Titanic Survival Prediction

Problem Statement

< 1 > Can you explain what are the characteristics of this case? Is it considered as supervised or unsupervised learning? Please explain it by pointing out the aspects of data!

Given a dataset with several variables in it (survival, ticket class, name, gender, etc.). Objective: Predict whether a passenger survived the sinking of the Titanic or not.

This is a binary classification problem. The goal of this case is to predict whether a passenger survived the sinking of the Titanic. We have a variable 'survived' in the dataset with possible values 0 or 1 to state whether a passenger survived. We will need to use a supervised learning model. Each data has been labeled survived=0 or survived=1. Using other variables in the dataset, our model will learn the likelihood a data labeled as survived=0 or survived=1.

Data Collection

In this case, we already have the dataset ready to download here:
http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic3.xls

Data and Features (Data Wrangling, Exploration, Transformation)

< 2 > What are the information in data that can be considered as features? Please explain the reasoning and make reader can understand easily (i.e.: add visualization)!

< 3 > Do you need to create any new features? If so, please explain the rationalization!

Generally, a variable with high correlation to the target (survived) is a good candidate to be included as a feature. It is best if the feature has high correlation with the target and at the same time it has low correlation with other features, that means it is less redundant and it has significant impact to the learning. We don't actually need to assume too much, the model sometimes combines some seemingly weak features and find a relation from them.

It is better to have more number of strong features that can support the model. We can create a good feature from a feature that seemingly weak.

pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
3	1	Peter, Miss. Anna	female	NaN	1	1	2668	22.3583	F E69	C	D	NaN	NaN
3	0	Lundahl, Mr. Johan Svensson	male	51.0	0	0	347743	7.0542	NaN	S	NaN	NaN	NaN
2	0	Moraweck, Dr. Ernest	male	54.0	0	0	29011	14.0000	NaN	S	NaN	NaN	Frankfort, KY
2	0	Matthews, Mr. William John	male	30.0	0	0	28228	13.0000	NaN	S	NaN	NaN	St Austall, Cornwall
3	0	Conlon, Mr. Thomas Henry	male	31.0	0	0	21332	7.7333	NaN	Q	NaN	NaN	Philadelphia, PA

pclass, sex, embarked

They are categorical data. I'd need to manipulate them into another format so that the model can process it. Using pandas.get_dummies, they'd be converted into several new features (sex_male, sex_female, etc) each has binary value 0 or 1. Pic 002 shows that female passengers have significantly higher chance of survival than male passengers, that means sex can be a strong feature for the model.

name

We can't directly process 'name'. I created features **'name_length'** and **'title'** from it. Someone with longer name might have a high social status, thus higher priority for getting a boat. Title shows the gender, age, married or single. Those are assumptions and I didn't know for sure if they'd be strong features, it's good to have them and later evaluate them e.g. find the feature importance.

age, fare

It is a good idea to split 'age' and 'fare' into several bins, this can minimize outliers as they'd be included in the bin with other data. I created **'age_bin'** and **'fare_bin'**.

sibsp, parch

From these features, I created '**family_size'** the sum of the passenger, 'sibsp', and 'parch'. I also created 'is_alone' to tell whether passenger traveling alone or with family. It is good to have them and evaluate those features. The feature **'is_alone'** has higher correlation to 'survived' than 'sibsp' or 'parch' to 'survived'.

