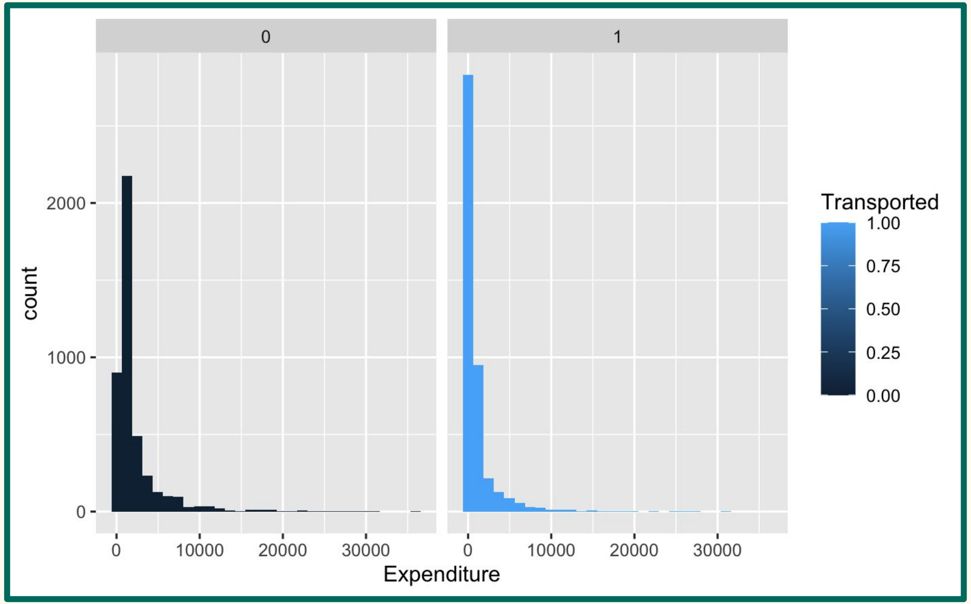
Histogram of **Age** colored by **Transported**

- **Very young children had high probability of transported to non-transported**
- Decrease in young adults that were transported
- Small decrease of transported individual for ages around 40 years old



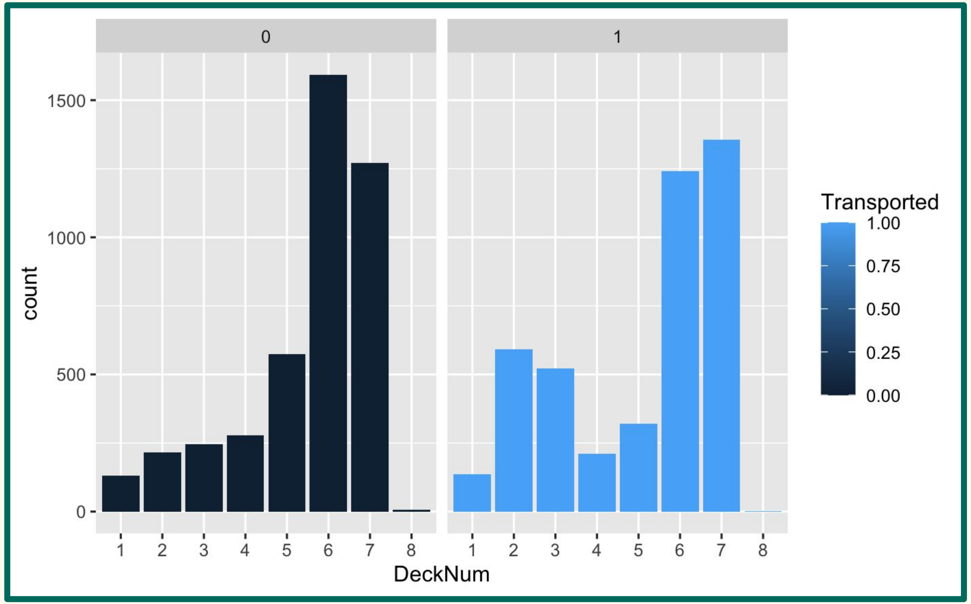
Histogram of **Expenditure** colored by **Transported**

- **Increase in transported individuals whose total expenditure was zero**
- Decrease transported individuals when some expenditure was present



Barchart of **Deck** colored by **Transported**

- **Highest percentage of transported passengers in decks 2, 3, and 7**
- Decrease of transportation amounts for individuals in decks 5 and 6



Project Overview

1. Data Pre-Processing
2. Model Building
3. Performance Evaluation

1. Data Pre-Processing

Imputation

Feature Engineering

Imputation

Imputation is the practice of **assigning a missing value** to an observation/record by inference or prediction.

