SU2023 CS 667-VT / 767-VT Machine Learning

Final Project

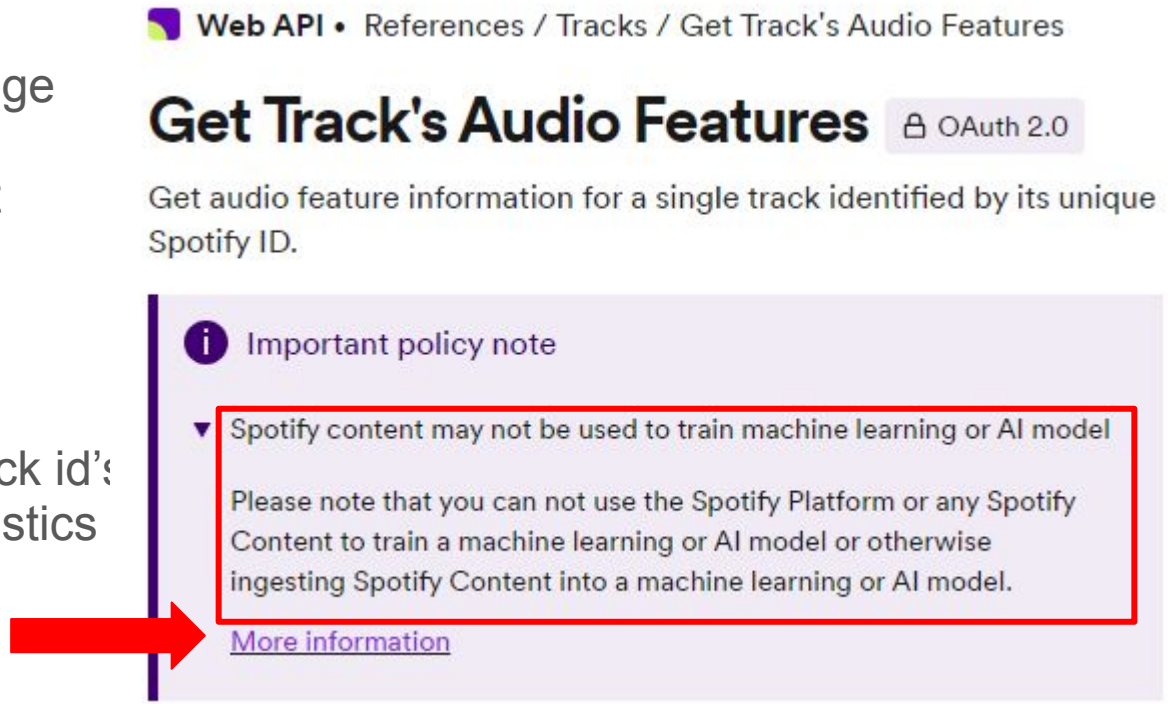
Group# 4

Trenton Tidwell, David Rubey, Yoga Murugan

Titanic Survival Prediction

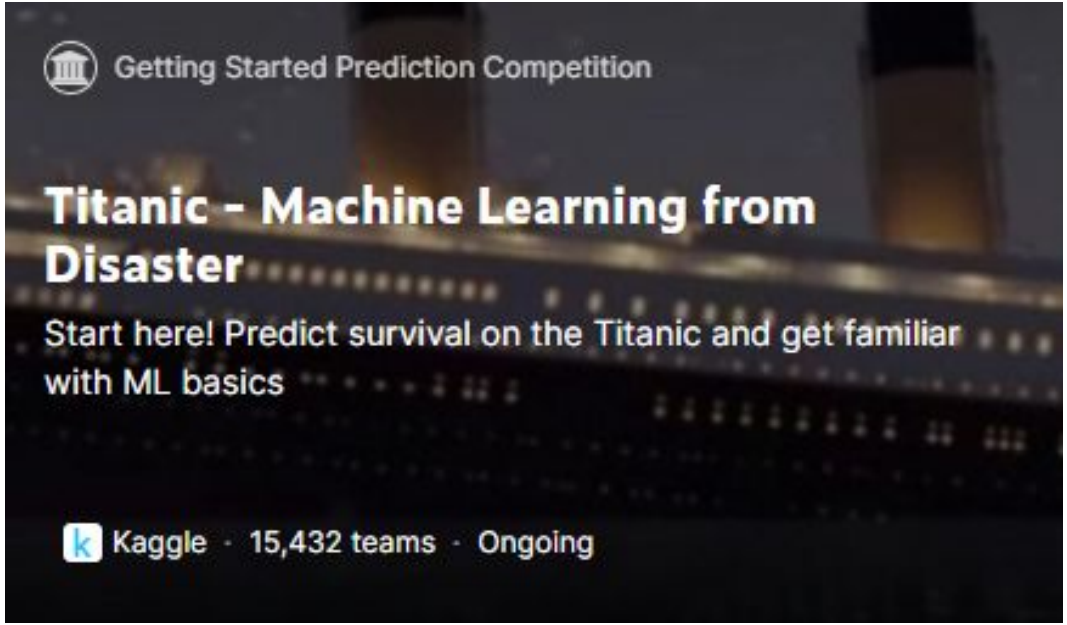
Spotify Music Playlist Recommendation (Initial Project Idea)

- ACM RecSys Challenge 2018
- Spotify Million Playlist Dataset Challenge
- Predict genre of song
- Predict new song for existing playlists
- Use spotify Api on track id's to get song characteristics
- Oops, time to pivot



Titanic Survival Prediction (Backup Project)

- Kaggle Machine Learning Competition
- Titanic RMS sank in 1912
  - 1317 passengers
  - 492 survived
- Predict the survival of passengers based on characteristics such as:
  - Sex
  - Age
  - Ticket Price
  - Cabin Number



Titanic Dataset

https://www.kaggle.com/competitions/titanic

- train.csv - 891 rows
- test.csv - 418 rows

Competition data has no ground truth values in Test.csv

Complete Dataset

https://www.kaggle.com/datasets/vinicius150987/titanic3

- titanic3.xls - 1309 rows

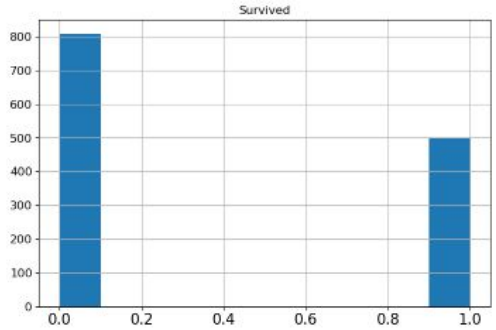
Complete data is same as above combined with all ground truth values

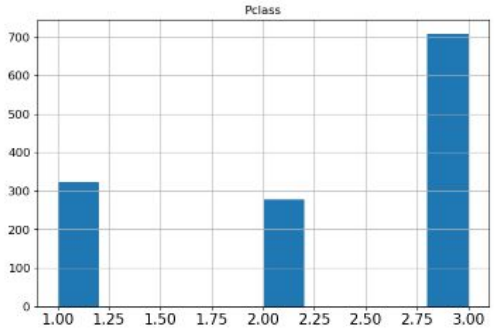
Variable	Definition	Key
survival	Survival (ground truth)	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

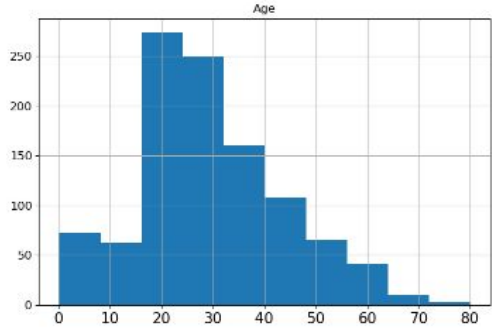
Titanic Dataset

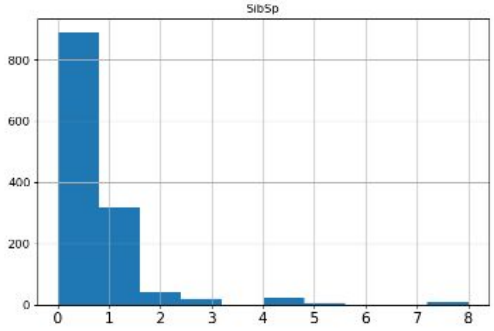
Histograms
(raw numeric data)

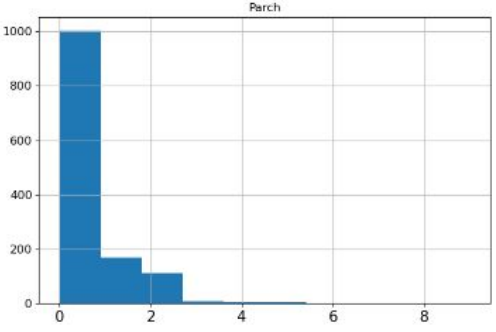
- Survived
- Pclass
- Age
- Sibsp
- Parch
- Fare

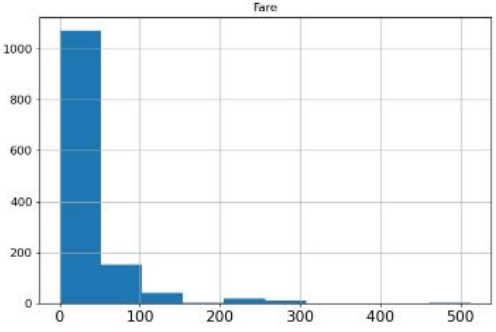












Data Preprocessing

Feature Selection

We threw out ticket number and passenger name, and used a subset of the characteristics to predict survival of an individual passenger:

