



&



Presents

## Prediction of *Titanic* Passenger Survival

STAT5302 Kaggle Assignment

### **Members:**

Harshit Jain <[jain0149@umn.edu](mailto:jain0149@umn.edu)>

Zachary Levonian <[levon003@umn.edu](mailto:levon003@umn.edu)>

Fatemeh Nosrat <[nosra002@umn.edu](mailto:nosra002@umn.edu)>

Shriya Rai <[rai00016@umn.edu](mailto:rai00016@umn.edu)>

Siwen Wang <[wang7610@umn.edu](mailto:wang7610@umn.edu)>

### **Guided by:**

Prof. Jie Ding

Yiqing Shen

# Table of Contents :

1. Abstract.....	
2. Background/Problem Statement.....	
3. Preliminary Data Study.....	
4. Feature Engineering.....	
5. Selected Model.....	
6. Alternative Models.....	
7. Conclusion.....	

## 1. Abstract

We constructed binary classifiers to predict the survival of Titanic passengers for the Kaggle competition [8]. We tried three primary types of models: logistic regression, generalized logistic regression with an elastic-net penalty, and gradient-boosted decision trees. The generalized regression model performed the best, achieving 78.5% accuracy on the test set. Our models include variables with missing values, for which we impute new values, and new variables derived from the raw data provided with the competition. We have made our code available on Github [6].

## 2. Background/Problem Statement:

We chose to pursue the Titanic Kaggle competition, which is a binary classification task to predict the survival of *Titanic* passengers from a subset of the data. In the following sections, we will discuss our preliminary study of the data, the selection of features, and the selection of model approaches.

### 3. Preliminary Data Study

The data was given to us in two files:

- Labeled training set (train.csv)
- Unlabeled test set (test.csv)

The labeled data have 891 passengers and test data have 418 passengers. The “Survived” variable is our response for the prediction while all other variables are evaluated for use as predictors.

#### 3.1 Data Dictionary

We discuss other variables derived from these values in section 4.

The “Null Counts” column gives the number of Null/NA/Missing values in the labeled and unlabeled data respectively.

Variable	Definition	Categorical/Continuous Variable	Null Counts
PassengerId	Id for Passenger	n/a	0/0
survival	Survival	Categorical	0/0
pclass	Ticket class	Categorical	0/0
Name	Name of the passenger	n/a	0/0
sex	Passenger Sex	Categorical	0/0
Age	Age in years	Continuous	177/86

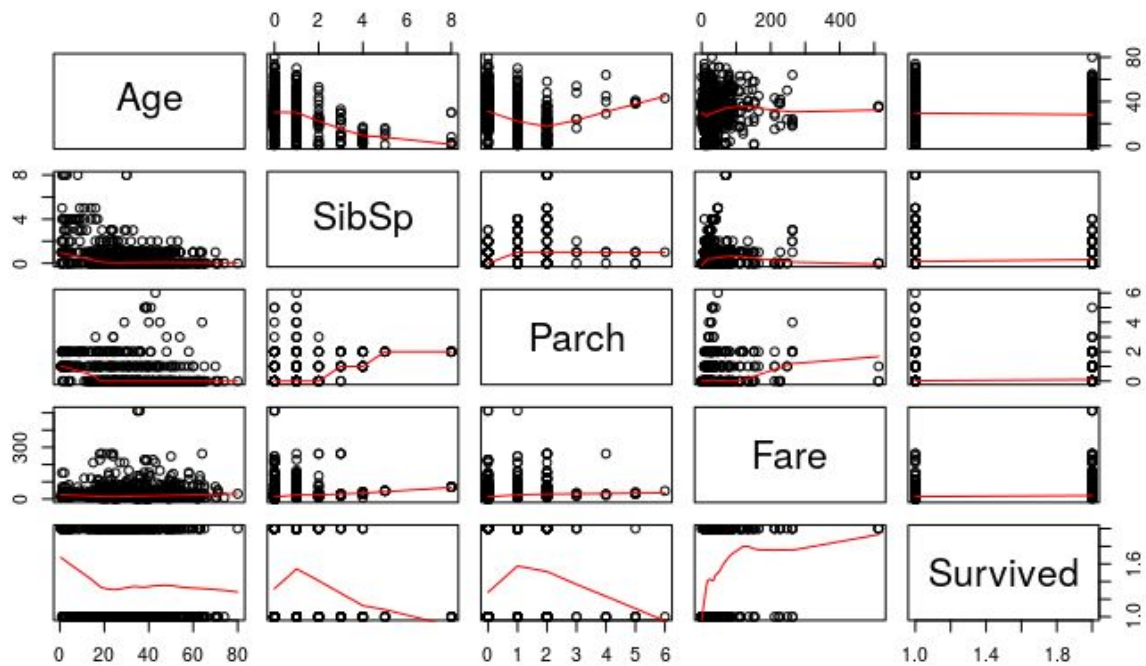
sibsp	# of siblings / spouses aboard the Titanic	Continuous	0/0
parch	# of parents / children aboard the Titanic	Continuous	0/0
ticket	Ticket number		0/0
fare	Passenger fare	Continuous	0/1
cabin	Cabin number		687/327
embarked	Port of Embarkation	Categorical	2/0

