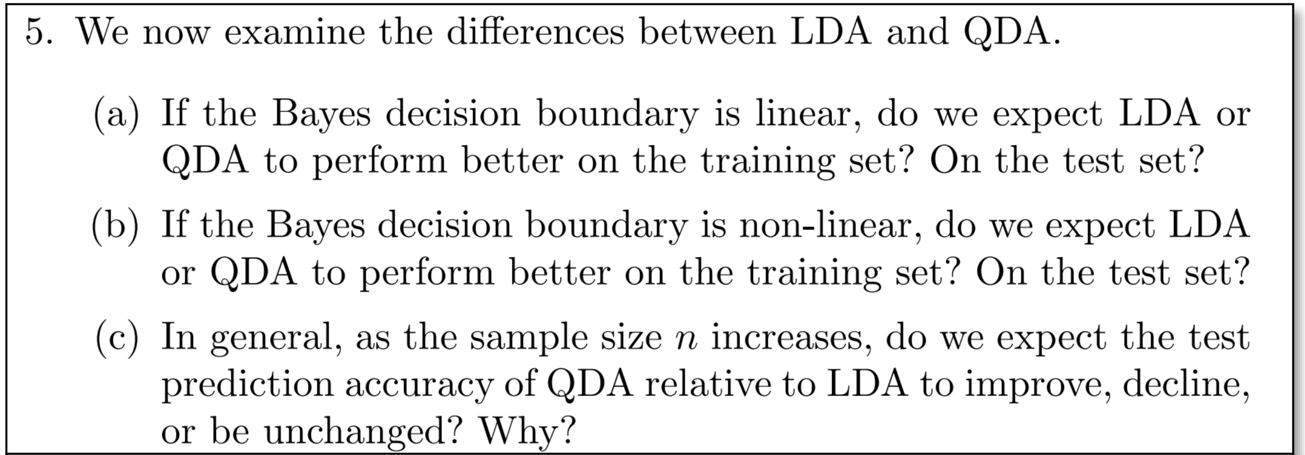
Classification Exercises 1

Microsoft

Conceptual: 4.5

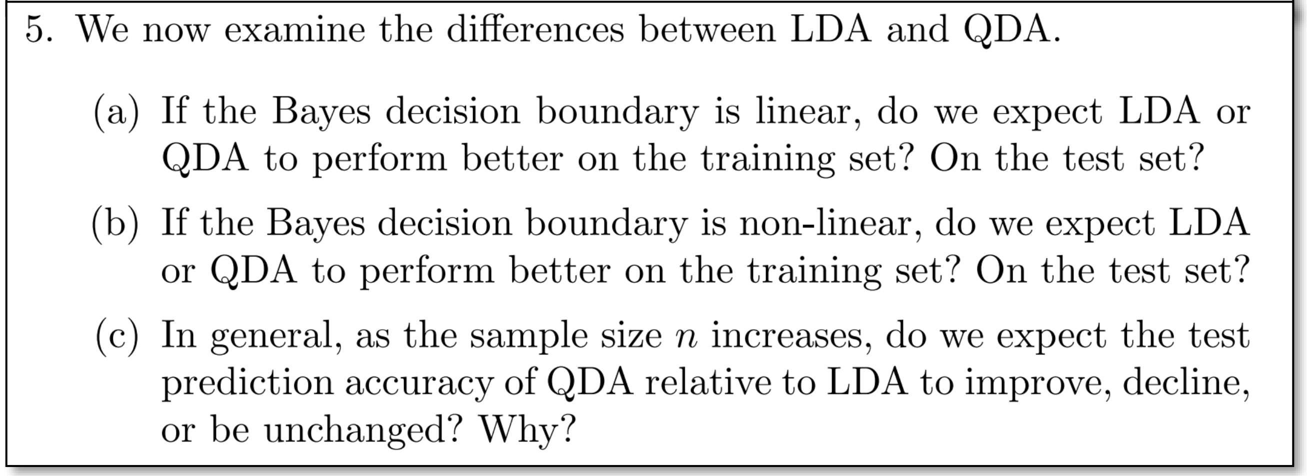


5. We now examine the differences between LDA and QDA.

   (a) If the Bayes decision boundary is linear, do we expect LDA or QDA to perform better on the training set? On the test set?

   (b) If the Bayes decision boundary is non-linear, do we expect LDA or QDA to perform better on the training set? On the test set?

   (c) In general, as the sample size n increases, do we expect the test prediction accuracy of QDA relative to LDA to improve, decline, or be unchanged? Why?



5. We now examine the differences between LDA and QDA.

   (a) If the Bayes decision boundary is linear, do we expect LDA or QDA to perform better on the training set? On the test set?

   (b) If the Bayes decision boundary is non-linear, do we expect LDA or QDA to perform better on the training set? On the test set?