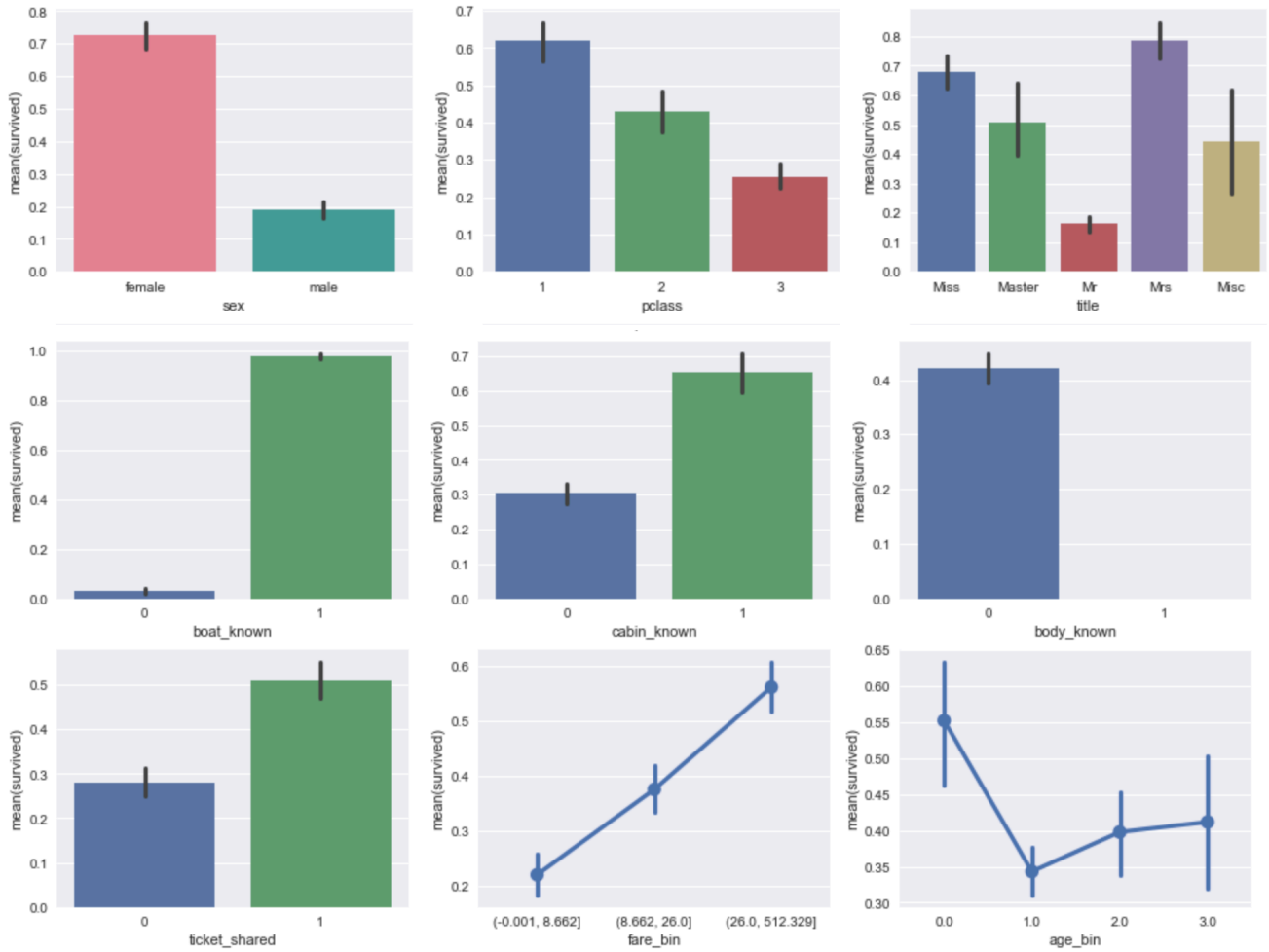
ticket

I found duplicates in the ticket numbers. Passenger with same ticket numbers might traveled together, I created **'ticket_shared'** to show this. I also created **'group_count'** to depict the number of passengers within the same ticket group.

cabin, boat, body, home.dest

They have so many null values. That doesn't always mean we cannot use them. A passenger with body identification means the passenger did not survive. The number of null values in feature 'boat' is close to the number of feature 'survived' with value 0. That does make sense, the