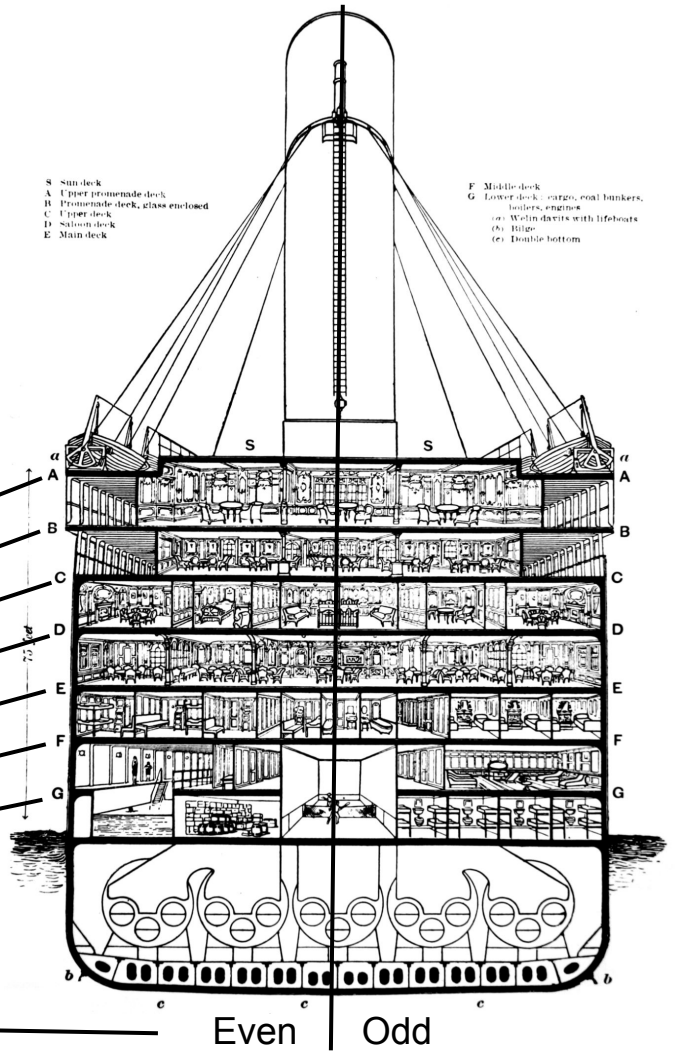
- **Pclass**: Ticket class (1st, 2nd, 3rd)
- **Sex**: Male or Female
- **Age**: Age in years (0.92 data point represents an 11 month old child)
- **sibsp**: Number of siblings aboard the ship
- **parch**: #of of parents / children aboard the ship
- **fare**: ticket price
- **cabin**: Cabin number
- **embarked**: Port of embarkation (C=Cherbourg, Q=Queenstown, S=Southampton)

Data Preprocessing

One-Hot Encoding

We transformed our categorical dimensions into numeric data so that it could be operated on by multiple machine learning algorithms.

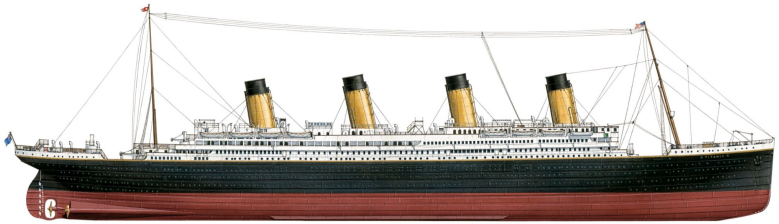
- **embarked**:
  - Embarked_C
  - Embarked_Q
  - Embarked_S
- **Pclass:**
  - Pclass_1,
  - Pclass_2,
  - Pclass_3
- **cabin**:
  - deck_A
  - deck_B
  - deck_C
  - deck_D
  - deck_E
  - deck_F
  - deck_G
- **cabin**:
  - deck_even



Data Preprocessing

Other Preprocessing

- **cabin**:
  - deck_num
    - we parsed the cabin number as a numeric ordinal value
    - ranges from 2 to 121 and represents the front to back dimension
    - loosely represent distance of the cabin from the midpoint



121 2

- **Sex**:
  - we transformed this dimension from textual values (female / male) into a binary representation of (0 / 1)

Data Preprocessing

Missing Data (NaNs)

We encoded some of our missing data using interpolation techniques to improve the accuracy of some models

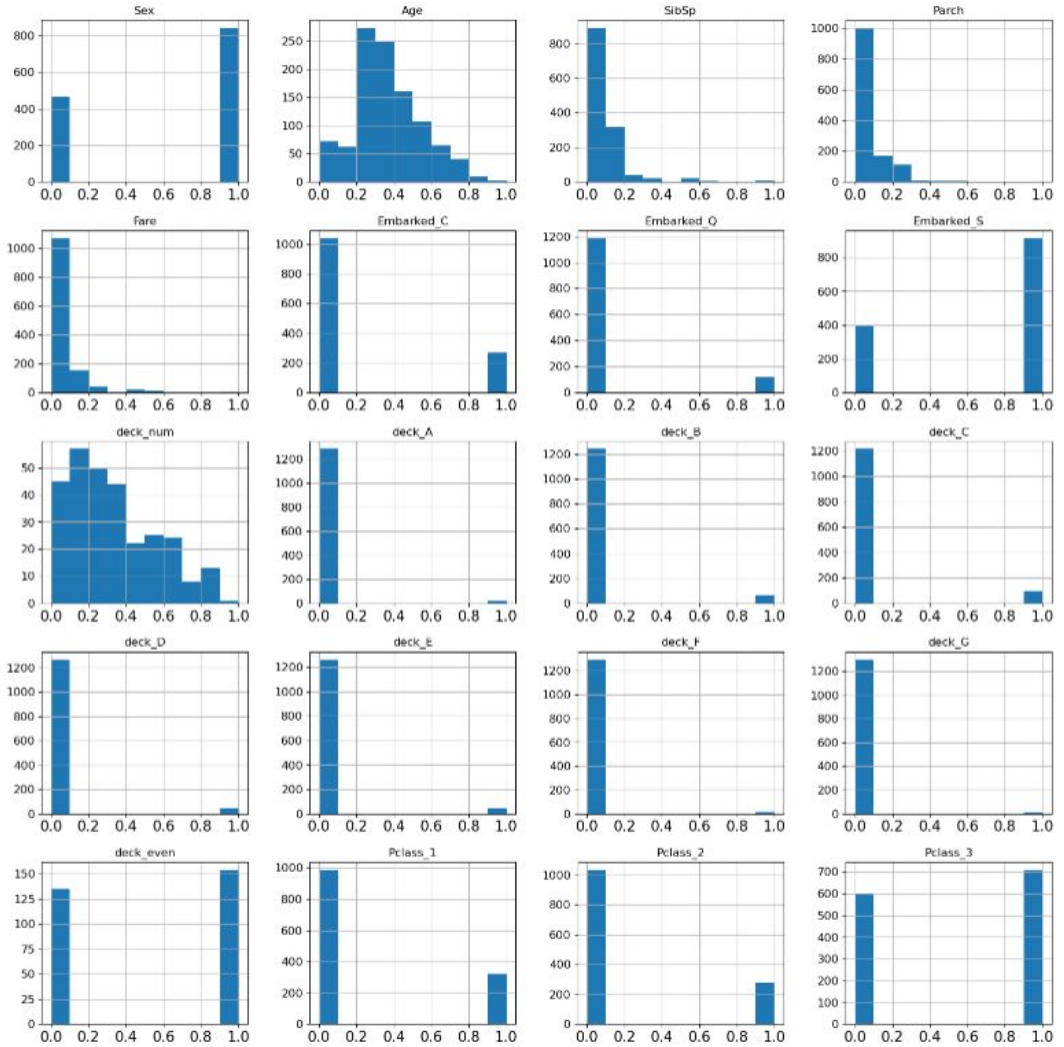
- **deck_even**: We turned NaNs in the cabin even/odd feature into 0.5 since it is between 0 and 1 and reflects our knowledge of the value. This however negates the feature's status as a categorical variable.
- **deck_num**: For, Cabin number (deck_num) we represented missing values with the midpoint of the range of values since that is a reflection of the likely position of the person on the boat. This is under the assumption that all decks have the same number of rooms of equidistant spacing (this is untrue). The range of this data is 2 to 121, so we replaced missing values with the midpoint of these values: 60.
- **age**: NaNs were replaced by the average age which is 29.88 for the full dataset
- **fare**: NaNs were replaced by the average Fare which is 33.29 for the full dataset
- **embarked**: NaNs were inherently ingested into the Embarked encoding as [0, 0, 0]

For comparative purposes, we will train models on both the full dataset with NaNs replaced as well as a reduced NaN-dropped dataset that has all rows that have NaN values in them removed.