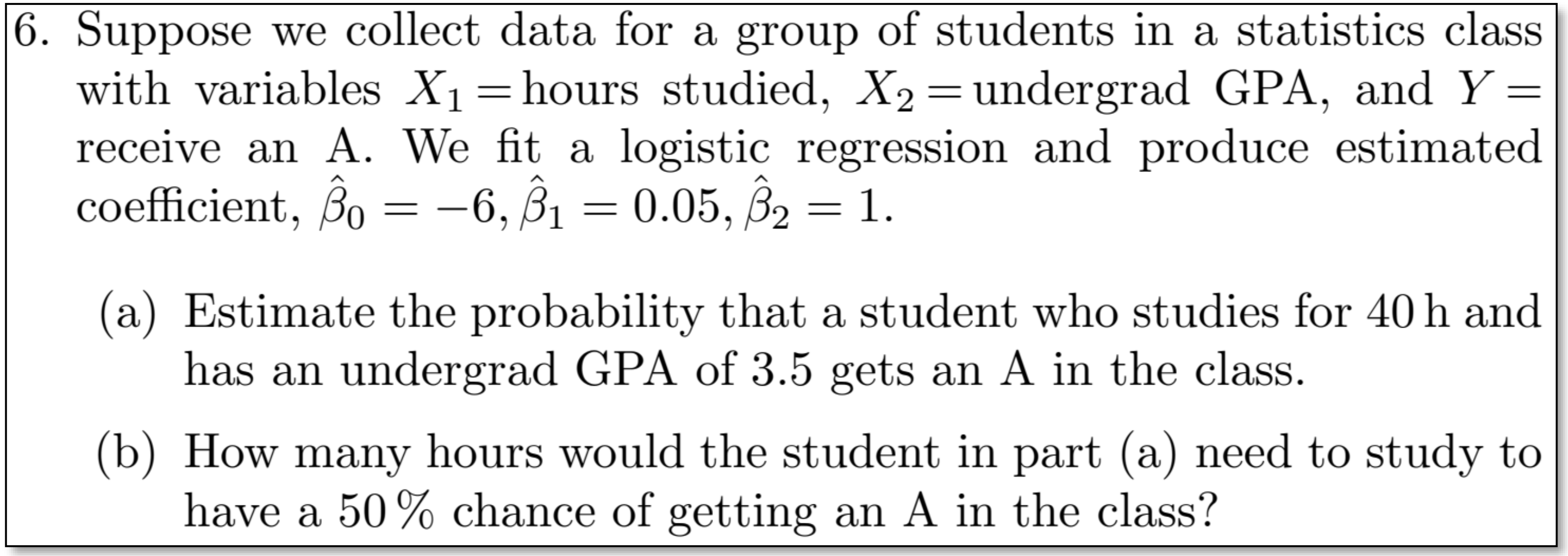
   (c) In general, as the sample size n increases, do we expect the test prediction accuracy of QDA relative to LDA to improve, decline, or be unchanged? Why?

https://botlnec.github.io/islp/

https://botlnec.github.io/islp/sols/chapter4/exercise5/

http://faculty.marshall.usc.edu/gareth-james/ISL/

Conceptual: 4.6



6. Suppose we collect data for a group of students in a statistics class with variables $X_1 =$ hours studied, $X_2 =$ undergrad GPA, and $Y =$ receive an A. We fit a logistic regression and produce estimated coefficient, $\hat{\beta}_0 = -6, \hat{\beta}_1 = 0.05, \hat{\beta}_2 = 1$.

   (a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

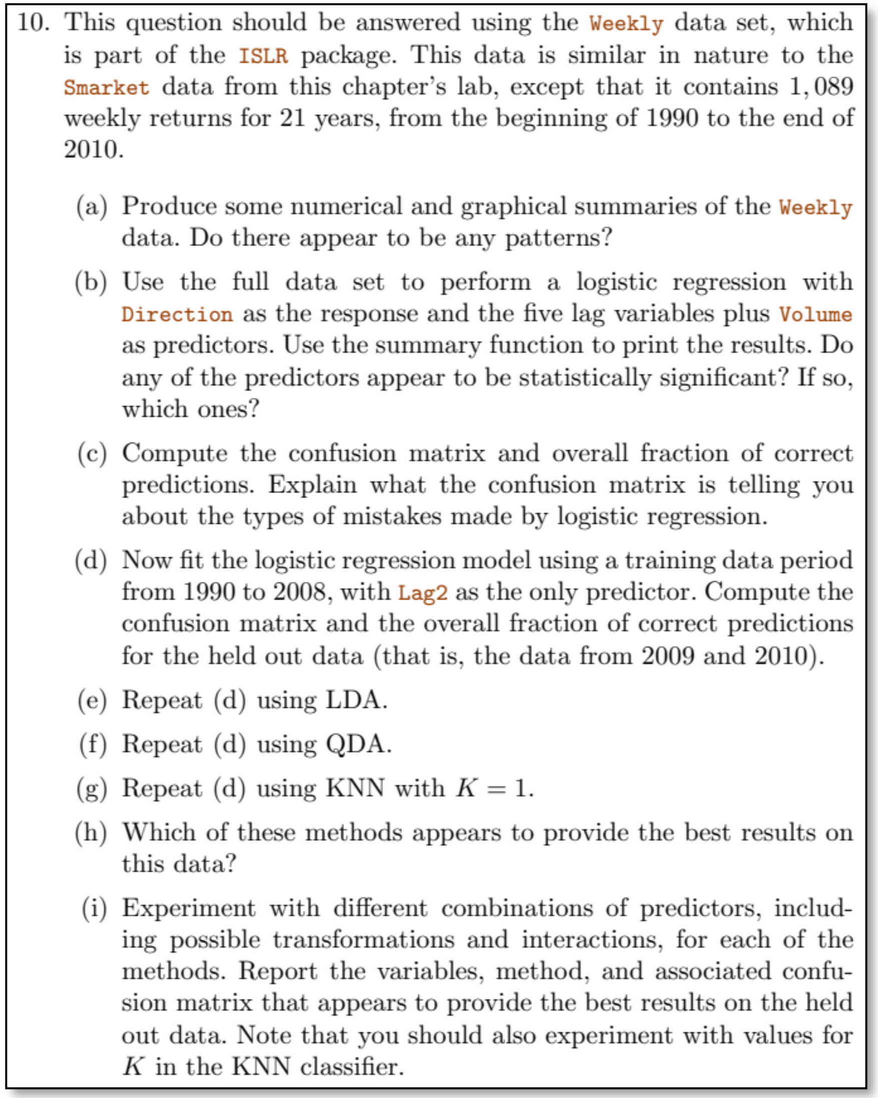
   (b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

https://botlnec.github.io/islp/

https://botlnec.github.io/islp/sols/chapter4/exercise6/

http://faculty.marshall.usc.edu/gareth-james/ISL/

Applied: 4.10



10. This question should be answered using the `Weekly` data set, which is part of the `ISLR` package. This data is similar in nature to the `Smarket` data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

    (a) Produce some numerical and graphical summaries of the `Weekly` data. Do there appear to be any patterns?