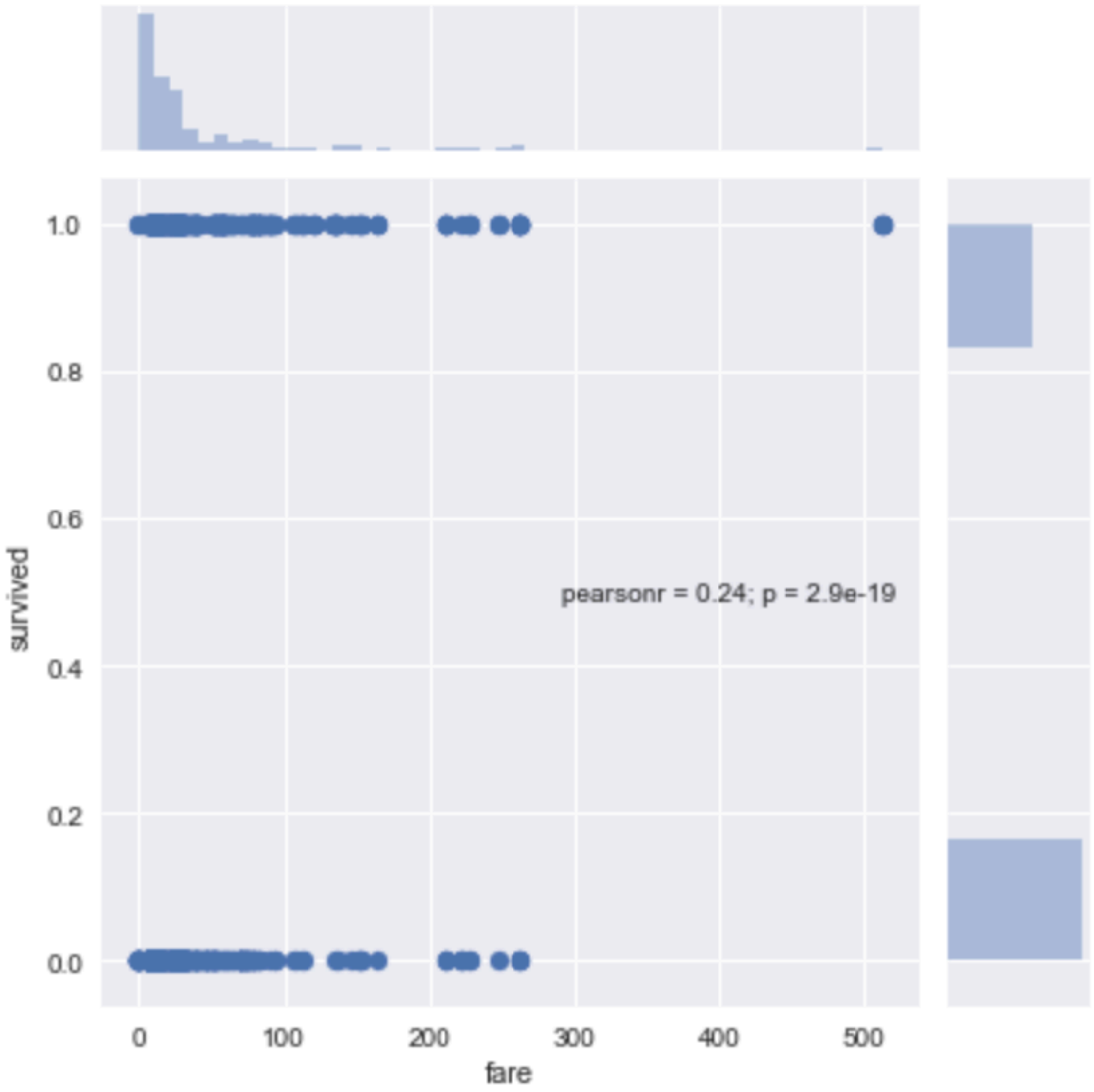
passengers that got into a boat are likelier to survive. I dichotomized ‘boat’, created **‘boat_known’**. It has binary value 0 or 1, ‘boat_known’ = 0 if ‘boat’ has null value. I did the same to ‘cabin’ and ‘body’, created **‘cabin_known’** and **‘body_known’**. As expected, passengers with known boat number are a lot likelier to survive. Feature ‘cabin_known’ and ‘body_known’ also relatively have a high correlation to ‘survived’ (see pic 003). The number of null values in ‘home.dest’ however, is not similar to the number of survived/not survived passenger. I can’t deduct anything from this feature, therefore I’d remove ‘home.dest’ to cleanup the data.



< 7 > Do you notice any outliers or noise in the data? If you do, please explain how you notice it and what you are going to do with the outliers!

From this plot we can see there are a few samples with fare above 500. I investigated further. I found that passengers that shared the same ticket number have a lot higher fare. Seems that the fare is accumulation of prices form each individual in the same group. So, I updated the fare by dividing it with the number of the people in the group. Now the data seemed better. There were

still some samples with fare above 100, far from the mean value. I assigned new values to them, assuming the data are noises.



< 4 > Do you need to scale any of the features? If so, please explain the rationalization!

Feature 'fare' was skewed, so I need to perform feature scaling. The skewed data will harm the performance of certain models since some data might overvalued and the rest will be hardly distinguishable.

< 5 > Please recap the missing values on the dataset. What will you do with the missing data? What do you think the best action in this case: clear row, clear column, ignore, or impute the data? Please explain the action you take and the rationalization. Please explain further, by explain other cases that other action may work better!

```
pclass          0
survived        0
name            0
sex             0
age           263
sibsp           0
parch           0
ticket          0
fare            1
cabin        1014
embarked        2
boat          823
body         1188
home.dest     564
```

I imputed the null values in 'fare' and 'embarked', the other columns in these samples are still good and might be valuable. I dropped the columns 'cabin', 'boat', 'body', and 'home.dest' as there are too many null values in these columns. Before I deleted these columns, I created new features 'cabin_known', 'boat_known', 'body_known' as I mentioned earlier. Arguably column 'age' has a lot of null values. I tolerated it, so that it can be processed further, but kept this in mind. If this feature turned out not really helpful, I can drop it. We cannot just ignore the missing values and then include them in the learning process, since the model cannot handle them. Clear row works better if a lot of other columns from the same row are also corrupted.