Data Preprocessing

Histograms (normalized)

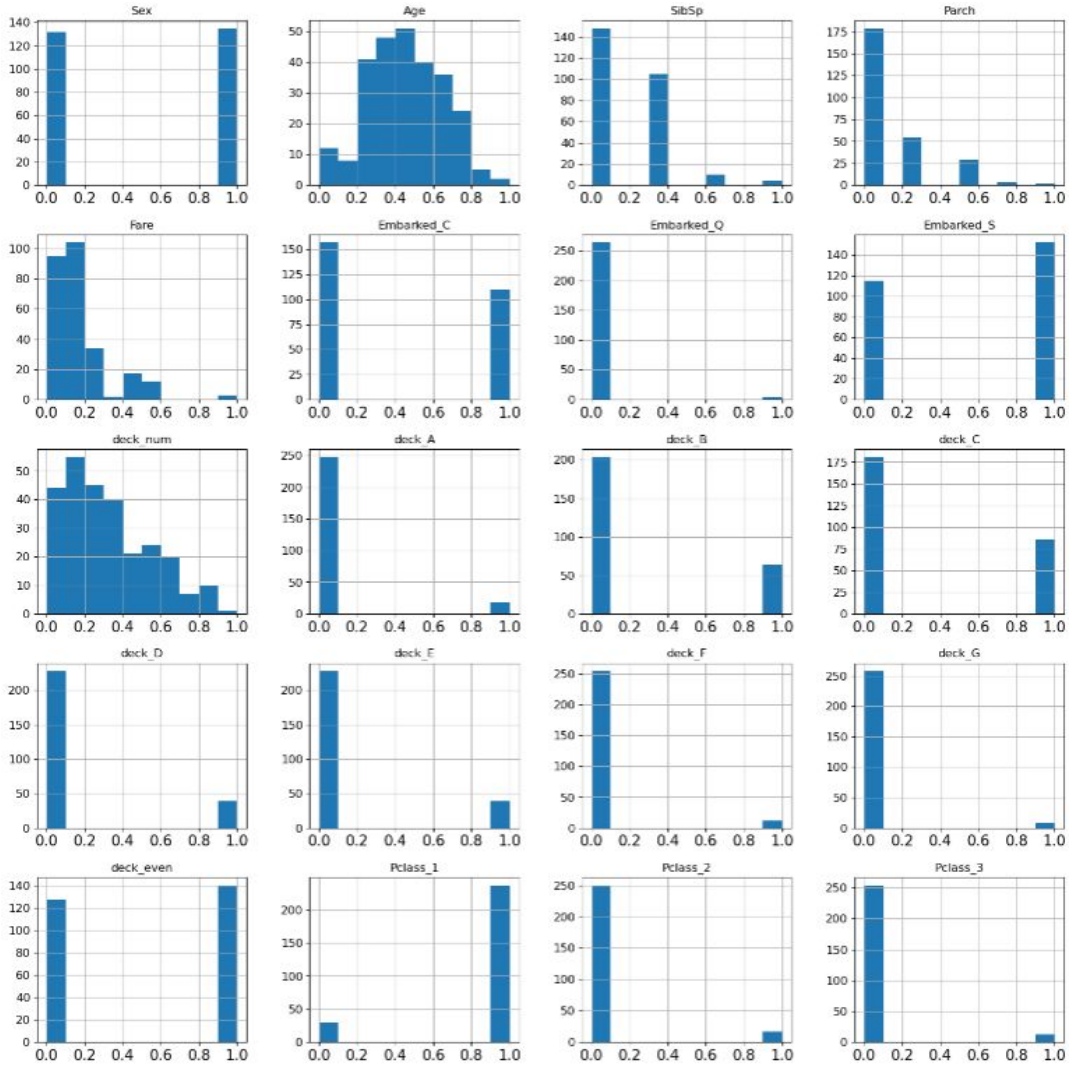
- Sex
- Age
- Sibsp
- Parch
- Fare
- Embarked_C
- Embarked_Q
- Embarked_S
- deck_num
- deck_A
- deck_B
- deck_C
- deck_D
- deck_E
- deck_F
- deck_G
- deck_even
- Pclass_1
- Pclass_2
- Pclass_3



Data Preprocessing

Histograms (normalized NaN-dropped)

Notice the size of the unedited NaN-dropped dataset vs the original (267 vs 1309 data points)

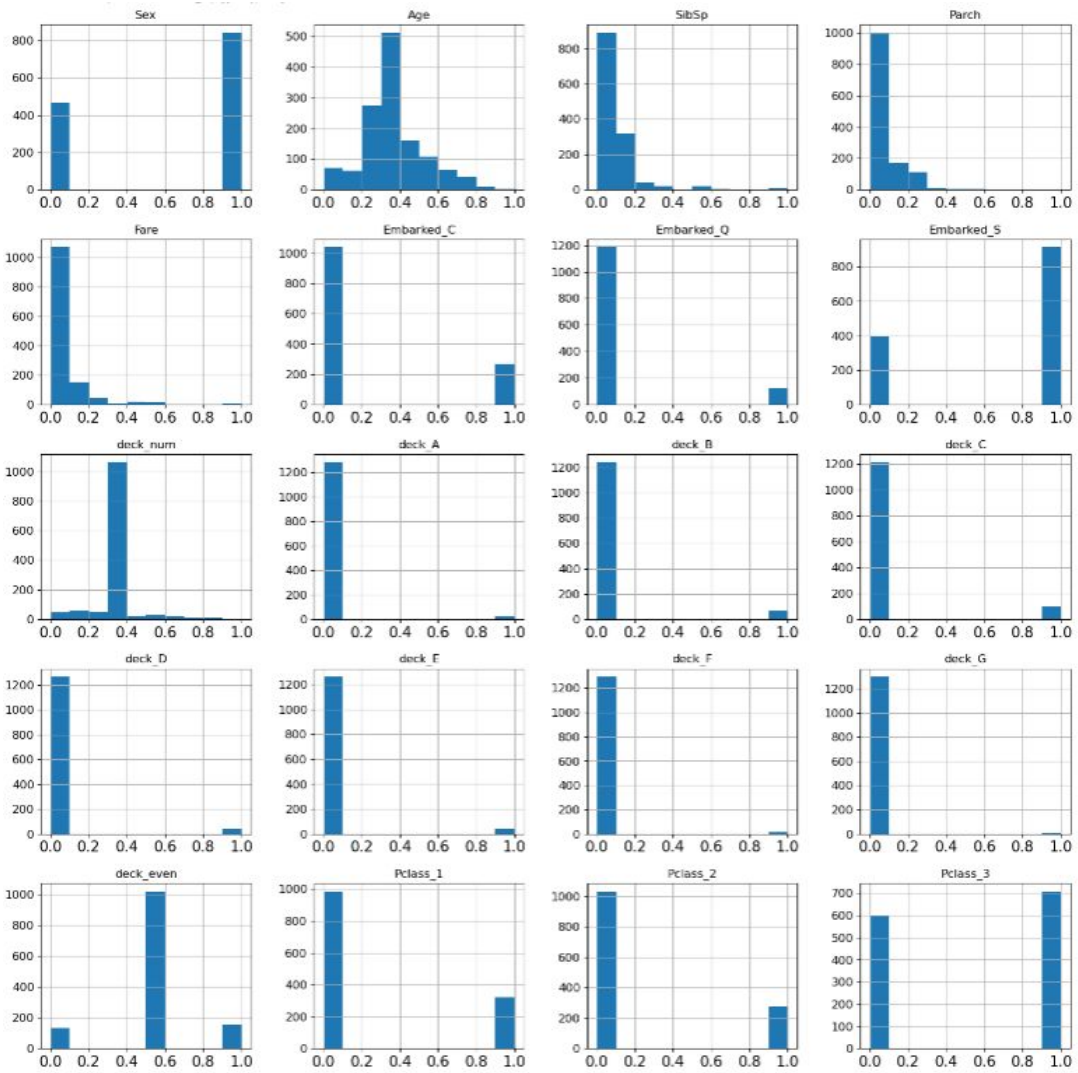


Data Preprocessing

Histograms (normalized NaN-replaced)

Notice the middle column of "deck_even" histogram showing a considerable size of would-be NaNs for just that feature

Notice the difference in the Age and Fare histogram plots between the 3 differently preprocessed datasets



Methods

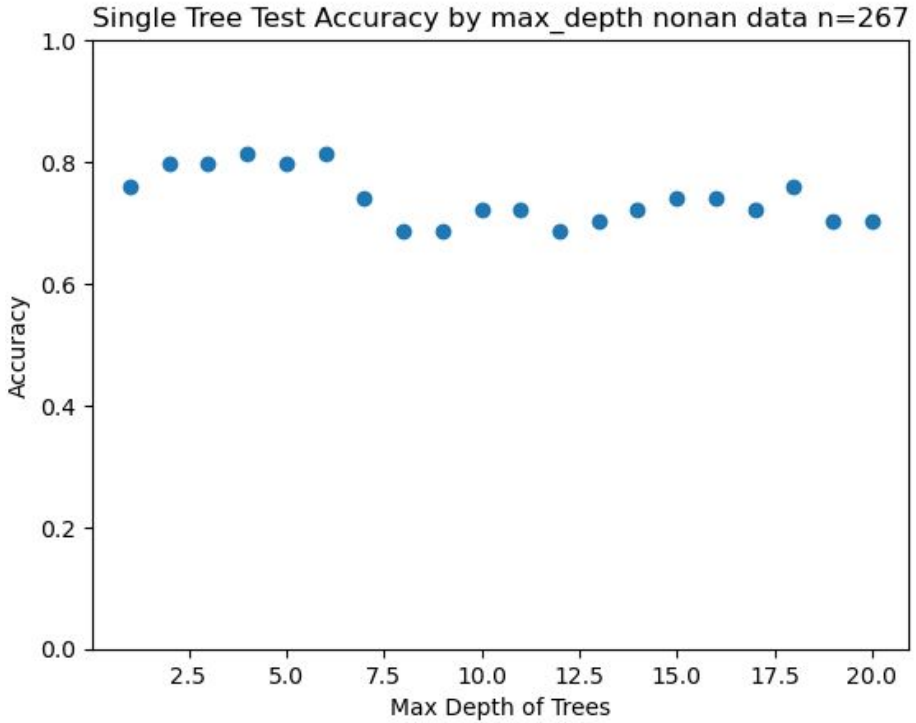
Classifier	Dataset	Max Depth	Max Leaf Nodes	Learning Rate	PCA Max Components
Decision Tree	NaNs dropped (m=267)	1 to 20			
Decision Tree	NaNs replaced (m=1309)	1 to 20			
Random Forest	NaNs dropped (m=267)	1 to 20			
Random Forest	NaNs replaced (m=1309)	1 to 20			
Random Forest	NaNs dropped (m=267) normalized	1 to 20			
Random Forest	NaNs replaced (m=1309) normalized	1 to 20			
Random Forest PCA	NaNs dropped (m=267) normalized	1 to 20			1 to 20
Random Forest PCA	NaNs replaced (m=1309) normalized	1 to 20			1 to 20
Histogram XG Boosted Random Forest	NaNs included (m=1309)	1 to 20		.001 to 25	
Histogram XG Boosted Random Forest	NaNs replaced (m=1309)	1 to 20		.001 to 25	
Histogram XG Boosted Random Forest	NaNs included (m=1309)	1 to 20	2 to 10		
Histogram XG Boosted Random Forest	NaNs replaced (m=1309)	1 to 20	2 to 10		
Histogram XG Boosted Random Forest	NaNs included (m=1309)	1 to 20	1 to 47		
Histogram XG Boosted Random Forest	NaNs replaced (m=1309)	1 to 19	1 to 47		