    (b) Use the full data set to perform a logistic regression with `Direction` as the response and the five lag variables plus `Volume` as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

    (c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

    (d) Now fit the logistic regression model using a training data period from 1990 to 2008, with `Lag2` as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

    (e) Repeat (d) using LDA.

    (f) Repeat (d) using QDA.

    (g) Repeat (d) using KNN with $K = 1$.

    (h) Which of these methods appears to provide the best results on this data?

    (i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

https://botlnec.github.io/islp/

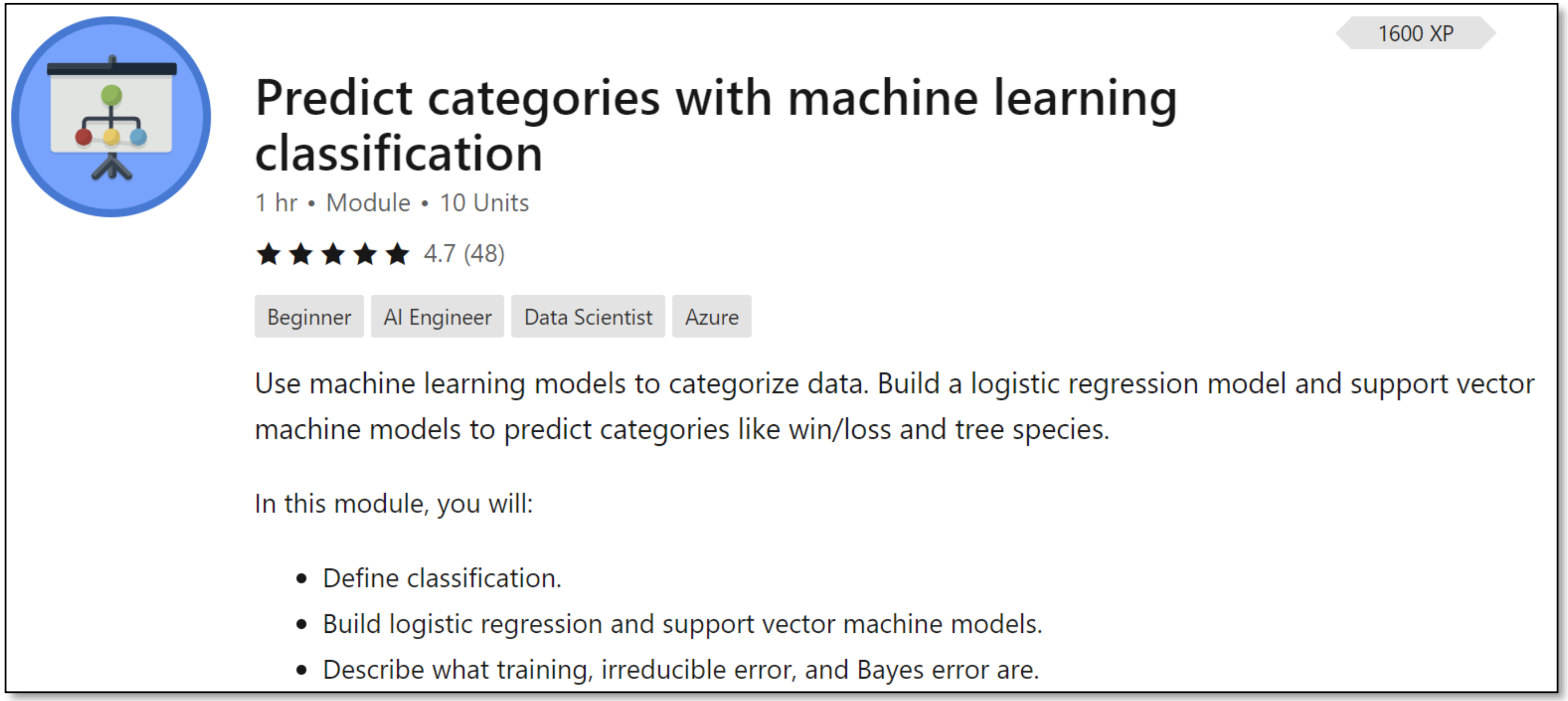
https://botlnec.github.io/islp/sols/chapter4/exercise10/

http://faculty.marshall.usc.edu/gareth-james/ISL/

Classification Exercises 2

Microsoft

Predict Categories



1600 XP

Predict categories with machine learning classification

1 hr • Module • 10 Units

★★★★★ 4.7 (48)

Beginner | AI Engineer | Data Scientist | Azure

Use machine learning models to categorize data. Build a logistic regression model and support vector machine models to predict categories like win/loss and tree species.

