Source:

#ref #ret

---

1 | Corrections!

1.1 | Validation

org: [[KBValidationPB]]

```
Problem 2c. This is the lowest possible score, but is it the worst score? Consider: let's say you wantec
Wesley Chao, Oct 23 at 5:26pm
Problems 5 & 6: how do you know the probability is 100%? Is it necessary to return a probability of 100%
```

For problem 2c, the least helpful score would be 0.5, in which every sample is sorted randomly. This would give virtually no information, and is just as good as random.

The 'worst' possible score, however, would still be 0. This is just a definitions games, as, of course, you could simply flip the results around and get a perfect score.

In actuality we don't really know the probability that a perfect model will return. Thus, all we can say is that the probability will be >= 50%.

1.2 | Classification

```
Excellent work, Huxley! You only had to do exercises in two of the four notebooks, but you not only did

Decision Tree: good work on the standard exercise; I agreed with your answers. A question on #5: what ha

On the advanced exercise, good work figuring out how to implement a random forest classifier and calcula

Logistic Regression, generated data: Very nice work tackling both advanced exercises! Your plot of the r

Logistic Regression, iris data: Great work here as well (although, as I'm sure you noticed, the exercise

Naive Bayes: I agree that words like "mars" and "shaders" which show up exclusively in one category modi

In the advanced exercise, you wrote efficient, technically solid, well-documented code… but you didn't a
```

Hi Wes,

Each of these notebooks do introduce new concepts to me, so I am definitely still learning. They just go by pretty quickly. However, I would love to work on some more projects, especially once we get to the more advanced topics!

#5: Using sepal length and sepal width, the structure is significantly more complicated. See attached image.

As for the random forest and decision tree, if I were to compare them, I would compare their F1 scores on test data.

For Naive Bayes, less obvious words which sway the prediction immensely include: 'on', 'that', and 'of'.

The words 'on', 'that', and 'of' appear much more in sci.space than the other categories, and tend to sway the prediction towards sci.space greatly. This does not match my intuition of how the model should work.

Thanks for the feedback!

1.3 | **Linear Regression**

```
General note: when you submit via a git repo, please tell me what I should be looking at, even if it sh
```

```
Exercise 1-2. These would be the numbers you expect if the model found weights exactly corresponding to
```

```
Nice work taking on the advanced exercises! For exercise 3, you used support vector machine-based regre
```

```
For exercise 4, you implemented and plotted a regularized best-fit line using ridge regression. But thi
```

Sorry for the confusion! In the future, I'll submit the direct github link to the file, or add a comment with all the direct links.

Exercises 1-2: I would expect the model to predict numbers according to the intercept and coefficient that it did find, which, "if the linear regression code was working," should be close the to function that the model is trying to fit, right? I think I may be misunderstanding your question.

I don't have a lot of logic behind why I chose to use SVM besides the fact that it looked interesting and I wanted to figure out how to use it. I believe that polynomial regression is more prone to overfitting, but I'm not entirely sure on that.

More broadly speaking, to know if my regularization code worked, I could look for an increase in R^2. This is because a completely over-fitted model would, by definition, have an R^2 of 0. Generalizing, or regularizing the model would make the model less specific to the training data, increasing the R^2.