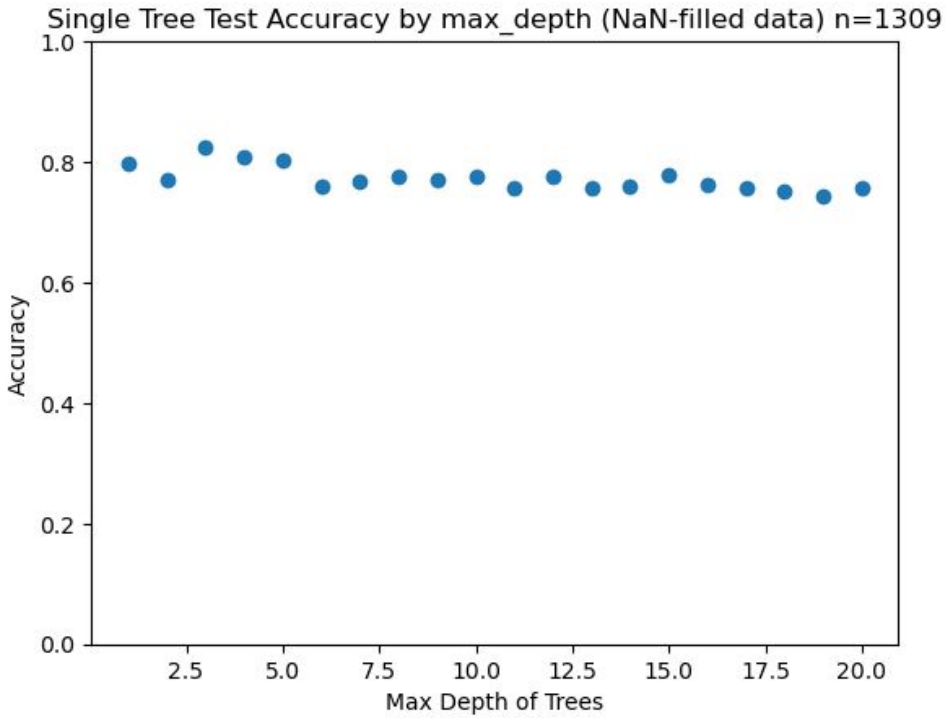
Methods (continued)

Classifier	Dataset	L1 Ratio	Regularization	PCA Max Components
Logistic Regression	NaNs dropped (m=267) normalized			
Logistic Regression	NaNs replaced (m=1309) normalized			
Logistic Regression PCA	NaNs replaced (m=1309) normalized			1 to 20
Logistic Regression PCA	NaNs replaced (m=1309) normalized		L2	1 to 20
Logistic Regression PCA	NaNs replaced (m=1309) normalized		L1	1 to 20
Logistic Regression PCA	NaNs replaced (m=1309) normalized		None	1 to 20
Logistic Regression PCA	NaNs replaced (m=1309) normalized	0.1 to 1	Elasticnet	1 to 20