- **Nonparametric**: KNN, Mean/mode, etc.
- **Parametric**: Linear Regression, Logistic Regression, LDA, etc.

Our 3 Imputation Methods:

1. Group-based

- Some **variables tend to have similar values within a group** (e.g. social status, spending habits, cryosleep preference, trip start/destination)
- To estimate missing values of non-solo passengers, **we can utilize group modes/means**.

Imputation

2. Planet-based

- If a passenger is travelling alone and missing a destination, we can predict it by using the **most common destination from their home planet**- and vice versa.

3. Multivariate parametric models

- Remaining missing values were imputed with models from the MICE library.
  - Lasso regression (**numeric**); Logistic regression + lasso (**binary**); LDA (**nominal**)

The most any variable was missing was 2% of its data.

So, although imputation certainly affects a model's potential, we don't expect this process to have great impact on later performance.

Example: How did we decide if groups had common values?

All groups

```
RoomService
Mean var: 482145.8
Mean stand dev: 276.1676

FoodCourt
Mean var: 3470367
Mean stand dev: 742.0944

ShoppingMall
Mean var: 457243.4
Mean stand dev: 229.4796

Spa
Mean var: 1528962
Mean stand dev: 492.121

VRDeck
Mean var: 1722467
Mean stand dev: 495.0647
```

Groups with NA member(s)

```
RoomService
Mean var: 329882.2
Mean stand dev: 234.8871

FoodCourt
Mean var: 2864001
Mean stand dev: 795.2028

ShoppingMall
Mean var: 61437.45
Mean stand dev: 114.8587

Spa
Mean var: 1612618
Mean stand dev: 602.0361

VRDeck
Mean var: 1724427
Mean stand dev: 613.5658
```

Pay attention to a variable's average standard deviation differs for all/NA groups.

See how average group deviation from the mean is lower for **ShoppingMall** and higher for **Spa**?

Feature Engineering

What is feature engineering?

The process of **creating new predictors** from raw data to increase the predictive power of the learning algorithm.

Engineered features should **capture additional information that is not easily apparent** in the original feature set.*

* Source: University of Connecticut - https://ailab.its.uconn.edu/feature-engineering/

Feature Engineering

- Convert "**Deck**" into ordinal categorical
- Convert "**Side**" into a binary categorical
    - "Port" = 0 vs "Starboard" = 1
- Convert "**HomePlanet**" to binary categorical
    - Europa = 0 vs. Earth/Mars = 1
- Create categorical variables for solo travelers
- Add **rnorm** amount to features related to spending
    - Prevent ties in KNN model
    - train$RoomService ← train$RoomService + rnorm(nrow(train),0,0.0000001)

```
#Create an ordinal variable for Deck such that
# A = 1
# B = 2
# C = 3
# D = 4
# E = 5
# F = 6
# G = 7
# T = 8

train$DeckNum <- as.factor(ifelse(train$Deck == 'A', 1,
                           ifelse(train$Deck == 'B', 2,
                           ifelse(train$Deck == 'C', 3,
                           ifelse(train$Deck == 'D', 4,
                           ifelse(train$Deck == 'E', 5,
                           ifelse(train$Deck == 'F', 6,
                           ifelse(train$Deck == 'G', 7,
                           ifelse(train$Deck == 'T', 8,
                                  'NA')))))))))
```

2. Model Building

Model Selection & Training

Data Split & Model Training

The dataset was provided in two CSV files, for training and testing.

- We further split the training set into a **training** **(75%)** and **validation** **(25%)** set.
    - Number of records: Train ~ 6,500; Validation ~2,200; Test ~4,300

From the training set, we trained **1 non-parametric** and **6 parametric** models of varying complexity.

- In the following slides, we'll provide a brief motivation for each model and any notes on its fitting process.