\*\* Categorical variables are treated as factors

Below is some further analysis on the continuous variables:

Variable	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Age	0.42	20.12	28.00	29.70	38.00	80.00
SibSp	0.000	0.000	0.000	0.523	1.000	8.000
Parch	0.0000	0.0000	0.0000	0.3816	0.0000	6.0000
Fare	0.00	7.91	7.91	32.20	31.00	512.33

From the above analysis it seems it might be reasonable to factorise SibSp and Parch as the values taken by them are (0,1,2,3,4,5,8) and (0,1,2,3,4,5,6) respectively, which is only a small range of values.



The plot of Age versus SibSp shows that the covariance between Age and SibSp is negative. When SibSp is increasing, the Age is decreasing. Red lines are smooth fits of the relationships between the data. Looking only at the bottom row, we can see a weakly non-linear association between the continuous predictor and survival. This visualization provides initial evidence that this will be a hard problem with low accuracy.

## 4. Feature Engineering:

A few steps we took to work with the features:

1. **Standardizing missing values as NA** : We filled the null and empty values with NA during data loading so that while filling values we have a symmetry between the values which are not available.
2. **Age field** : Age column had 177 missing values from the column. We discuss imputation of age in the next section.
3. **Extracting the passenger titles** : We noticed that each name has a title in it. Examples include Mr., Mrs., Master etc. We discuss derived columns in the next section.

4. **Embarked field:** Embarked had 2 missing values.
5. **Fare Value:** We had 1 missing value in fare.
6. **Dummies:** We created the dummy variables from the categorical fields.

#### 4.1 Imputation of Missing Data

Data is missing from four of the columns. One and two values are missing from Fare and Embarked, 177 values are missing from the Age column, and most rows are missing Cabin information. To build models that handle missing data, a few approaches are commonly used. The first is complete case analysis, where we do a listwise deletion of rows with missing data. The second is to impute the missing values before modeling. We evaluated complete case analysis as well as three imputation approaches, compared in Table 1. Note that complete case analysis was handled in the test data by assigning a 0 to all unlabeled data that do not contain an age. In order to evaluate the appropriateness of these various approaches, it was important to first evaluate the patterns of missingness. In particular, data can be missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR) [1]. Because the only column with a substantial number of missing values is Age, we focus our analysis on that column.

We conducted Little's MCAR test to evaluate the null hypothesis that the missing Age data are MCAR, rather than MAR [3]. The test was significant, so we reject the null hypothesis and conclude that the Age data is MAR (or MNAR). Because we can't test for MNAR and we have no prior information that would tell us *why* this data is missing, we assume the data is MAR and select imputation procedures accordingly. Following Enders' guidance [2], we explore the variables that correlate with a missing value for age. A missing age value is correlated positively with passenger class ( $r = 0.2082$ ) and negatively with point of embarkment ( $r = -0.1672$ ) and passenger fare ( $r = -0.1306$ ). (Other correlations are generally smaller.) We're inclined to think that the true mediator of missing age (among the variables in the dataset) is passenger class, which point of embarkment and fare both correlate with. Because fare is continuous, it was possible to conduct a two-sample  $t$ -test for fare when age is missing vs present. The test was significant at the 99% confidence level ( $t$ -statistic = -6.97,  $p < 0.001$ ), indicating that fare is meaningfully correlated with a missing age value.