Results

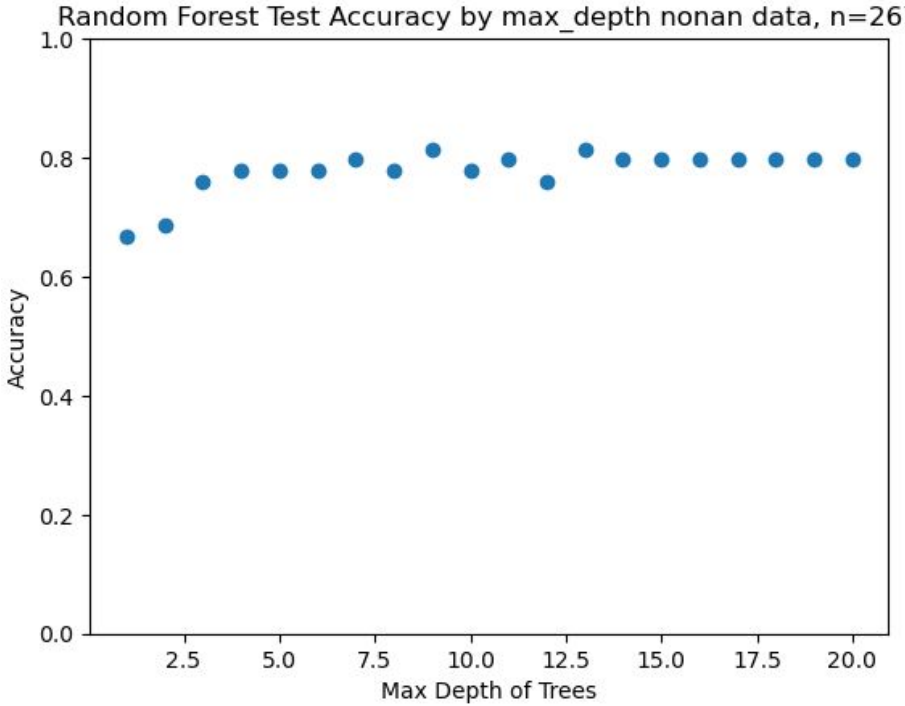
Single Tree

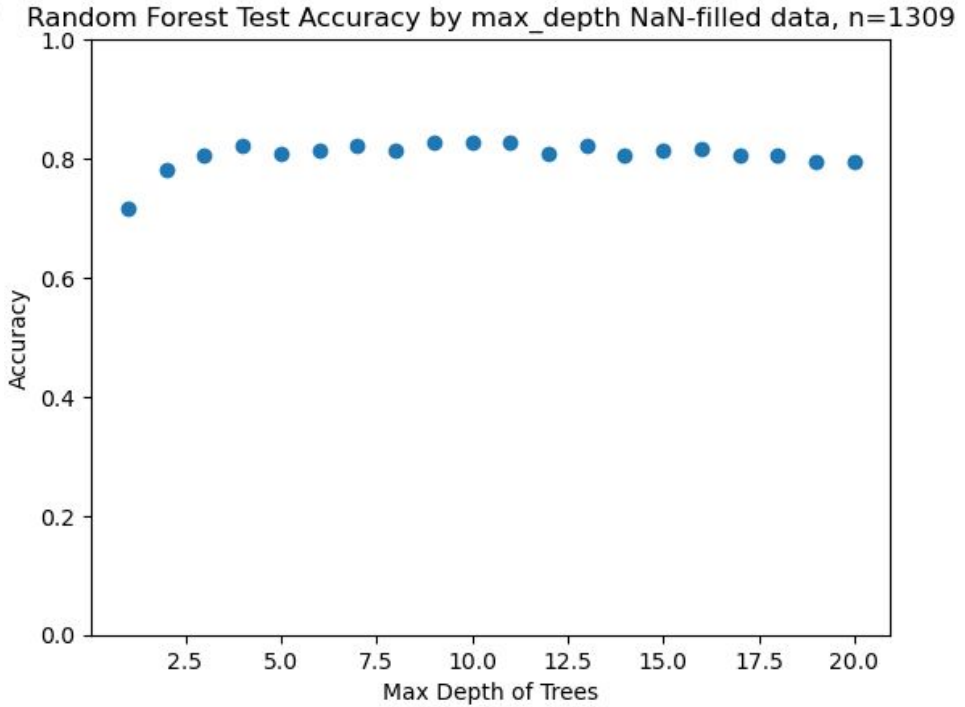




Results

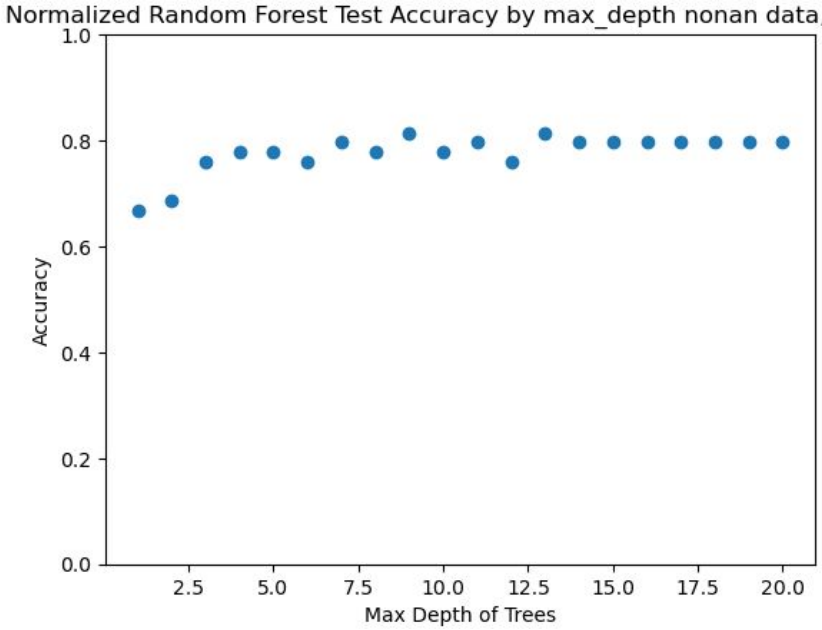
Random Forest

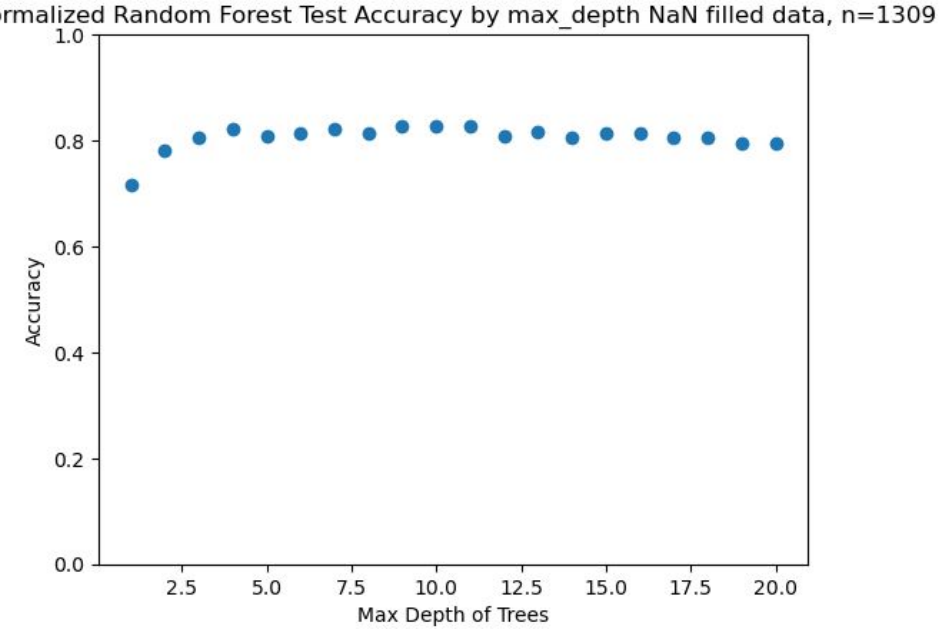




Results

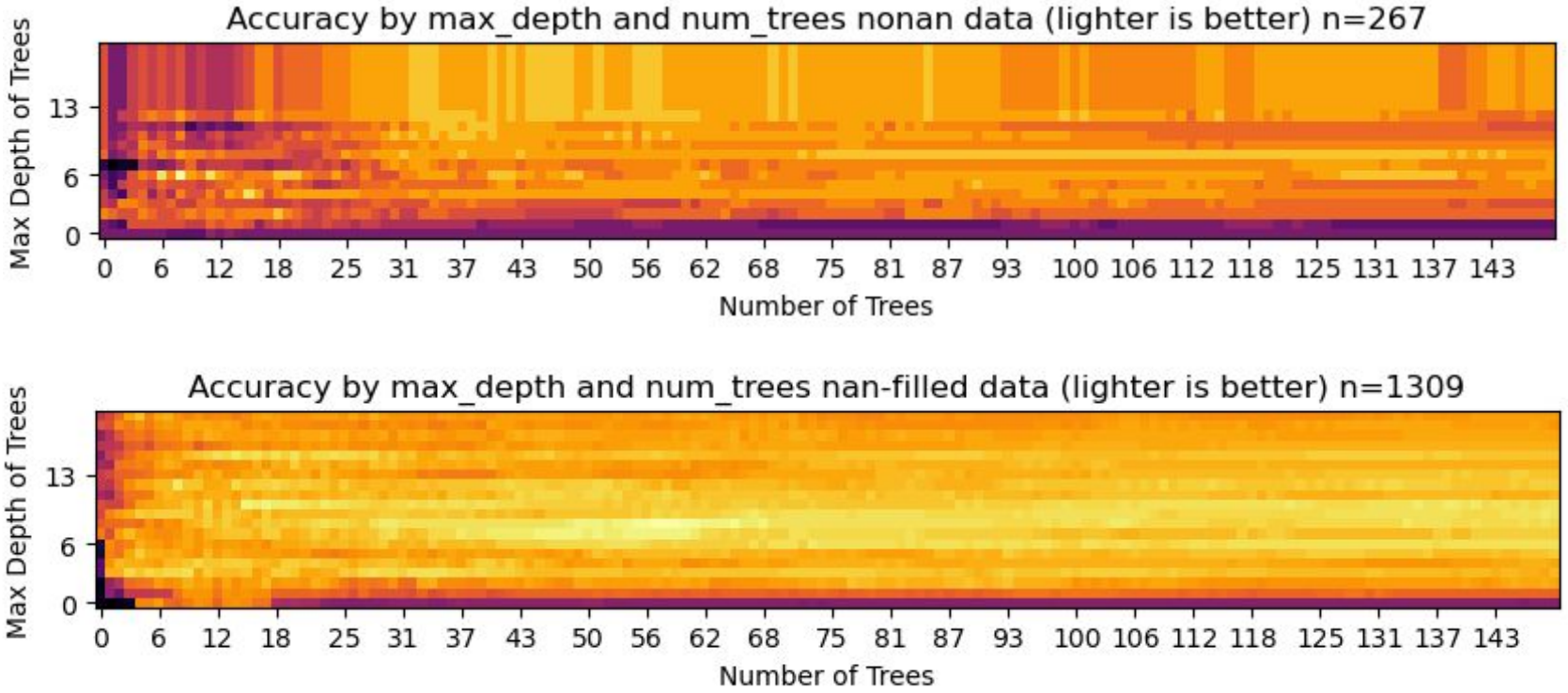
Random Forest (normalized)





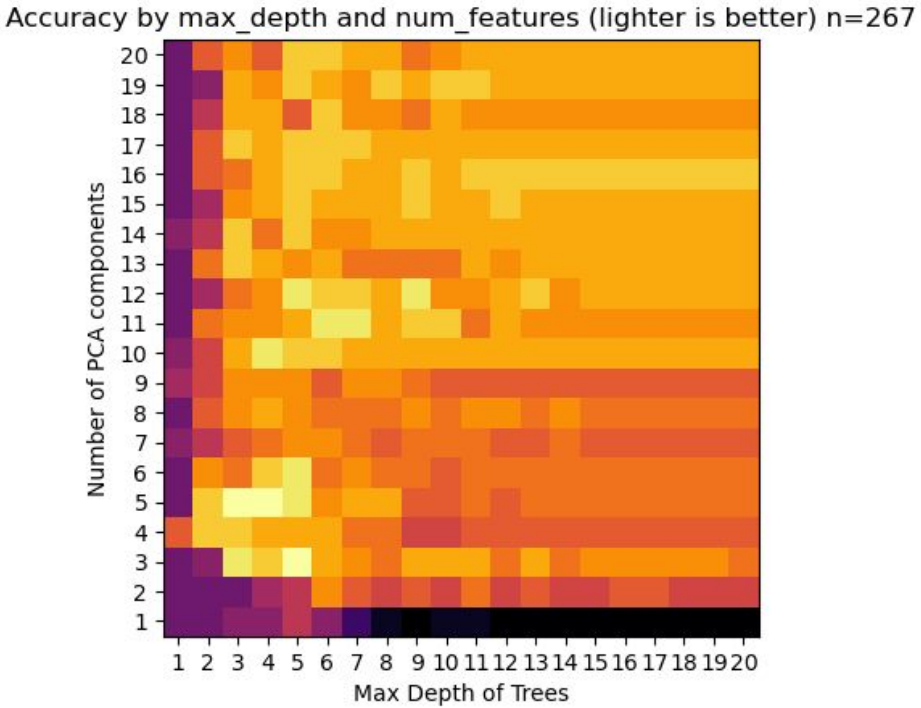
Results

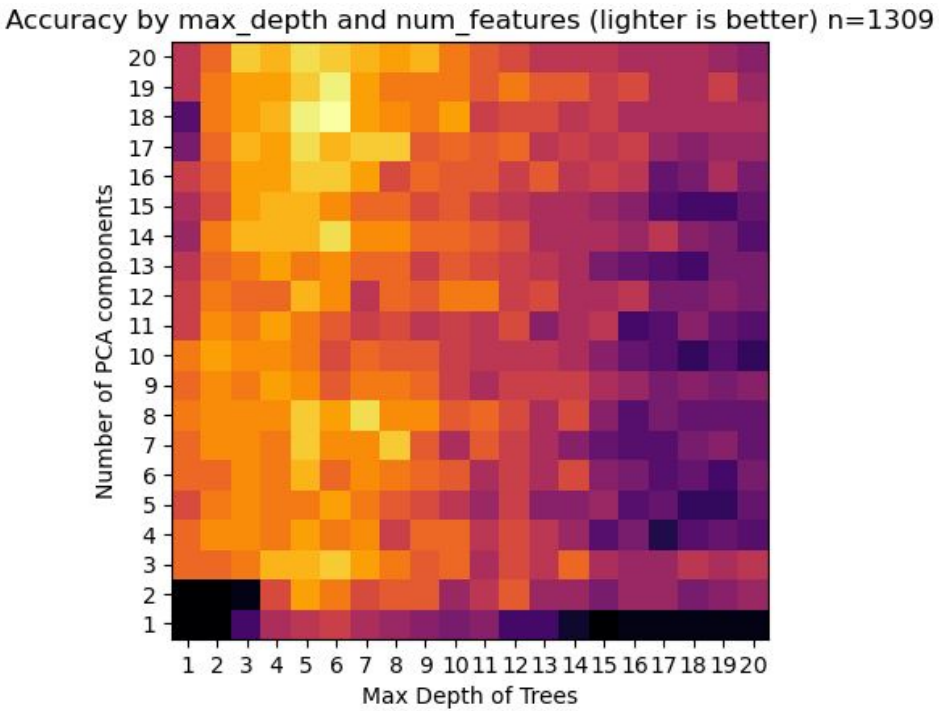
Random Forest (normalized)