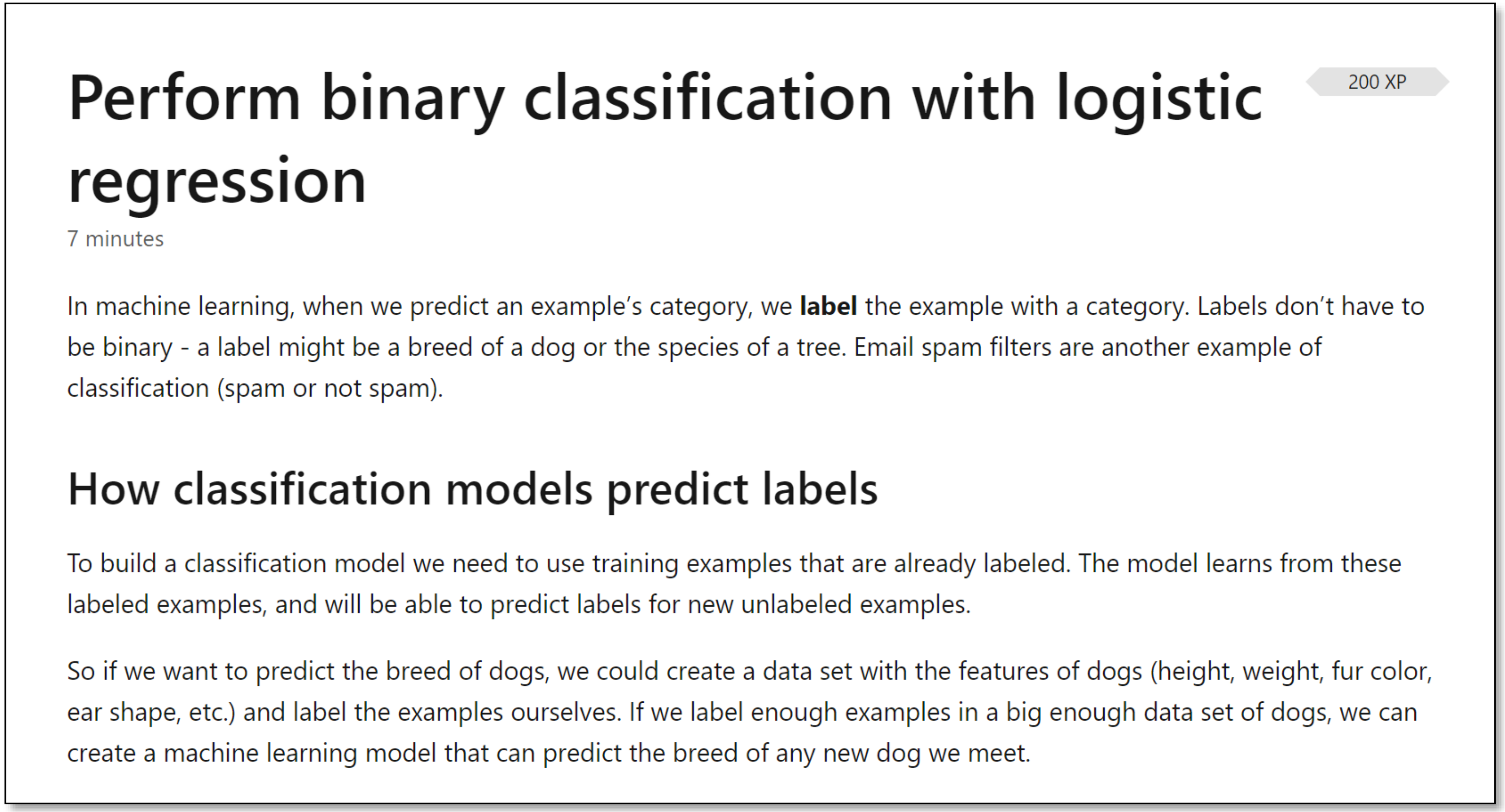
In this module, you will:

- Define classification.
- Build logistic regression and support vector machine models.
- Describe what training, irreducible error, and Bayes error are.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/index

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Perform binary classification with logistic regression

7 minutes

In machine learning, when we predict an example's category, we **label** the example with a category. Labels don't have to be binary - a label might be a breed of a dog or the species of a tree. Email spam filters are another example of classification (spam or not spam).

How classification models predict labels

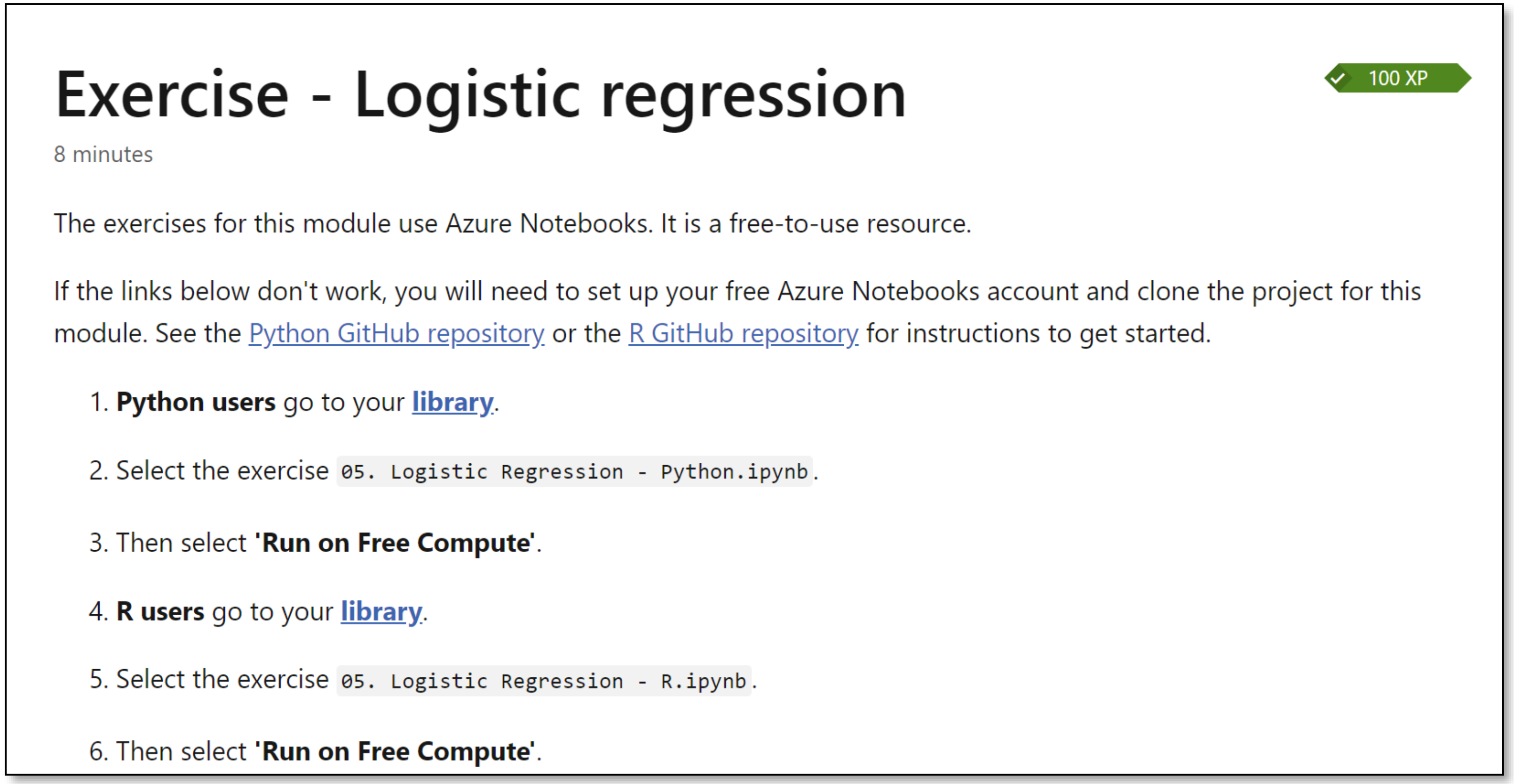
To build a classification model we need to use training examples that are already labeled. The model learns from these labeled examples, and will be able to predict labels for new unlabeled examples.