The three imputation approaches we evaluated were mean imputation, stochastic regression imputation, and predictive mean matching. Table 1 demonstrates that predictive mean matching performs the best, as evaluated via 50-fold CV accuracy evaluation on the labeled training dataset. This result is expected in part because of the distributions of imputed values, which can be seen in Figure 1. Mean imputation massively decreases the variance for age, creating a huge spike at the mean age. Regression imputation, even stochastic regression imputation, systematically underestimates the variance of imputed data [1]. Furthermore, it can

produce nonsensical values. Observe in Figure 1 that regression imputation produces multiple negative values for age. In contrast, predictive mean matching produces imputed values that are within bounded variance of the non-missing ages.

We used the R package “mice” to use PMM to impute values for Age, Embarked, and Fare. These imputed data were used in all subsequently discussed models. Data for Cabin was handled separately.

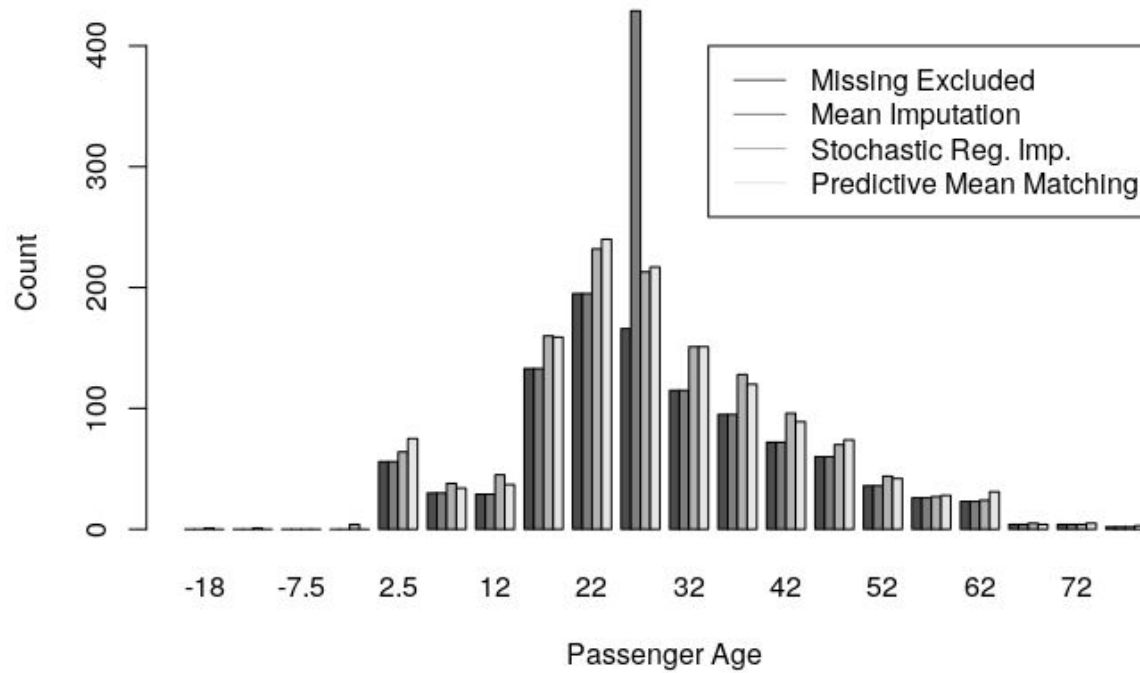


Figure 1: Distribution of Age after various imputation approaches

Imputation Approach	Validation Accuracy
Listwise Deletion	79.27%
Mean Imputation	81.85%
Stochastic Regression Imputation	81.91%
Predictive Mean Matching	<b>82.03%</b>

Table 1: 50-fold CV accuracy for various imputation approaches on Age

## 4.2 Derived variables

Several of the variables in the raw data are challenging to use as-is, so we produced additional variables derived from the raw variables.