Models Used to predict

1. **KNN:**
   - Homeplanet, CyroSleep, DeckNum, Side, Destination, Age, Expenditure, peaple_in_group
2. **Least-squares, Ridge, Lasso Regression:**
   - Homeplanet, CyroSleep, DeckNum, Side, Destination, Age, Expenditure, peaple_in_group
3. **Logistic regression:**
   - Homeplanet, CyroSleep, DeckNum, Side, Destination, Age, Expenditure, peaple_in_group
4. **SVM with Linear, Polynomial, and RBF kernels:**
   - Homeplanet, CyroSleep, DeckNum, Side, Destination, Age, Expenditure, peaple_in_group
5. **Fully-Connected Neural Network:**
   - GroupId, Age, CryoSleep, Expenditure (Run time was too long therefore cutshort))

Linear Regression (Least Squares)

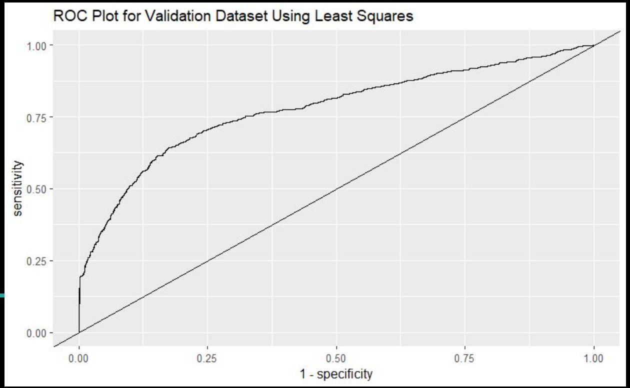
- Parametric predictors
- GLM method
- if any, motivation of predictors was the addition of a new predictor “DeckNum” where we are using it to predict if deck number/ level had an influence

Features

Plots and Charts

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.7279006
sens	binary	0.7004115
spec	binary	0.7628004
ppv	binary	0.7894249
npv	binary	0.6672761

	Truth	
Prediction	No	Yes
No	851	364
Yes	227	730



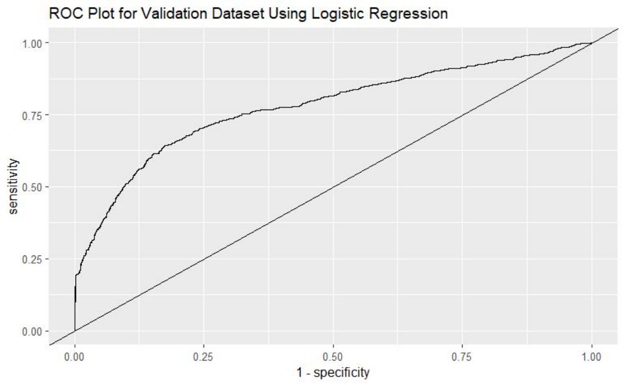
Logistic Regression

- For this model we will be using a "glm" method
- parametric

Tuning Parameters

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.7279006
sens	binary	0.7004115
spec	binary	0.7628004
ppv	binary	0.7894249
npv	binary	0.6672761

	Truth	
Prediction	No	Yes
No	851	364
Yes	227	730



K-Nearest Neighbors

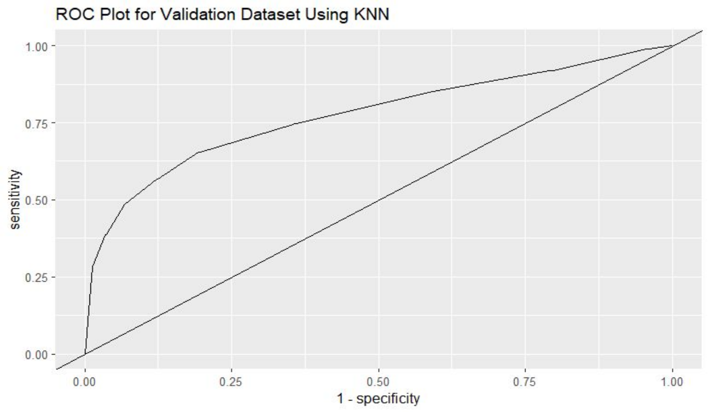
- Non-parametric predictors
- if any, motivation of predictors
- optimal tuning parameters
- Uses k-nearest neighbors in our data (similarities) to estimate the likelihood of those who were transported based on the information of the nearest data points it is evaluating