Results

Random Forest (normalized) with PCA

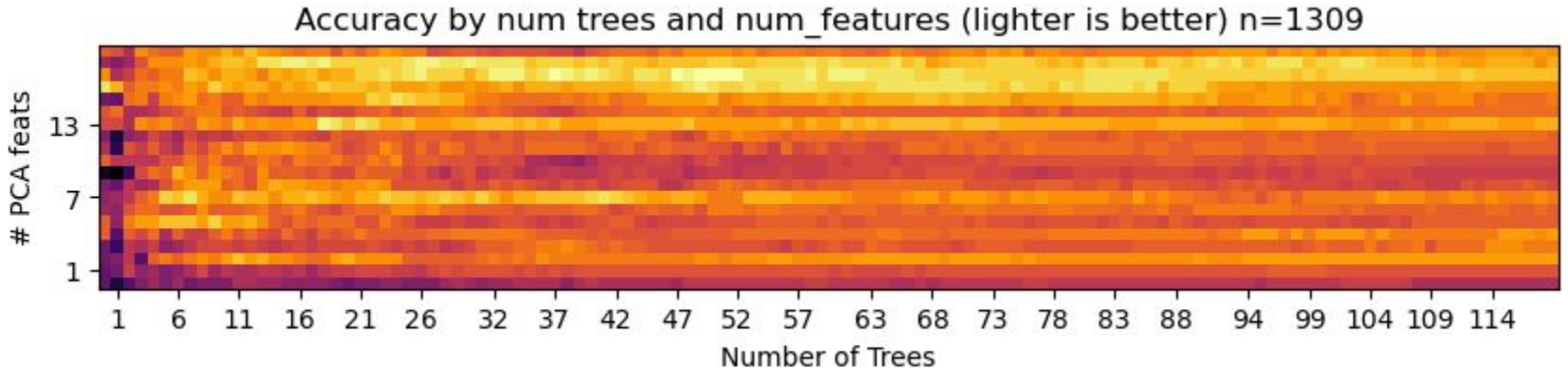




Results

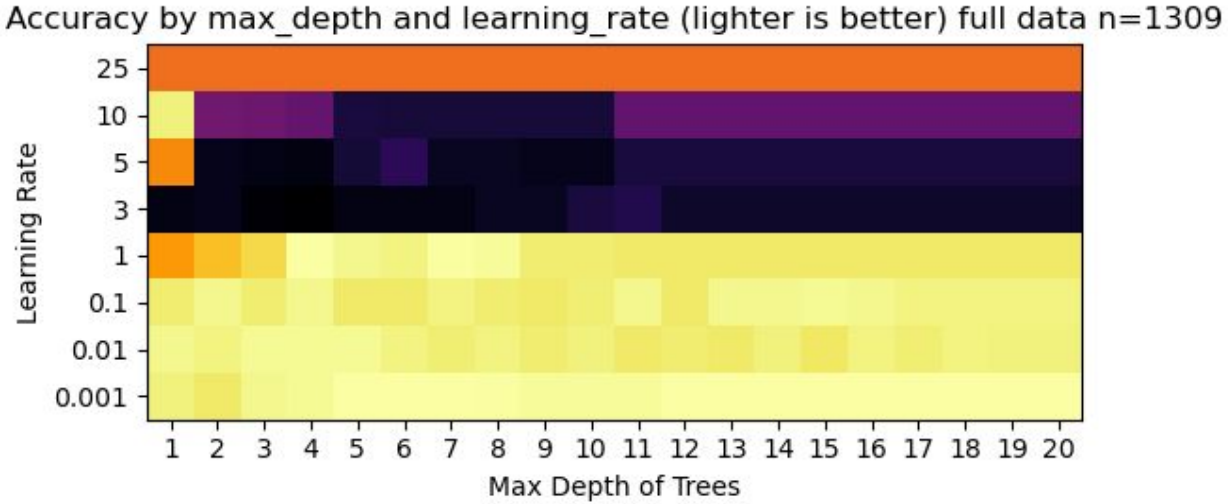
Random Forest (normalized) with PCA (continued)

Let's hold on to look between Max-Depth of [3, 7] and iterate over PCA components and Number of Trees, but we'll only plot the max-depth=6 value (THIS WILL TAKE A LONG TIME IF TRYING LOTS OF VALUES)



Results

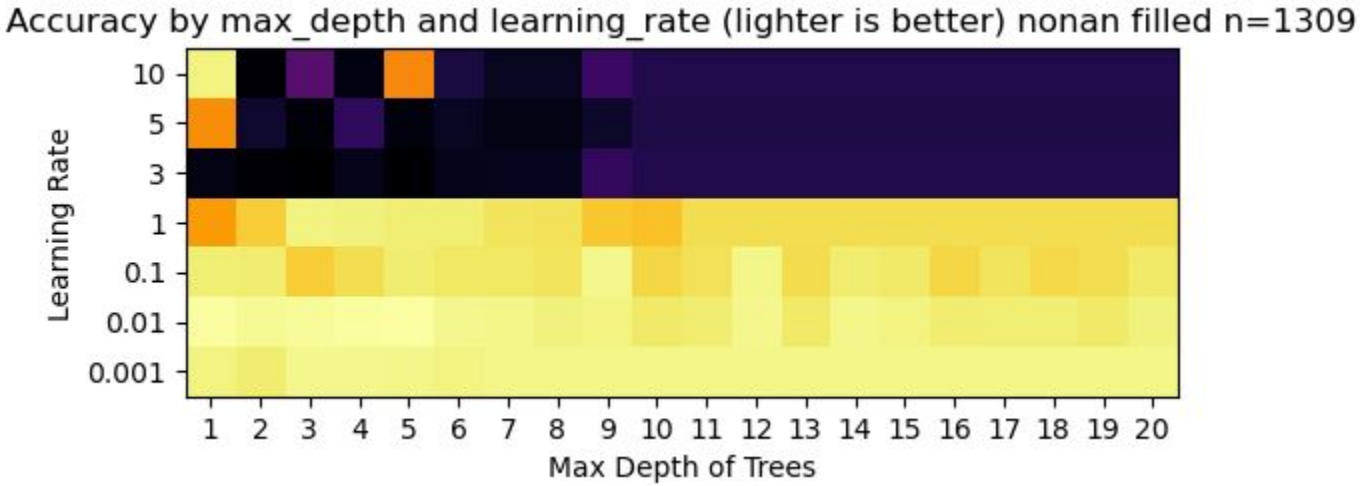
Histogram XG Boosted Random Forest (NaN Included)



lightest horizontal streak is on the second row from the bottom which corresponds to learning rate of 0.01

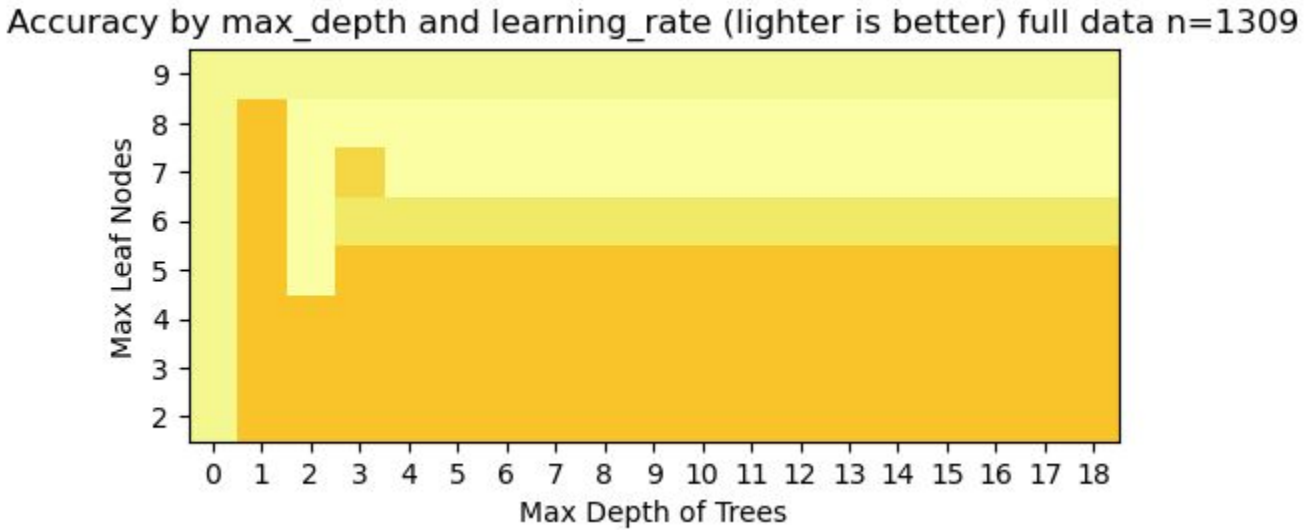
Results

Histogram XG Boosted Random Forest (NaN Replaced)



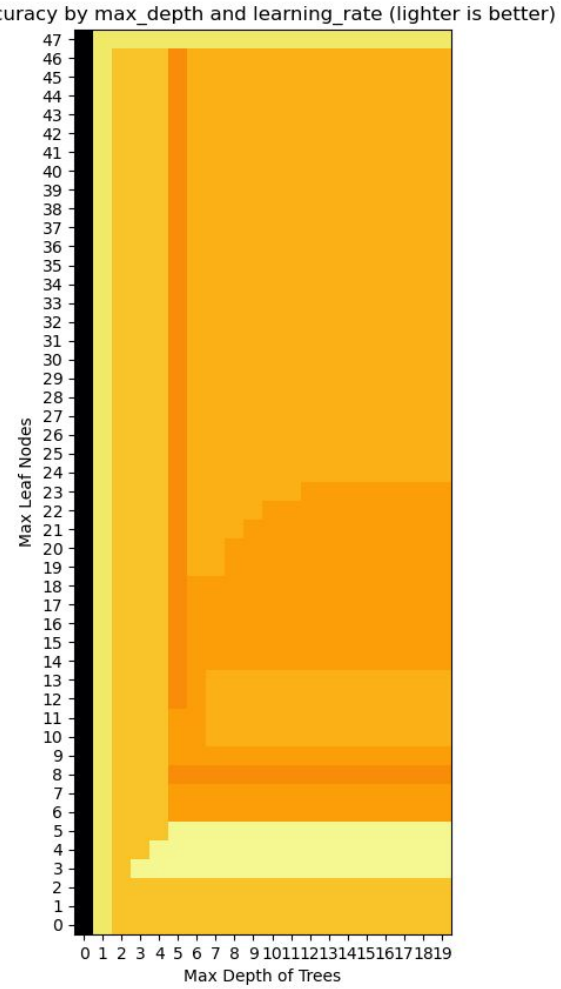
Results

Histogram XG Boosted Random Forest (NaN Included)



