Titanic - ML Project

Bogdanowicz Michal Kamil, Geraci Luca

May 9, 2020

Abstract

Report of the project for the course of ML 2019/2020

1 Proposition

The proposition is using a set of machine learning methods to predict the people that would survive the Titanic sinking of 15 April 1912.

The methods that have been used are:

1. Logistic Regression TODO
2. Decision trees and Random Forest
3. Neural Networks TODO

2 Data

2.1 Incomplete Data

The Data propsed to the professor was incomplete. It heavily reduced the avialble data to perform the best practices for evaluating a model. The test and training were already separated and ground truth was missing. So the complete dataset has been taken from the internet. In addition the complete list of surviors can be found at the wikipedia page here (not a ML-friendly format). This data sadly is used to cheat on the kaggle competition. Making it a quite infamous one.

2.2 Data Format

- Ticket class
- Survival
- Name
- Sex

- Age in years
- sibsp : number of siblings / spouses aboard the Titanic ¡– This might be problematic as it doesn't seem to make much sense.
- parch : number of parents / children aboard the Titanic¡– This might be problematic as it doesn't seem to make much sense.
- Ticket number
- Fare
- Cabin/s assigned
- Port of Embarkation
- Rescue Boat
- Body
- Destination

With additional notes of : Ticket class: A proxy for socio-economic status (SES) 1st = Upper 2nd = Middle 3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way Parent = mother, father Child = daughter, son, stepdaughter, stepson Some children travelled only with a nanny, therefore parch=0 for them.

2.3 Data Preparation

The data cannot be used as it is. The names are going to be deleted, as that should not inlfuence the resut.

3 Logistic Regression

One important remark. Since a lot of the data is a binary variable (0,1). The square of 0 and 1 comes back to the same value. That means that eleveting features which have only values 0 and 1 adds nothing to the model. Even more, it adds a fully correlated feature. Which has a negative impact on the model (even though it has not been the case empirically for this model. The reasons seems to be that the ensuring overestimation has no effect on the final result. The reasoning used still stands : Avoid correlated features for ML).
This is the procedure that has been used to select the model.

1. Choose the maximum polynomial to use by comparing the measures with CV (Cross Validation)

2. Check for Bias and variance by visualizing the Error in CV and Error on Training set on a graph with X as the degree of polynomial

3. Choose the regularization parameter that reducces the CV Error the most by using the polynomial choseon and Lambda going from 0.01 to 5.

4 Decision trees and random forest

4.1 Introduction to decision trees

A decision tree builds upon iteratively asking questions to partition data. Our aim is to increase the predictiveness of the model as much as possible at each partitioning so that the model keeps gaining information about the dataset. There are two ways to measure the quality of a split:

1. Gini-Index: The measure is about the impurity of a node. The aim is to reduce the impurity (reduce randomness) of data to achieve correct classification (labeling).

2. Information Gain (Entropy): The information gain is the difference between entropy before and after the split. When splitting decision trees try to be more predictive, less impure and reduce entropy. Entropy is a measure of uncertainty or randomness. The more randomness a variable (feature) has, the higher the entropy is.

4.2 Introduction to random forests

Random forest is an ensemble of many decision trees. Random forests are built using a method called bagging in which each decision tree is used as parallel estimator. Random forests reduce the risk of overfitting and accuracy is much higher than a single decision tree. Furthermore, decision trees in a random forest run in parallel so that the time does not become a bottleneck.

4.3 Procedure

We used the M5PrimeLab Octave toolbox as follows:

1. Normalize data introducing Dummy Variables and transform feature in numerical data

2. Build and plot the decision tree to inspect generated number of rules

3. Use 10-fold Cross Validation to calculate the key indicators MAE, MSE, RMSE, RRMSE, R2, nRules, nVars

4. Build and plot the precision tree to inspect generated number of rules

5. Use 10-fold Cross Validation on precision tree to calculate the key indicators MAE, MSE, RMSE, RRMSE, R2, nRules, nVars

6. Calculate prediction, training set mean and input variable contributions

7. Extract the decision rules from the tree

8. Build tree forest (ensemble)

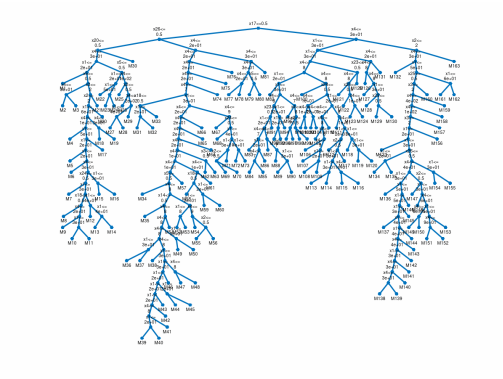
9. Calculate the out-of-bag Mean Squared Error (MSE)

10. Plot the variable importance

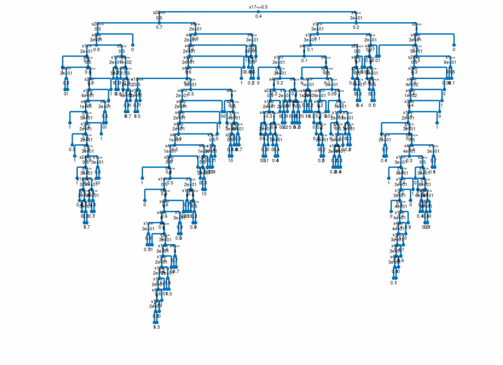
11. Predict the forest

12. Use 10-fold Cross Validation to evaluate the predictive performance of the forest

4.4 Results



(a) decision tree



(b) precision tree

4.5 Lesson learned

4.5.1 Advantages of Decision Trees

1. No normalization or scaling of features
2. Suitable for mixed feature data types
3. Easy results interpretation

4.5.2 Disadvantages of Random Trees

1. Prone to overfit and need to build forests to get good results

4.5.3 Advantages of Random Forests

1. Powerful, highly accurate model on many different problems
2. No normalization or scaling of features
3. Suitable for mixed feature data types
4. Parallel computation for stable performance

4.5.4 Disadvantages of Random Forests

1. Not a good choice for high-dimensional data

5 Conclusion

Write your conclusion here.