Plots and Charts

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.7283610
sens	binary	0.6958266
spec	binary	0.7721382
ppv	binary	0.8042672
npv	binary	0.6535649

5 rows

	Truth	
Prediction	No	Yes
No	867	379
Yes	211	715

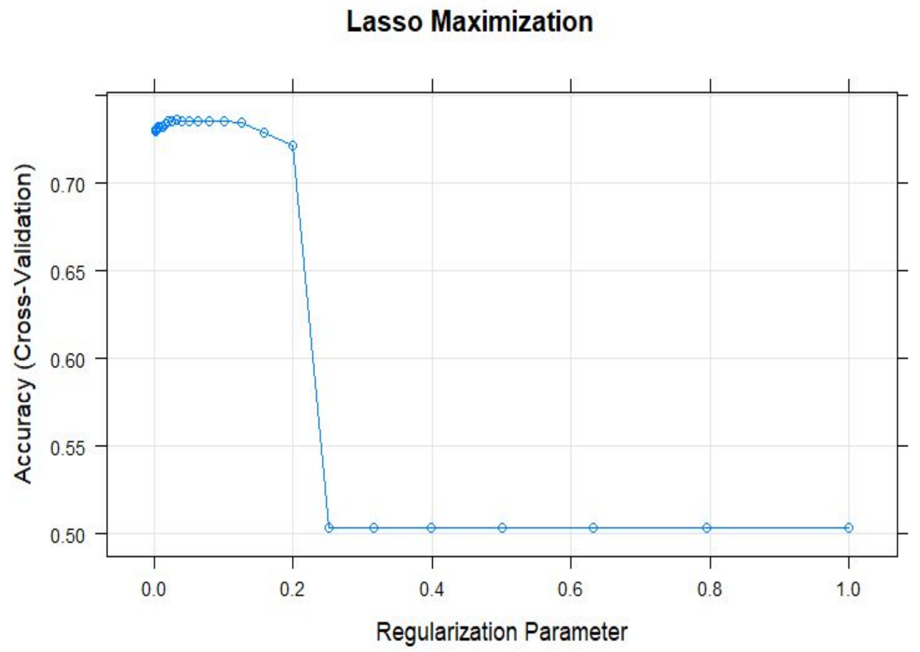


Lasso Classification

- Using the “glmnet” method to generalize the model
- This model was selected because of it’s interpretability since we are working with a few predictors
- Since we are using LASSO alpha =1
- Using CV 5 fold
- Using this model will cause some coefficients to shrink to zero only using the intercept
- We chose this model because we believe we have collinearity among or chosen predictors when predicting if someone was transported

Tuning Parameters

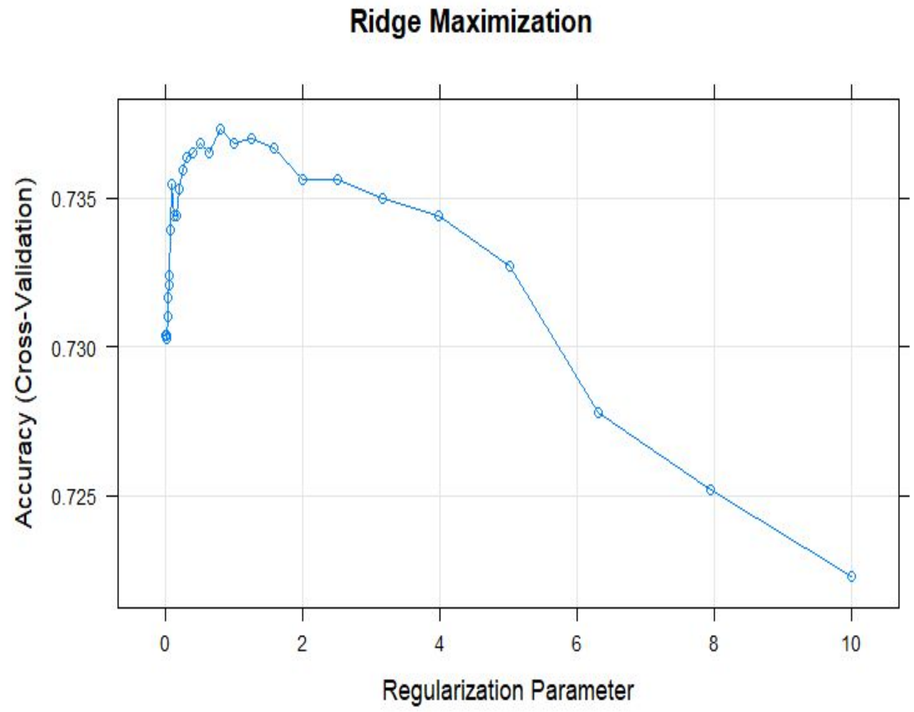
- LASSO maximization plot using a tuning parameter grid