- From Cabin, we constructed a factor for the first letter of the cabin, or “o” when the cabin is not assigned. After consolidation, this factor has 6 levels.
- For Ticket, we extracted and normalized the ticket prefixes into a factor. After consideration of the small counts of some ticket prefixes, prefixes occurring fewer than 7 times were lumped into the factor level “other”. These are the 15 produced factor levels and their associated counts: ('digit', 957), ('PC', 92), ('CA', 68), ('other', 53), ('A/5', 25), ('SOTON/OQ', 24), ('W/C', 15), ('STON/O 2', 14), ('SC/PARIS', 14), ('A/4', 9), ('FCC', 9), ('C', 8), ('STON/O2', 7), ('SOC', 7), ('SO/PP', 7).
- For Name, we constructed multiple derived columns. First, we identified titles within the names. We manually mapped lower-frequency titles to custom category names “Military” and “Nobility”. The mapping we created is visible in ColumnExtraction.ipynb. These are the 8 produced factor levels and their associated counts: ('Mr.', 757), ('Miss.', 264), ('Mrs.', 198), ('Master.', 61), ('Rev.', 8), ('Dr.', 8), ('Military', 7), ('Nobility', 6). We also removed the titles and the maiden names from the Name values and computed their lengths in terms of number of characters and number of words. While there is no intuitive usefulness to these columns, we did find them to be generally helpful. We hypothesize that name length may have class correlations not detected in Fare or Pclass.

These additional values were used in all models discussed in the next sections.

## 5. Selected Model: Generalized Linear Model (glmnet)

We fit a model using the R package “glmnet” [7], which implements a fast optimizer for models of the following form:

$$\min_{\beta_0, \beta} \frac{1}{N} \sum_{i=1}^N w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda \left[ (1 - \alpha) \|\beta\|_2^2 / 2 + \alpha \|\beta\|_1 \right]$$

Note that when  $\alpha=1$  this function implements LASSO regression, and when  $\alpha=0$  it implements Ridge regression. With  $1 > \alpha > 0$ , the penalty implements “elastic net” regularization that mixes the two. A grid search was done on  $\alpha$  in (0, 1) and  $\lambda$  in (0.001, 0.4); 20-fold cross validation selected optimal parameters on 80% of the data as  $\alpha=0.5$  and  $\lambda=0.022$ . This model obtained 80.2% accuracy on a held-out 20% validation set after 20-fold cross validation for training [5]. After training on the full set of training data, within-set accuracy on the training set was 83.95% which suggests that the regularization is working: while the accuracy is higher than on the held-out set, it is only modestly so which suggests that the model is not too overfit to the data. Kaggle submission accuracy was 78.468%.

A manual inspection of the accuracy during the grid search suggests that, surprisingly, the value for  $\alpha$  was only weakly associated with accuracy, but generally lower  $\alpha$  performed better. Why is Ridge regression so effective for this problem? Ridge regression is typically used to ameliorate problems with collinearity. Perhaps there are complex interactions among the



variables that are not accounted for by our model, even though the correlations among variables are quite modest. The highest correlation is between the two derived name columns—`title_char_length` and `title_name_length`—but even that correlation is  $< 0.8$  (it is  $r=0.6734$ ).