Data Modeling

< 6 > Intuitively, what are first three algorithms that you will to create prediction model for this dataset? Please explain why you choose the algorithms as well!

The dataset is relatively small. I did not worry too much about execution time. We are at a risk of overfitting with this small dataset. Any ensemble classifier would be a good bet. I choose Extremely Randomized Trees. This classifier has one more level of randomness than Random Forest. As mentioned in scikit-learn documentation (http://scikit-learn.org/stable/modules/ensemble.html): "As in random forests, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these random-generated thresholds is picked as the splitting rule". This might result in slightly less accuracy, but ensure less overfitting, meaning more confidence in predicting unseen data.

My next two models are XGBoost and Support Vector Machines (SVM). XGBoost is a type of Gradient Boosting with better performance for low resource (using distributed training on cloud). Gradient Boosting basically allows a collection of weak learners to solve small problems, then

combines them to solve bigger problem. SVM uses hyperplane to do the classification. One of my reason to choose them is because they have distinct algorithms, so that three of them can make a good ensemble. Unrelated models possibly don't make errors at the same sample. Using voting, they can correct each other, resulting in a boost of accuracy.

Evaluation

< 8 > How do you measure the quality of your prediction? What kind of metrics' score will you use?

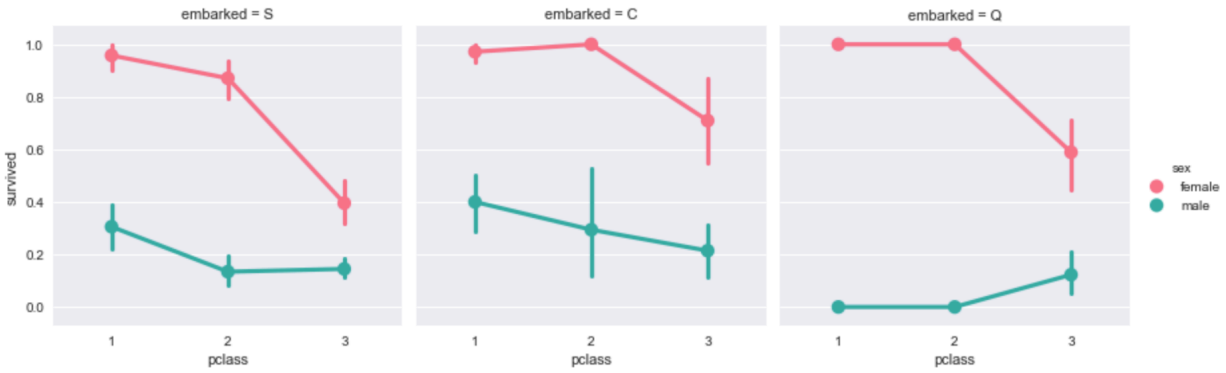
I used accuracy score as the metric. It is a percentage of correctly predicted data. I did the measurement using cross validation. It divides the data into several folds, uses a fold unseen as a test data and repeated them for all folds. I then got several scores, I could see if the model overfit from the variance of the scores. Low variance of scores means the model is in a good shape.

I also used Randomized Search to choose the correct parameters for the model. It chose the best possible set of parameter by randomly try the options.

< 9 > Are there any insights you get when you explore the data? If there are, please explain it and tell us why it might be a good insight!

Several insights that I got:

- Female passengers have significantly higher survive rate than male passenger, that means 'sex' will be a good feature
- Female passengers from third class have significantly lower survival rate than other classes. Using both features 'sex' and 'pclass' might really help the model.



- 'Boat known' has very high correlation with survival rate. Their correlation score is 0.95, that means I could get 95% accuracy just using this feature, using other features we could get more.
- Some passengers share the same ticket number, that might mean they travel together.

< 10 > Bonus: Please also append your code when explore with the data in an Appendix section of your document. We prefer if you use something like Jupyter Notebook (example) to explain it best, but feel free to use other formats. Please also give relevant output that helps you when exploring the dataset (e.g. plotting, score, etc)

Please see AppendixATitanicSurvivalPrediction.ipynb for the code, plots, and score.

References

Titanic datasets has been published for quite awhile and there has been a Kaggle competition for it. I learned from some kernels published in this website. Scikit-learn and its documentation also makes this work possible.

https://www.kaggle.com/arthurtok/introduction-to-ensembling-stacking-in-python

https://www.kaggle.com/headsortails/pytanic

https://www.kaggle.com/ldfreeman3/a-data-science-framework-to-achieve-99-accuracy

http://scikit-learn.org/stable/documentation.html