Ridge Classification

- Using the "glmnet" method to generalize the model
- Added a penalizing tuning parameter for lambda to find its minimum because it was not hitting it
- We will have non-zero coefficients
- Alpha = 0 for Ridge
- Using CV 5 fold

Tuning Parameter



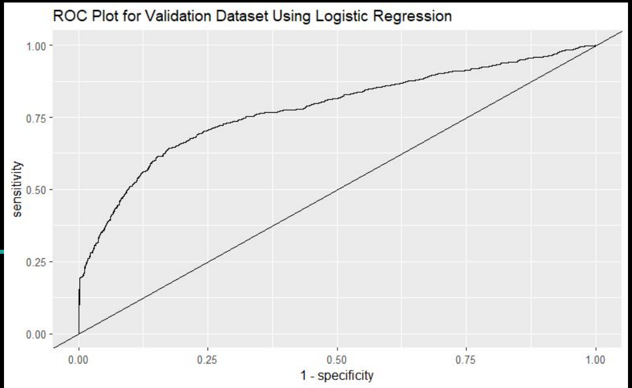
Linear SVM

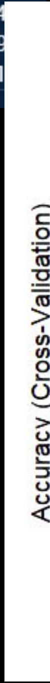
- This method was implemented to make use of the decision boundary it produces in our classification model
- Using this method created a hyperplane since we are using multiple predictors in R^n where n is a natural number
- Using C the cost to optimize and scale our tuning parameter
- Using the probabilistic outcomes based upon our cutoff to produce our boundary

Plots and Charts

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.7279006
sens	binary	0.7004115
spec	binary	0.7628004
ppv	binary	0.7894249
npv	binary	0.6672761

	Truth	
Prediction	No	Yes
No	851	364
Yes	227	730





Polynomial SVM

- Using the SVM polynomial to better fit the decision boundary
- We will fit using degrees 1, 2, 3
- We must be careful of higher degrees because it can lead us to over estimating
- Our degrees 2 and 3 will allows our boundaries to be more flexible while the first degree will be a linear separation.

Tuning Parameter

-

RBF SVM

- Implementing this algorithm method to learn and find our non linear classifier
- We must keep in mind the distance we establish for our support vectors because widening them can lead to over supporting our boundary

Tuning Parameters

-

Fully Connected Neural Network

- Using 15 nodes (neurons)
- Parameters are unknown and must be estimated from our data
- Predictors used were GroupID, CryoSleep, Age, Expenditures

Predictors Used:

- Predictors used were GroupID, CryoSleep, Age, Expenditures

3. Performance Evaluation

Sensitivity/Specificity Metrics

ROC Curves

Selected Metrics

1. **ROC Curves** and **AUC**

Common metrics of classification accuracy for positive and negative class at all possible thresholds.

2. **Confusion Matrix**

Table of correct/incorrect positive/negative classifications.

3. **Metrics of Sensitivity and Specificity**

Model Performance

Comparing model performance on the Validation set:

- **KNN**:

accuracy	binary	0.7255985

	Truth	
Prediction	No	Yes
No	860	378
Yes	218	716

- **Least Squares**:

accuracy	binary	0.7279006

	Truth	
Prediction	No	Yes
No	851	364
Yes	227	730

Model Performance

Lasso

```
10 x 1 sparse Matrix of class "dgCMatrix"
                                 s1
(Intercept)             -0.07566701
HomePlanet              -0.34892593
CryoSleepTRUE            0.99299837
DeckNum                 -0.01132919
Side                     0.06005447
DestinationPSO J318.5-22 .
DestinationTRAPPIST-1e   .
Age                      .
Expenditure             -0.03925901
people_in_group          .
```

accuracy	binary	0.73562

Ridge

```
10 x 1 sparse Matrix of class "dgCMatrix"
                                  s1
(Intercept)               0.238479349
HomePlanet               -0.235670437
CryoSleepTRUE             0.570758739
DeckNum                  -0.042069199
Side                      0.151696298
DestinationPSO J318.5-22 -0.054520119
DestinationTRAPPIST-1e   -0.100827893
Age                      -0.003052072
Expenditure              -0.024559215
people_in_group           0.017332049
```