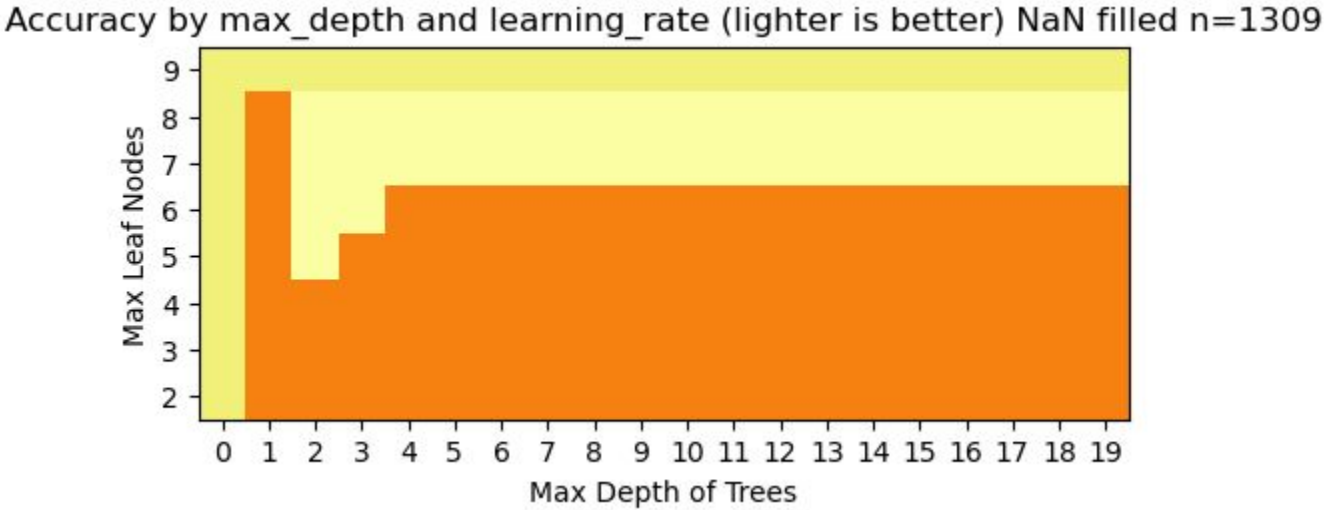
Results

Histogram XG Boosted Random Forest (NaN Included)



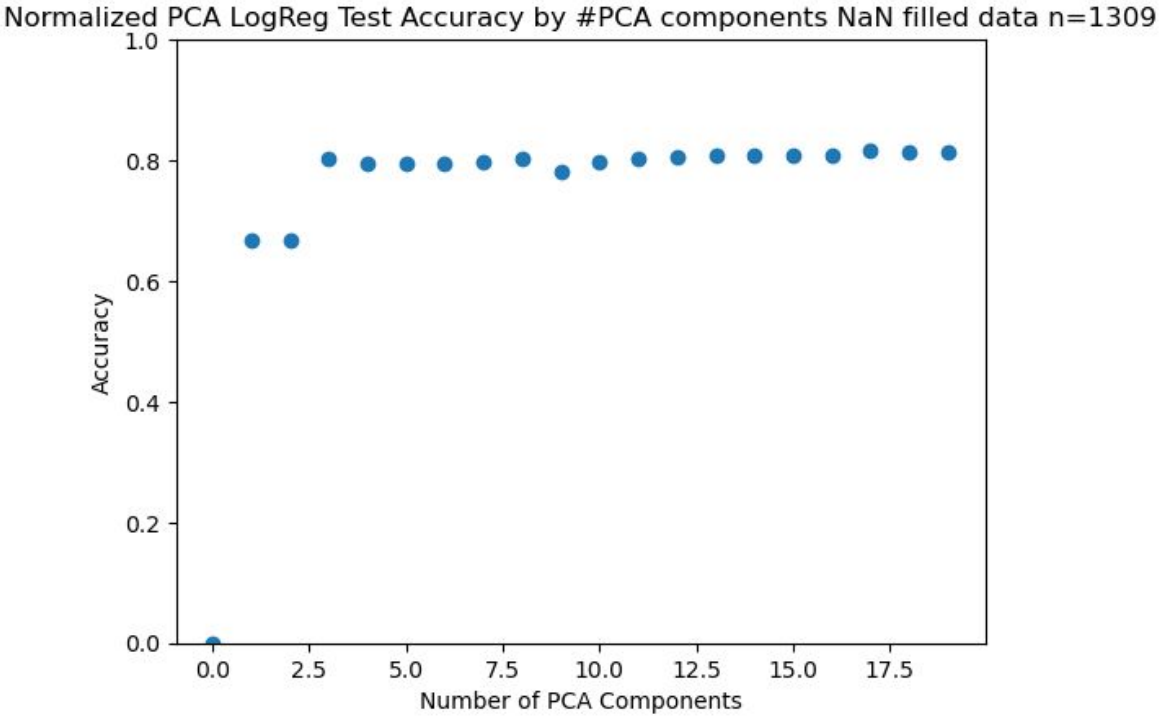
Results

Histogram XG Boosted Random Forest (NaN Replaced)



