

PROJECT

Titanic Survival Exploration

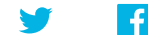
A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT



Congratulations for meeting all specifications from the zeroth project! :) I added an extended comment on **overfitting**, one of the most important topics you'll study in this nanodegree, and a brief comment on your last answer. Keep up the good work!

Answers to Each Question



The `predictions_0` function has been run and the accuracy of the predictions is reported.



The `predictions_1` function has been correctly implemented. The expected accuracy of the predictions is reported.



The `predictions_2` function has been correctly implemented. The expected accuracy of the predictions is reported.

Awesome

All predictions are correctly implemented, nice going!

Suggestion

You can get rid of the elif statement by writing:

```
if (passenger['Sex'] == 'female') or (passenger['Age'] < 10):  
    predictions.append(1)
```



The `predictions_3` function has been correctly implemented and obtains a prediction accuracy of at least 80%. The approach to the task has been documented, including features that were explored and intermediate steps taken to complete the function.

Awesome

Nice explanation of the reasoning behind your code, great job!

Comments

I don't know if you've heard of [overfitting](#) - if you haven't, you will: it comes up a lot in this ND. The idea is that you should worry if your model is too good, because it may mean the model has simply learned to "mimic" the training data but will be useless once new data comes in and it needs to make a prediction.

For an extreme example, suppose you define a model in which the survival rate is estimated using the name of the passenger: if the name is on the "survived" list, the passenger survived; if it isn't, they didn't. This would be a completely useless model: what would you do to predict the fate of a passenger whose name is not on either list? However, it would be 100% accurate if we consider only the data we had when we built it!

One of the lessons of overfitting is that, whenever possible, it's a good idea to justify the model you're using. If you add a variable, for instance, make sure it's a reasonable one given the problem at hand! Otherwise, you're just helping the model overfit.

In a way, this P0 is all about overfitting: just add any condition to help you cross the 80% threshold, no matter what. And a good part of the rest of this ND will be about how to avoid overfitting and make sure your model can generalize to unseen data.



A valid scenario where supervised learning can be applied is reported. A clear outcome variable and at least two potential predictor variables are identified as part of the description.

Comment

Note that "email content" is not as simple a variable as it may seem! There are many ways of representing this content - one of which you will learn in the Naïve Bayes video lessons - and most of them involve creating many more than one features. In any event, you present more than 2 features anyway, so you more than meet specifications here. :)

 [DOWNLOAD PROJECT](#)

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

[RETURN TO PATH](#)

[Student FAQ](#)