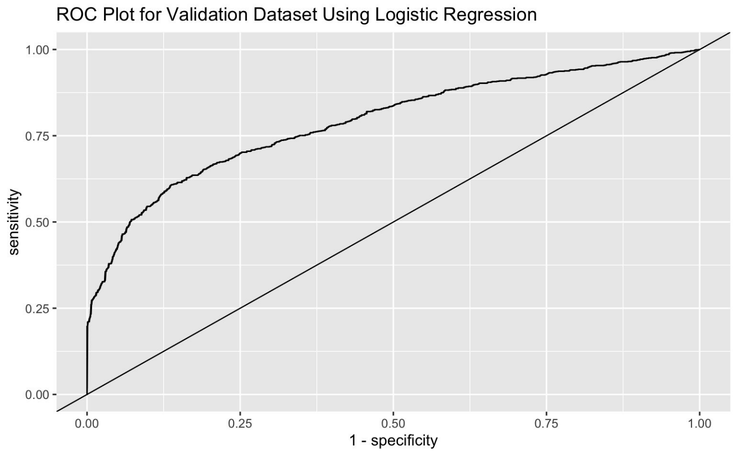
accuracy	binary	0.7376169

Model Performance

Logistic Regression

accuracy	binary	0.7279006

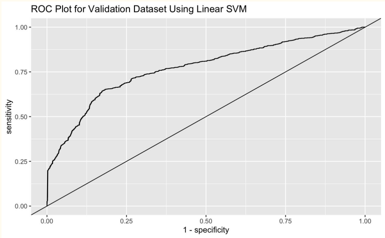
	Truth	
Prediction	No	Yes
No	851	364
Yes	227	730



Model Performance

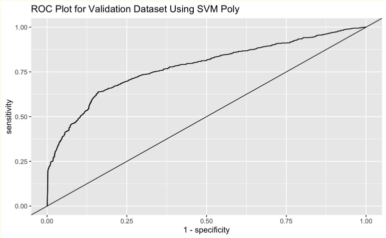
SVM Linear

accuracy	binary	0.7219153



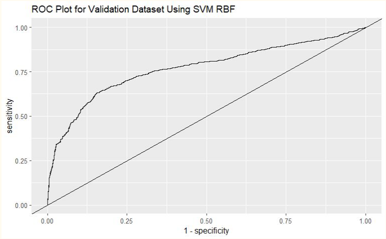
SVM Poly

accuracy	binary	0.7329650



SVM RBF

accuracy	binary	0.7320442



Model Performance

Fully Connected Neural Network:

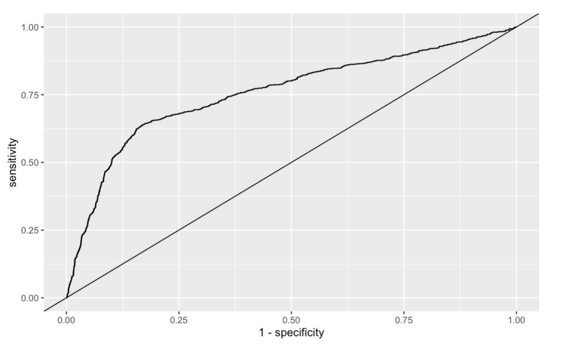
Metrics table:

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.7329650
sens	binary	0.6936236
spec	binary	0.7900677
ppv	binary	0.8274583
npv	binary	0.6398537

