

# ELECTRIC DREAM MACHINE

Our adventures in *SSSSPPAAAAACCCEE*

# OUR COMPETITION

Frame Story:

SPACESHIP TITANIC!

An evil wormhole  
blackhole asteroid  
iceberg sunk the space  
ocean liner cruise ship,  
and half the  
passengers teleported!

Question:

WhY DiD SoMe  
PaSsEngERs TeLePort  
ANd OthERs DidN'T!?

Problem:

Make a **binary  
classification** model  
from passenger  
information that can  
predict whether or not  
a passenger will be  
teleported

# ABOUT THE DATA

Training size: 800kb

Test size: 400kb

Num instances training

(m): 8694

Num instances testing

(m): 4277

Teleported/Not teleported:

- Roughly 50/50

Params: n=8?

- ★ Passenger Name (string)
- ★ Home Planet (enum)
- ★ Destination Planet (enum)
- ★ Cryosleep (bool)
- ★ Age (int)
- ★ VIP (bool)
- ★ Cabin (string/int/bool)
- ★ Money Spent (enum/double)



# INITIAL RESULTS FROM LOGISTIC REGRESSION

78.4% ACCURACY

BEST PARAMETERS :

'CLASSIFIER\_\_C': 100

'CLASSIFIER\_\_MAX\_ITER': 300

'CLASSIFIER\_\_SOLVER': 'LIBLINEAR'

# DECISION TREE

77.4% ACCURACY

BEST PARAMETERS :

'classifier\_\_criterion': 'gini'

'classifier\_\_max\_depth': 10

'classifier\_\_max\_features': None

'classifier\_\_min\_samples\_leaf': 1

'classifier\_\_min\_samples\_split': 20

# RANDOM FOREST

79.2% ACCURACY

BEST PARAMETERS :

'classifier\_\_n\_estimators': 300

'classifier\_\_min\_samples\_split': 2

'classifier\_\_min\_samples\_leaf': 1

'classifier\_\_max\_features': 'sqrt'

'classifier\_\_max\_depth': 10

# Naive Bayes

★ 76.4% ACCURACY (GAUSSIAN)

BEST PARAMETERS :

'CLASSIFIER\_\_VAR\_SMOOTHING': 0.036946

61.9% ACCURACY (MULTINOMIAL)

BEST PARAMETERS :

'classifier\_\_alpha': 0.001

'classifier\_\_fit\_prior': True

# MULTILAYER PERCEPTRON

80.1% ACCURACY

BEST PARAMETERS :

'hidden\_layer\_sizes': (100,)

'max\_iter': 500

'alpha': 0.0001





# MODEL TUNING

Neural-Network :

- Takes a long time to run.

- Maximum iterations - sometimes maximum iterations would be reached before the data converged, but more iterations would lead to possible overfitting.

# MODEL TUNING

The others :

- Take less time to run

- Used the GridSearchCV from sklearn to try all possible combinations of a set of parameters

- Once all possible combinations ran, we printed out the top 5 and their accuracies

- Substituted guess work of choosing parameters with brute computation

```
self.param_grid_dt = {  
    'classifier__max_depth': [None, 10, 20, 30, 40],  
    'classifier__min_samples_split': [2, 5, 10, 20],  
    'classifier__min_samples_leaf': [1, 2, 4, 6],  
    'classifier__max_features': [None, 'sqrt', 'log2'],  
    'classifier__criterion': ['gini', 'entropy']  
}
```

# TOP 3 CONFIGURATIONS

## MULTI-LAYER PERCEPTRON

- 80.13% AVERAGE CV ACCURACY
- 81.14% FINAL DATASET ACCURACY

## RANDOM FOREST

- 79.17% AVERAGE CV ACCURACY
- 80.28% FINAL DATASET ACCURACY

## LOGISTIC REGRESSION

- 78.43% AVERAGE CV ACCURACY
- 79.29% FINAL DATASET ACCURACY

# TOP PERFORMER

## MULTI-LAYER PERCEPTRON

- 80.13% AVERAGE CV ACCURACY
- 81.14% FINAL DATASET ACCURACY

TAKES MORE FACTORS INTO ACCOUNT THAN MERELY PRESENCE AND ABSENCE, LIKE NAIVE BAYES.

COMBS BACK AND FORTH, REDISTRIBUTING WEIGHTS AND IMPROVING ITSELF.

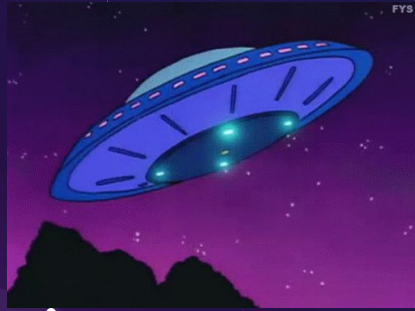
HOWEVER, WE HAD ONE LAST TRICK UP OUR SLEEVES...



# SUPER MEGA ULTIMATE ENSEMBLE

AN ENSEMBLE OF OUR TOP 3 CONFIGURATIONS

TAKES PREDICTIONS FROM ALL 3, RETURNS THE MAJORITY  
VOTE



HOW DID WE DO?



★ PLACEMENT: #335/2572

★ ACCURACY: 80.71%

# TIPS AND TRICKS

DIVIDING OUR "CABIN" ATTRIBUTE INTO 3 SEPARATE  
ATTRIBUTES

- MORE USEABLE DATA

PASSENGERID

- IDENTIFYING GROUP NUMBER