So if we want to predict the breed of dogs, we could create a data set with the features of dogs (height, weight, fur color, ear shape, etc.) and label the examples ourselves. If we label enough examples in a big enough data set of dogs, we can create a machine learning model that can predict the breed of any new dog we meet.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/2-logistic-regression

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Exercise - Logistic regression

8 minutes

The exercises for this module use Azure Notebooks. It is a free-to-use resource.

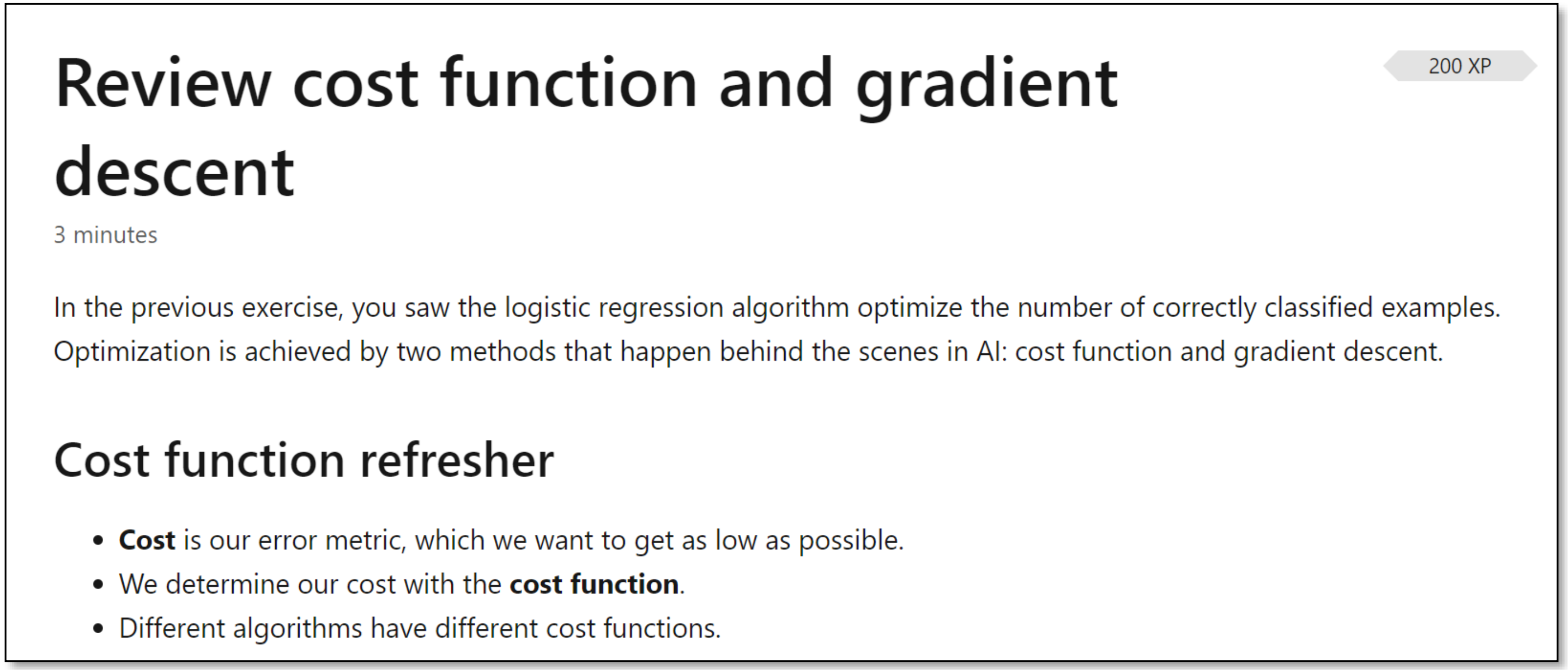
If the links below don't work, you will need to set up your free Azure Notebooks account and clone the project for this module. See the Python GitHub repository or the R GitHub repository for instructions to get started.