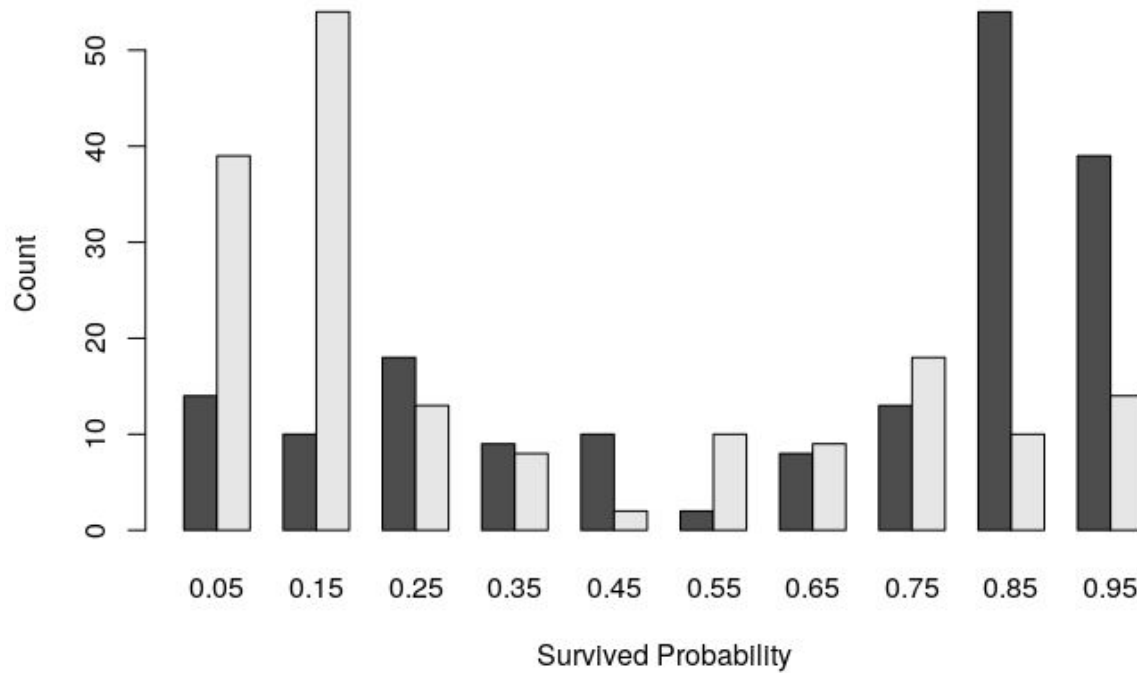


Figure 2: glmnet probability estimates for predicted 0s (black) and predicted 1s (white). The clear bimodal distribution for both indicates that a decent number of the data inputs are still highly confusable to this model.

## 6. Alternative Models

### 6.1 Logistic Regression

We constructed logistic regression models with the raw and derived variables discussed in section 4. While few of the predictors were statistically significant, we found that including all of the variables generally helped. Ultimately, logistic regression did not perform as well as a GLM approach with regularization. One likely reason for this is the size of the data; a shipwreck is an inherently high-variance survival environment, so lots of data is necessary to learn the patterns effectively. With a small dataset, avoiding overfitting is very challenging. Even doing better than a simple linear regression with a logistic link that includes only Age, Pclass, and Sex is quite challenging; ultimately our glmnet approach improved on this model by  $< 5\%$ .

### 6.1.1 Main Effects Model for Different Variables

We used a regular multiple linear regression (non logistic) to explore some of the variables usefulness. The table below describes how adding main effect of different variables affect the RSS and whether there is any significant improvement in RSS with help of ANOVA analysis.

Model	Variable Selected	RSS	RSS Improved (Anova table comparison) (Yes/No)
M0	Only Intercept	210.73	
M1	Added Pclass to M0	186.58	Yes
M2	Added Gender factors to M1	133.25	Yes
M3	Added Age to M2	129.90	Yes
M4	Added SibSp to M3	126.73	Yes
M5	Added Parch to M3	129.01	Yes but not by much
M6	Added Fare to M3	129.89	Yes but very little improvement (by 0.01)
M7	Added both SibSp and Parch to M3	126.63	Yes (Anova comparison with m4 )
M8	Added Fare to M7	126.44	Yes slight improvement
M9	Added factors for Embarked	125.81	Yes

\*\*Anova comparison is performed with the original model and the new model with added regressor to the original model unless specified otherwise

M9 model that includes main effects for Pclass, Gender, Age, SibSp, Parch, Fare and Embarked has the lowest RSS(125.81) and has Multiple R-squared: 0.403.

### 6.1.2 Interactions Between Variables

Consider the Cleansed data. Let “Survived” be the response variable, and “Pclass”, “Age”, “SibSp”, “Fare”, “Sex\_female”, and “Sex\_male” be predictors. We checked the higher-order interactions between variables. First, we checked the interaction

Pclass:Age:SibSp:Fare:Sex\_female:Sex\_male. The p-value for this interaction was not significant, so this regressor was omitted from the model. After checking the NH and AH hypotheses for different models, which have included lower-order interactions, the p-values for most of the interactions between predictors were not significant, and as a result, the NH hypothesis was rejected for each of those higher-order interactions. The only interaction with significant p-value is Pclass:Sex\_female. The p-values for the intercept, Pclass, Age, SibSp, Sex\_female are significant, so they are also in the model. The estimate for the intercept, Pclass, Age, SibSp, Sex\_Female, and Pclass:Sex\_female are 0.825641, -0.156051, -0.007769, -.062993, 0.784236, and -.128154, respectively. Multiple R-squared is 0.4187.