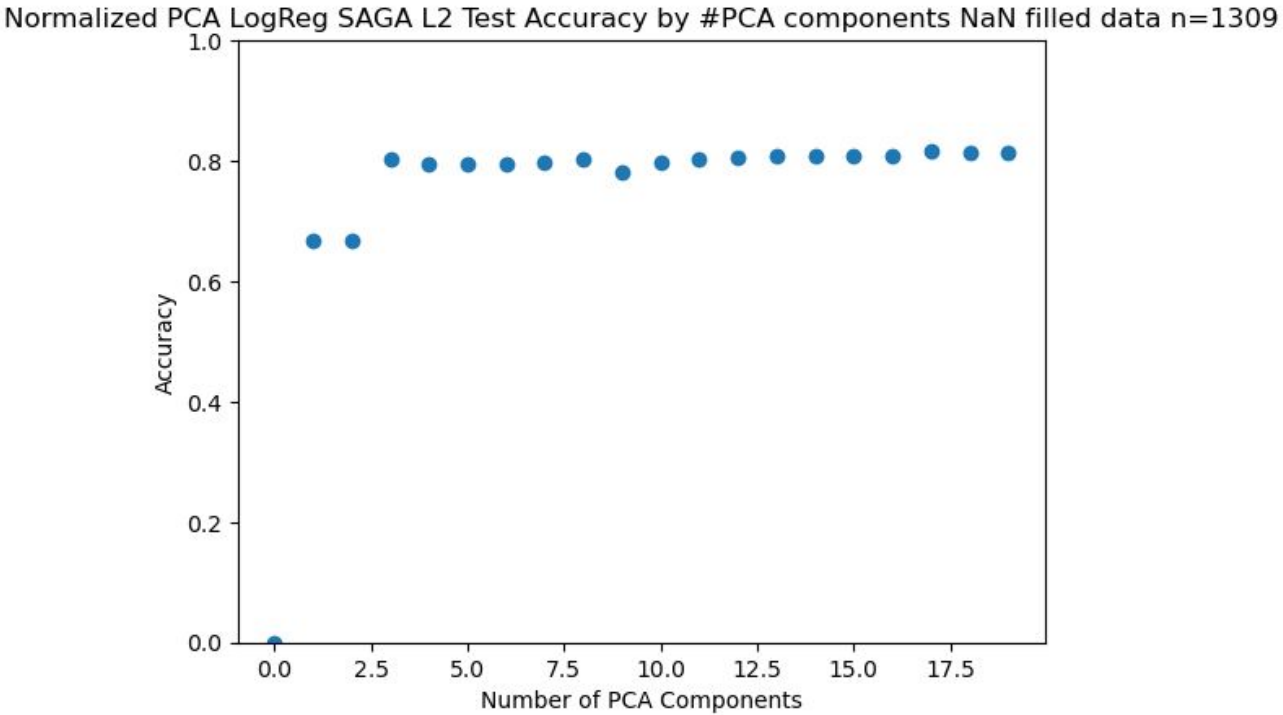
Results

Logistic Regression



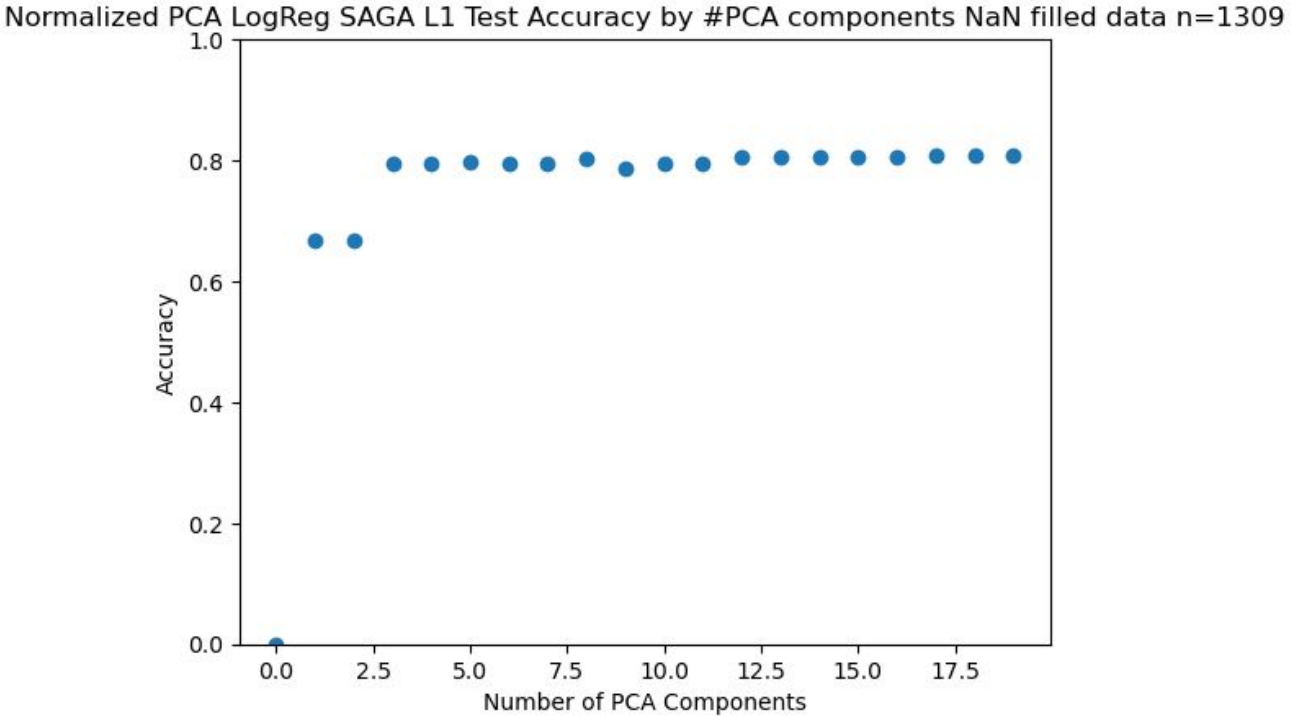
Results

Logistic Regression



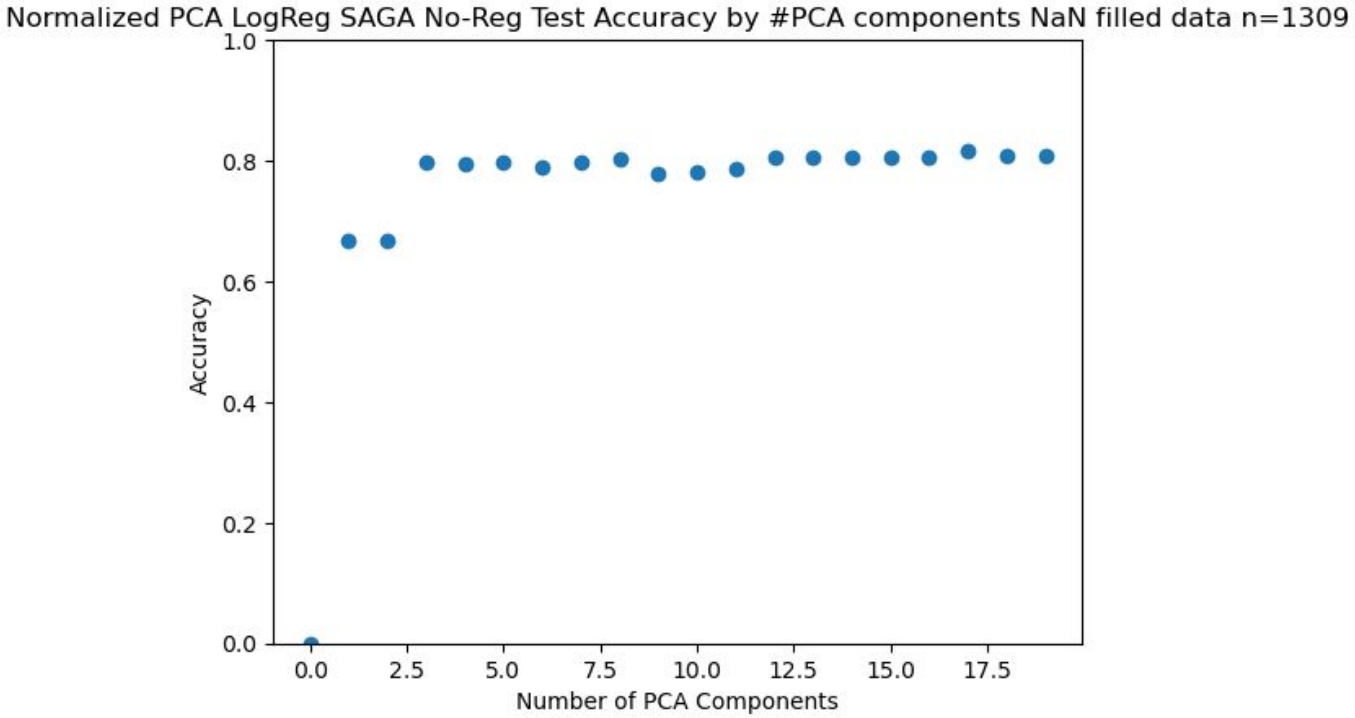
Results

Logistic Regression



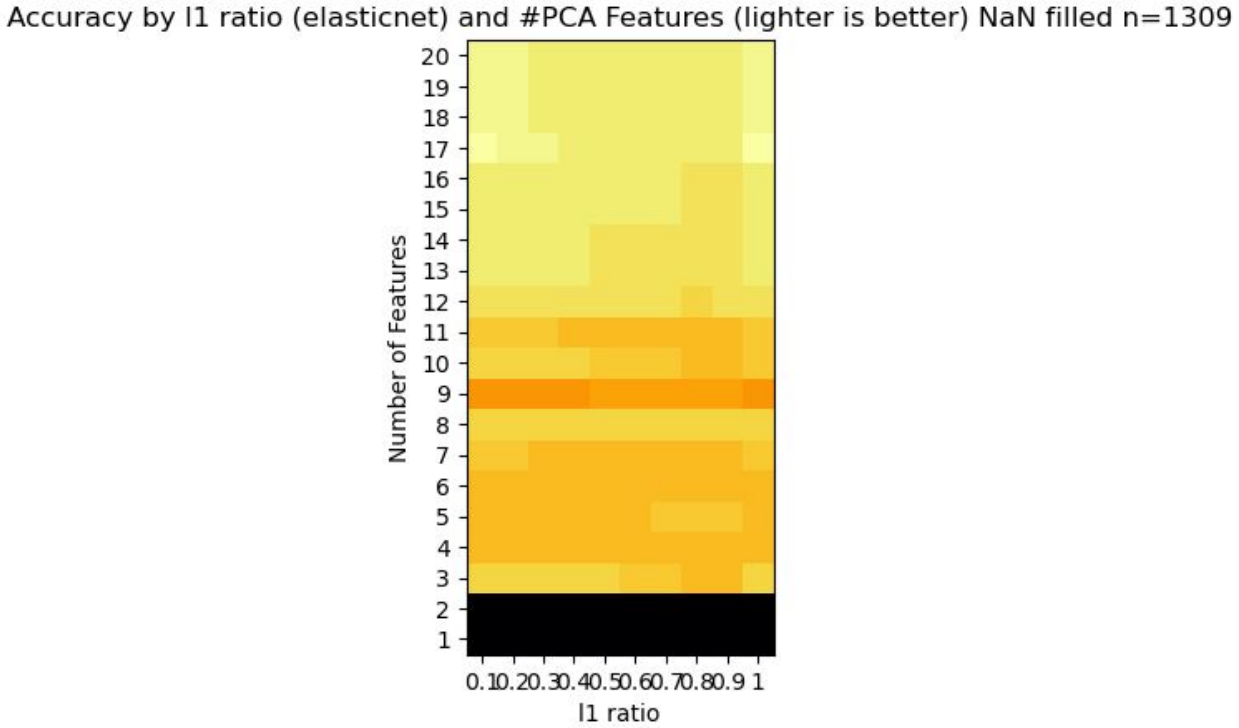
Results

Logistic Regression



Results

Logistic Regression



Results

Dataset	Best Classifier	Accuracy	Models Trained
NaNs included (m=1309)	Histogram XG Boosted Random Forest	82%	305
NaNs replaced (m=1309)	Random Forest	84%	16040
NaNs dropped (m=267)	Random Forest with PCA	87%	3462

Analysis

- Histogram XG Boosted Random Forest performed comparably to Random Forest, so it seems to do a pretty good job with its missing data replacement scheme.
- Although 87% appears to be the best accuracy score, we should be aware that the NaN-dropped dataset (m=267) is significantly smaller, approximately 20% of the complete dataset (m=1309) size.
- We varied the max depth hyperparameter from 1 to 20 and it appears that the best performing decision tree and random forest models had a max depth from 3 to 9.
- The best performing PCA transformed model had a max number of features = 3, but we should be aware that this was on the NaN-dropped dataset.
- The best performing PCA transformed models on the NaNs replaced dataset had a max number of features = 17 to 18.

References

- http://www.recsyschallenge.com/2018/
- https://developer.spotify.com/documentation/web-api/reference/get-audio-features
- https://developer.spotify.com/terms
- https://titanicfacts.net/
- https://www.kaggle.com/competitions/titanic
- https://www.kaggle.com/datasets/vinicius150987/titanic3
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html
- https://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html#logistic-regression-3-class-classifier
- https://upload.wikimedia.org/wikipedia/commons/0/0d/Olympic_%26_Titanic_cutaway_diagram.png
- https://commons.wikimedia.org/wiki/File:Titanic_Stardboard_Side_Diagram.jpg
- https://www.encyclopedia-titanica.org/titanic-deckplans/