1. **Python users** go to your **library**.
2. Select the exercise `05. Logistic Regression - Python.ipynb`.
3. Then select **'Run on Free Compute'**.
4. **R users** go to your **library**.
5. Select the exercise `05. Logistic Regression - R.ipynb`.
6. Then select **'Run on Free Compute'**.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/3-exercise-logistic-regression

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Review cost function and gradient descent

200 XP

3 minutes

In the previous exercise, you saw the logistic regression algorithm optimize the number of correctly classified examples. Optimization is achieved by two methods that happen behind the scenes in AI: cost function and gradient descent.

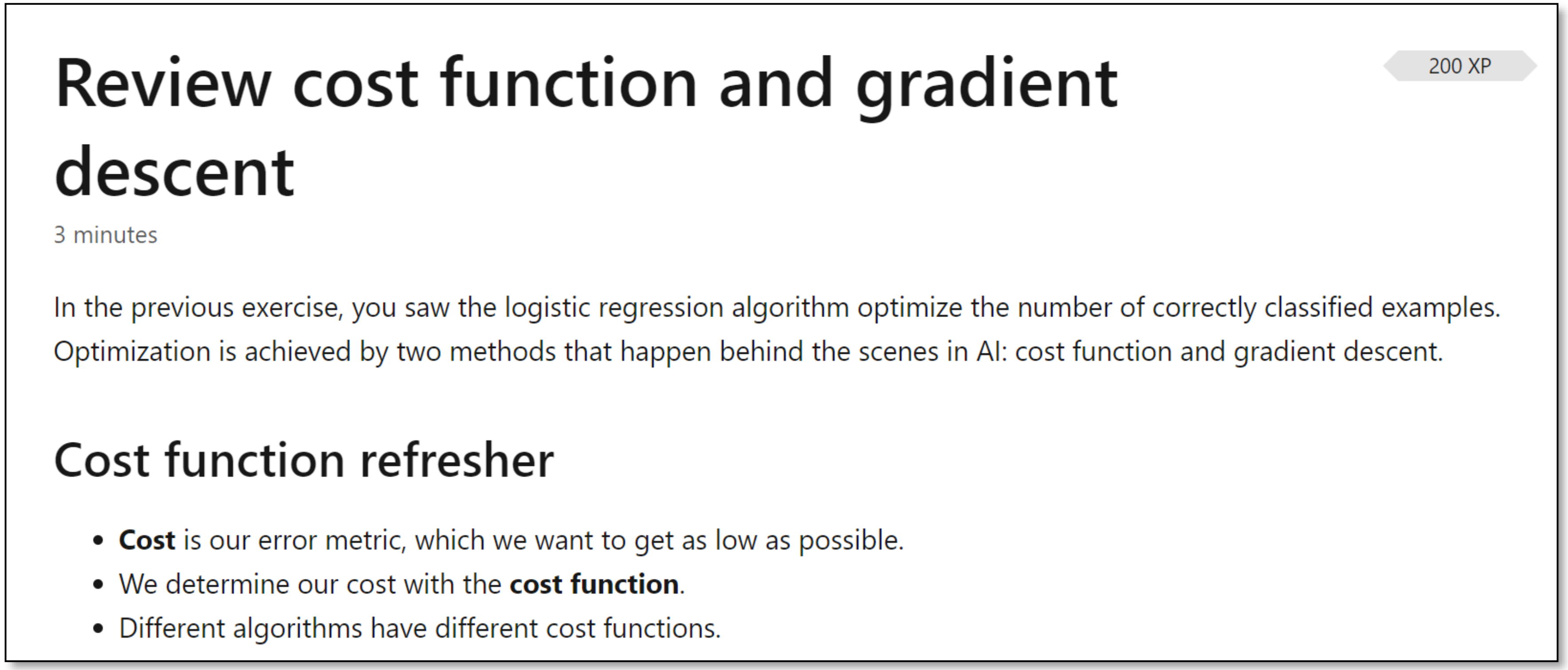
Cost function refresher

- **Cost** is our error metric, which we want to get as low as possible.
- We determine our cost with the **cost function**.
- Different algorithms have different cost functions.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/4-cost-function-gradient-descent

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Review cost function and gradient descent

200 XP

3 minutes

In the previous exercise, you saw the logistic regression algorithm optimize the number of correctly classified examples. Optimization is achieved by two methods that happen behind the scenes in AI: cost function and gradient descent.