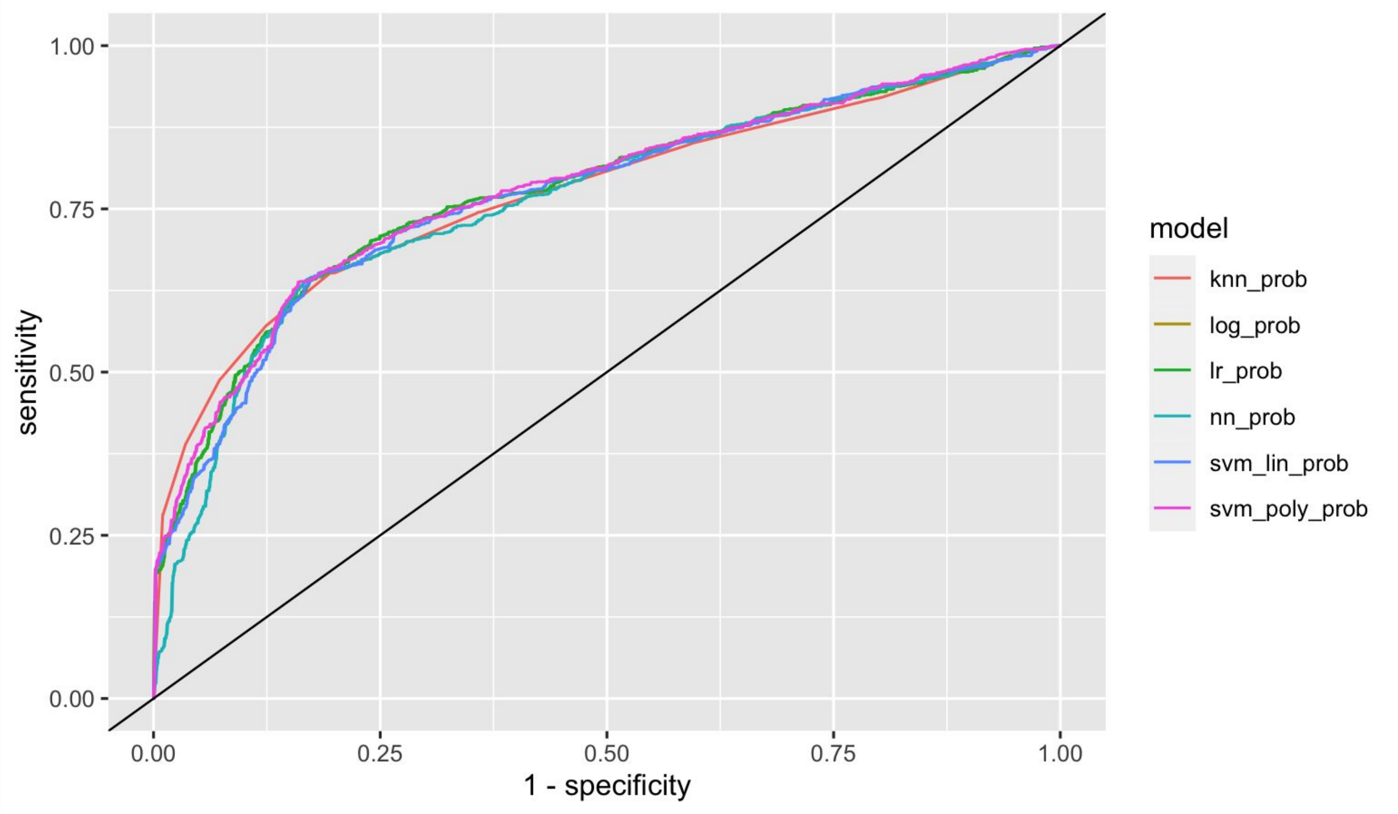
Confusion Matrix:

```
          Truth
Prediction  No Yes
       No  892 394
       Yes 186 700
```

ROC Curve:



All models ROC Overlaid (Validation Set)



Final Model & Recommendations

Recommended Model

Selecting a final model depends on our ultimate objective. Which do we value more, correct positive predictions, correct negative predictions, or overall accuracy?

For example, logistic regression maximized specificity (0.77) and ridge maximized sensitivity (0.82).

In our case, we selected our final model based on its AUC, because we believe it is a fair metric of predictive strength for both the positive and negative class at all possible thresholds.

Features used:

HomePlanet + CryoSleep + DeckNum + Side + Destination + Age + Expenditure + people_in_group

Kaggle Submission Results

1. First submission score: 0.00000

   (Incorrect **Transported** variables)

2. Final Ridge Score: 0.73766

Thank You!

Monica Amezquita; Alejandro Hernandez; Hugo Marquez

STA 4990 - Introduction to Data Science, Summer 2022