### 6.2 Gradient-boosted Decision Trees (xgboost)

The popular R/Python package “xgboost” was used to train gradient-boosted decision trees (GBTs). The intuition of GBTs is an extension of decision trees to make better stochastic variable decisions than greedy “gain” selections (e.g. the CART algorithm) by incorporating knowledge about the loss surface i.e. the gradient for each variable. We used the R package “caret” to perform a cross-validation grid search over the range of possible values for xgboost hyperparameters [4]. In particular, we evaluated optimal settings for the number of training rounds in (10, 400), the max depth of the tree in (2, 10), and the learning rate  $\eta$  in (0.01, 1). The highest 50-fold CV accuracy was obtained when the number of rounds, the max\_depth, and  $\eta$  are set to 200, 8, and 0.05 respectively. Accuracy on a held-out validation set comprising 20% of the labeled data was 79.1%. After retraining on the full set of labeled data for submission, we observed that the within-set accuracy was 97.76%. This provides some evidence that the GBT model was overfitting to the training data.

Submission of the best model produced a Kaggle submission score of 77.511%. In general, we observed that training for the gradient-boosting was high variance, with a second round of grid search and training producing a final Kaggle score of 77.033% when the tree depth was 6, the number of rounds was 100, and the learning rate was 0.05. We attempted to decrease the overfit of the model by decreasing the number of rounds and the learning rate, but such approaches produced worse accuracies overall. None of us are experienced enough with such gradient-boosting methods to know why this method is performing badly, but we speculate that the overfitting was the biggest reason for lackluster performance.

## 7. Conclusion

We discovered that binomial logistic regression with elastic-net regularization performed the best on the Kaggle Titanic competition. We discovered that there is an association between

passenger survival and their socio-economic status that is mediated by the primary predictor of survival, passenger sex. In general, prediction of survival is a hard problem, with survival being intrinsically a high-variance affair; there is no combination of variable values for which survival was guaranteed or death was assured.

We learned that imputation of missing data is a hard problem with a rich literature and a variety of approaches [1]. Ultimately, utilizing predictive mean matching to impute Age data improved our model's performance. We learned that variables that seem to have little use can add value to the model if additional features are derived from them: passenger title, cabin number, and ticket type all improved the model.

Although 800 labeled data seems like a lot, overfitting was a big problem for us. Ultimately, we performed best with the model that includes built-in regularization, which provides evidence that other approaches would be more effective if we had more data or additional more-predictive variables.

One way to improve the modeling for this problem is to incorporate other relevant data. For example, passenger survival data from other early-20th century shipwrecks could improve the generalizability of our model. Other data external to the competition could prove useful; for example, while we were able to get some value from the name column, the addition of information that captures the social-class associations of particular names could provide a stronger signal on passenger class than Fare.

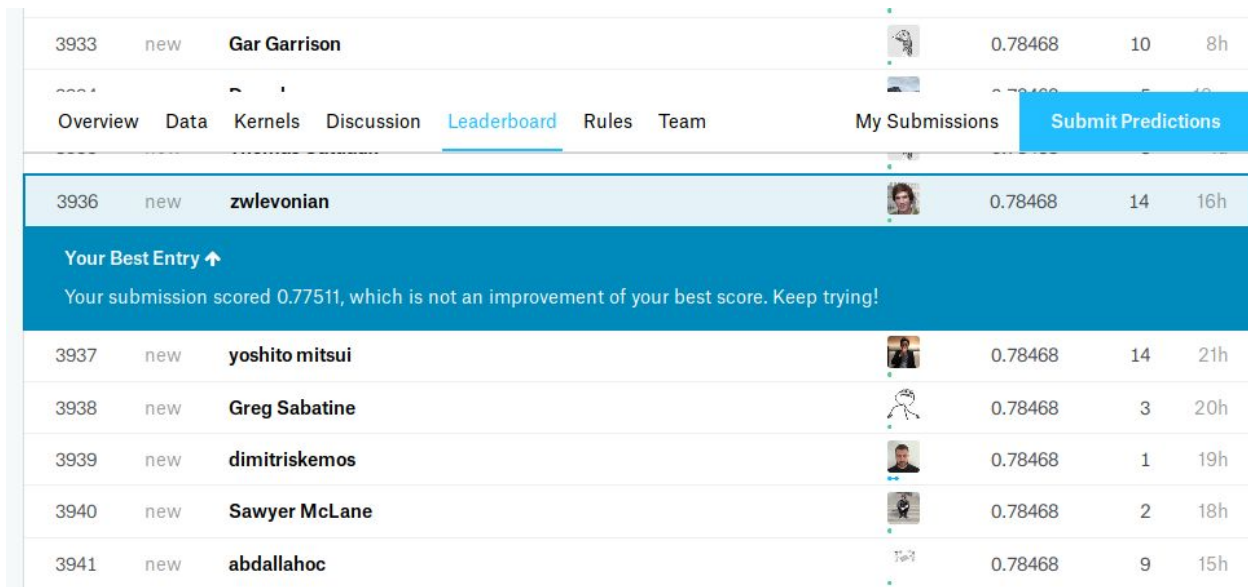
We did not have the time to evaluate other promising models: ensemble models and neural networks top this list. In particular, we may be able to perform better just by ensembling together the three models discussed here, which would be a valuable opportunity for future work. While we conducted some analysis of the relationships between variables, a more sophisticated attempt at inference could increase our domain knowledge and lead us to include a more appropriate subset of variables.