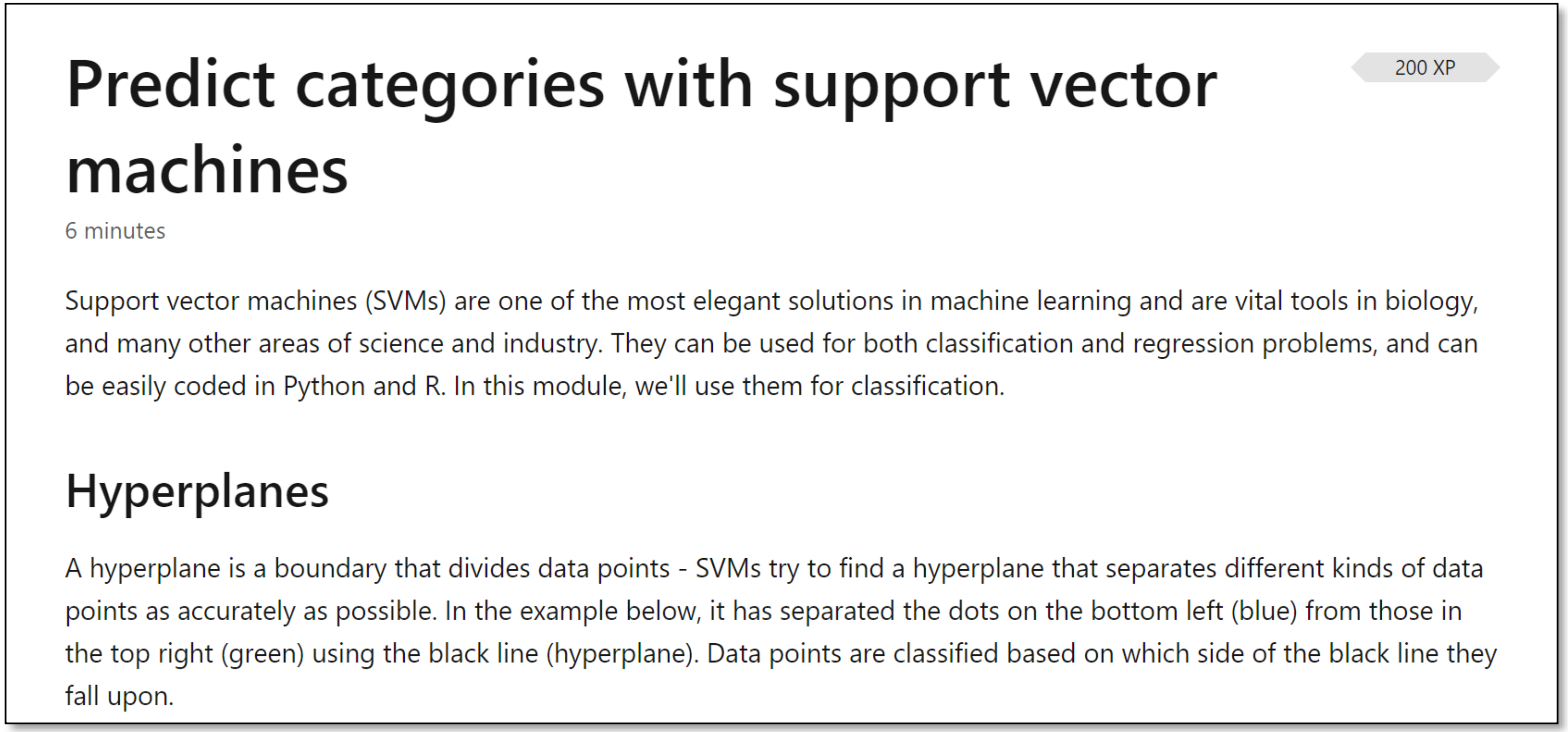
Cost function refresher

- **Cost** is our error metric, which we want to get as low as possible.
- We determine our cost with the **cost function**.
- Different algorithms have different cost functions.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/4-cost-function-gradient-descent

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Predict categories with support vector machines

6 minutes

Support vector machines (SVMs) are one of the most elegant solutions in machine learning and are vital tools in biology, and many other areas of science and industry. They can be used for both classification and regression problems, and can be easily coded in Python and R. In this module, we'll use them for classification.

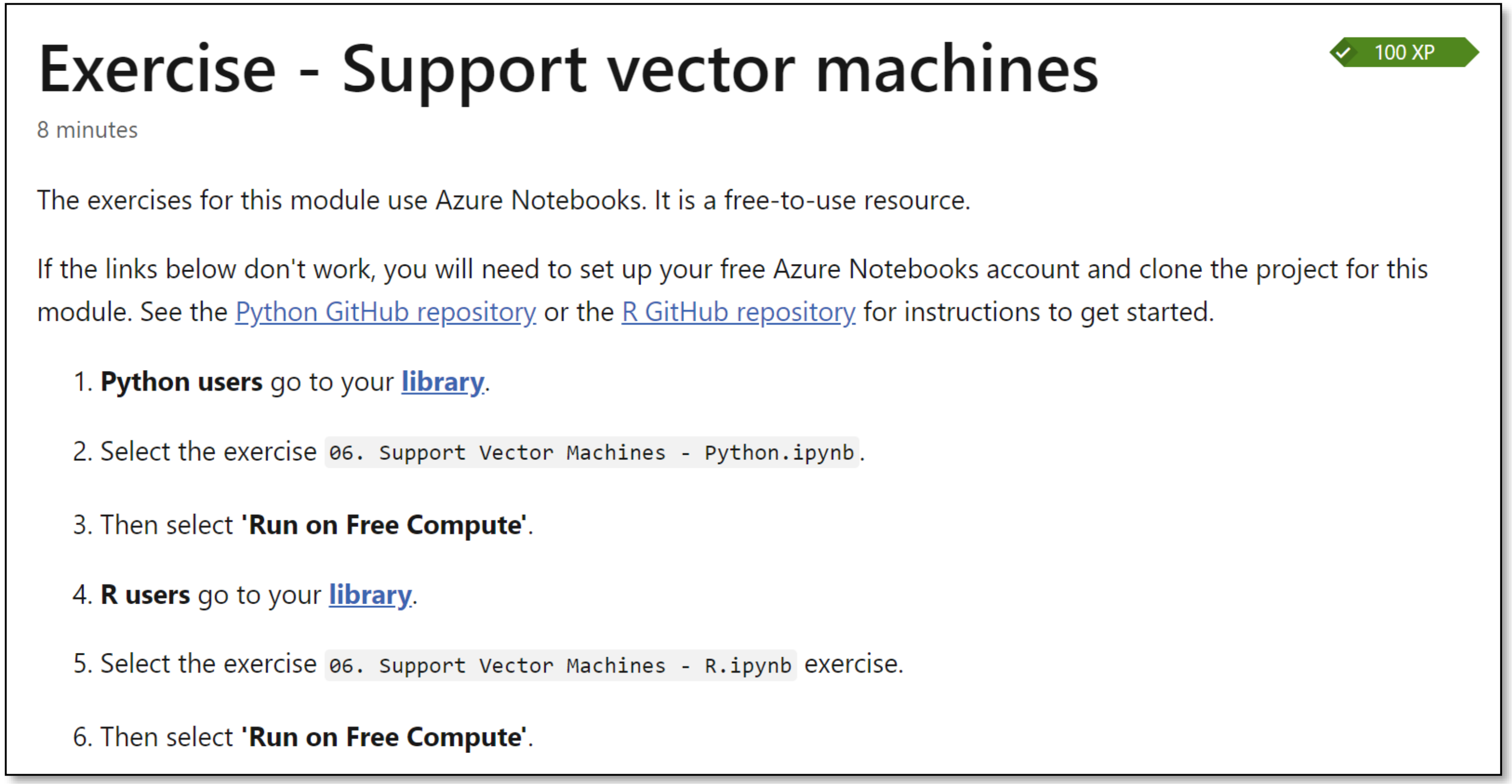
Hyperplanes

A hyperplane is a boundary that divides data points - SVMs try to find a hyperplane that separates different kinds of data points as accurately as possible. In the example below, it has separated the dots on the bottom left (blue) from those in the top right (green) using the black line (hyperplane). Data points are classified based on which side of the black line they fall upon.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/5-support-vector-machines