## References

- [1] Stef van Buuren. 2018. *Flexible Imputation of Missing Data, Second Edition*. Chapman and Hall/CRC. DOI:<https://doi.org/10.1201/9780429492259>
- [2] Craig K. Enders. 2010. *Applied Missing Data Analysis*. Guilford Press. Retrieved from <http://www.appliedmissingdata.com/>
- [3] Roderick J. A. Little. 1988. A Test of Missing Completely at Random for Multivariate Data with Missing Values. *Journal of the American Statistical Association* 83, 404 (December 1988), 1198–1202. DOI:<https://doi.org/10.1080/01621459.1988.10478722>
- [4] <https://www.kaggle.com/nagsdata/simple-r-xgboost-caret-kernel>
- [5] <http://www.milanor.net/blog/cross-validation-for-predictive-analytics-using-r/>
- [6] <https://github.com/levon003/kaggle-titanic>
- [7] [https://web.stanford.edu/~hastie/glmnet/glmnet\\_alpha.html](https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html)
- [8] <https://www.kaggle.com/c/titanic>

## Screenshot of Leaderboard Score

A screenshot of our best score on the leaderboard appears below. The “just-submitted” score of 0.77511 is from the xgboost model.



The screenshot shows a Kaggle leaderboard for a competition. The 'Leaderboard' tab is selected. A blue banner at the top of the submission list states: 'Your Best Entry ↑ Your submission scored 0.77511, which is not an improvement of your best score. Keep trying!'. The table lists several other submissions, all with a score of 0.78468.

Rank	Status	Username	Score	Count	Time
3933	new	Gar Garrison	0.78468	10	8h
3936	new	zwlevonian	0.78468	14	16h
3937	new	yoshito mitsui	0.78468	14	21h
3938	new	Greg Sabatine	0.78468	3	20h
3939	new	dimitriskemos	0.78468	1	19h
3940	new	Sawyer McLane	0.78468	2	18h
3941	new	abdallahoc	0.78468	9	15h

## Code

All code and derived data for this project is available in this Github repository:

<https://github.com/levon003/kaggle-titantic>

Code is in both R and Python, contained in the `code/r` and `code/python` directories respectively.

- `ColumnExtraction.ipynb` - The only Python code in our repository, this file produces the new columns discussed in section 4.2.
- `data_prep.Rmd` - Primary data prep notebook, including constructing factors from string data, the merging in of the new columns, and imputation of missing age data. Also includes the cross-validation experiments to determine imputation accuracy. Also includes some preliminary data study, including between-variable correlations and plots.
- `model_comparison.Rmd` - Primary model comparison notebook. This notebook includes code samples implementing logistic regression models, glmnet models, and xgboost models.
- `Titanic_shriya.R` - Preliminary Data Study, basic data Cleansing and Main Effects Anova Comparison
- `titanic.Rmd` - Initial Sample to construct logistic model for test submission to kaggle
- `Titanic_harshit.R` - Preliminary Data Study, Feature Engineering, Data modelling and random forest implementation

- Titanic-Fatemeh Nosrat.Rmd - Preliminary Data Study, Checking the second- and higher-order interactions between the predictors, Pclass, Sex\_female, Sex\_male, Age, Fare, SibSp.