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Exercise - Support vector machines

8 minutes

The exercises for this module use Azure Notebooks. It is a free-to-use resource.

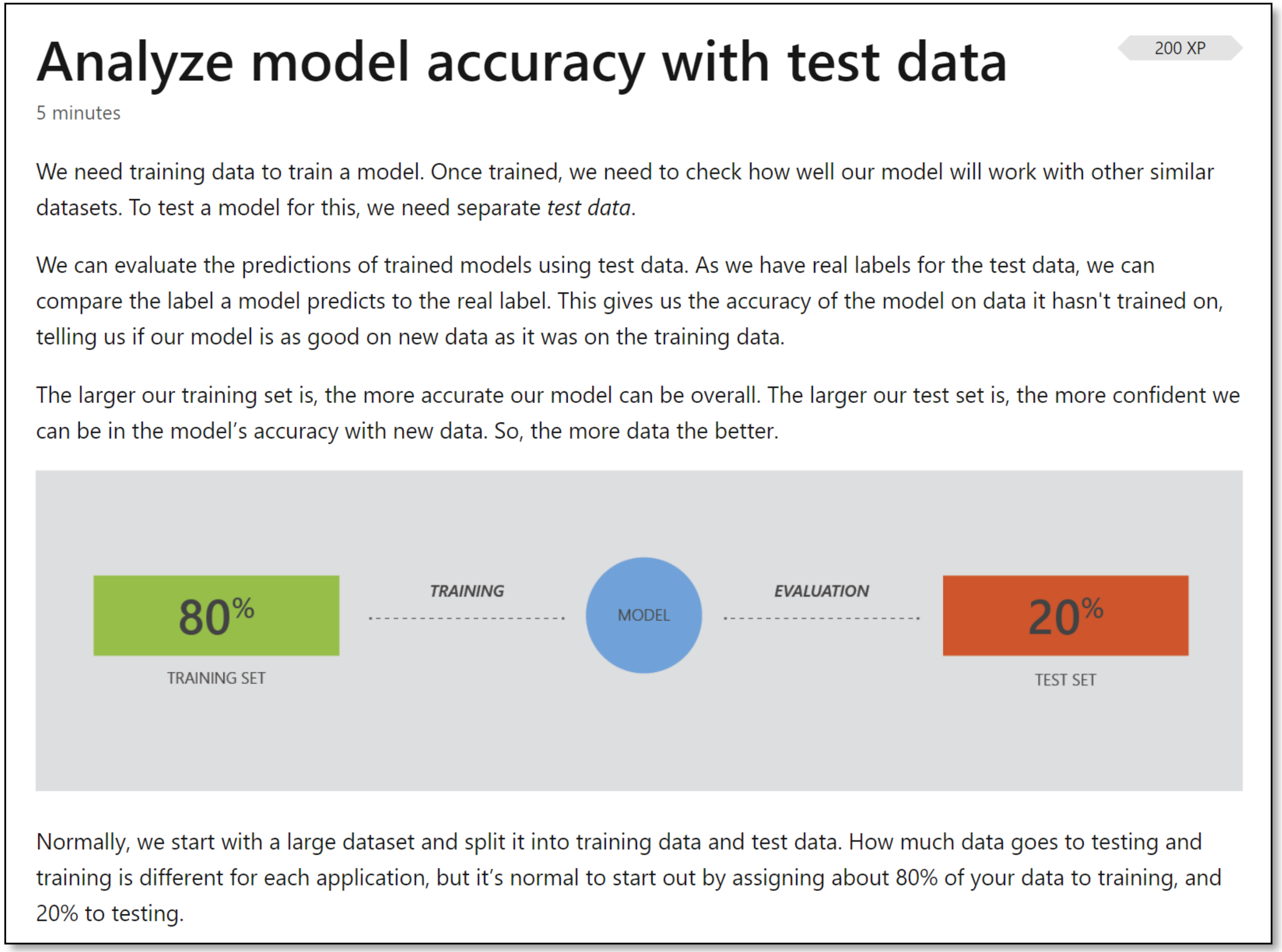
If the links below don't work, you will need to set up your free Azure Notebooks account and clone the project for this module. See the Python GitHub repository or the R GitHub repository for instructions to get started.

1. **Python users** go to your **library**.
2. Select the exercise `06. Support Vector Machines - Python.ipynb`.
3. Then select **'Run on Free Compute'**.
4. **R users** go to your **library**.
5. Select the exercise `06. Support Vector Machines - R.ipynb` exercise.
6. Then select **'Run on Free Compute'**.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/6-exercise-svms

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Analyze model accuracy with test data

200 XP

5 minutes

We need training data to train a model. Once trained, we need to check how well our model will work with other similar datasets. To test a model for this, we need separate *test data*.

We can evaluate the predictions of trained models using test data. As we have real labels for the test data, we can compare the label a model predicts to the real label. This gives us the accuracy of the model on data it hasn't trained on, telling us if our model is as good on new data as it was on the training data.

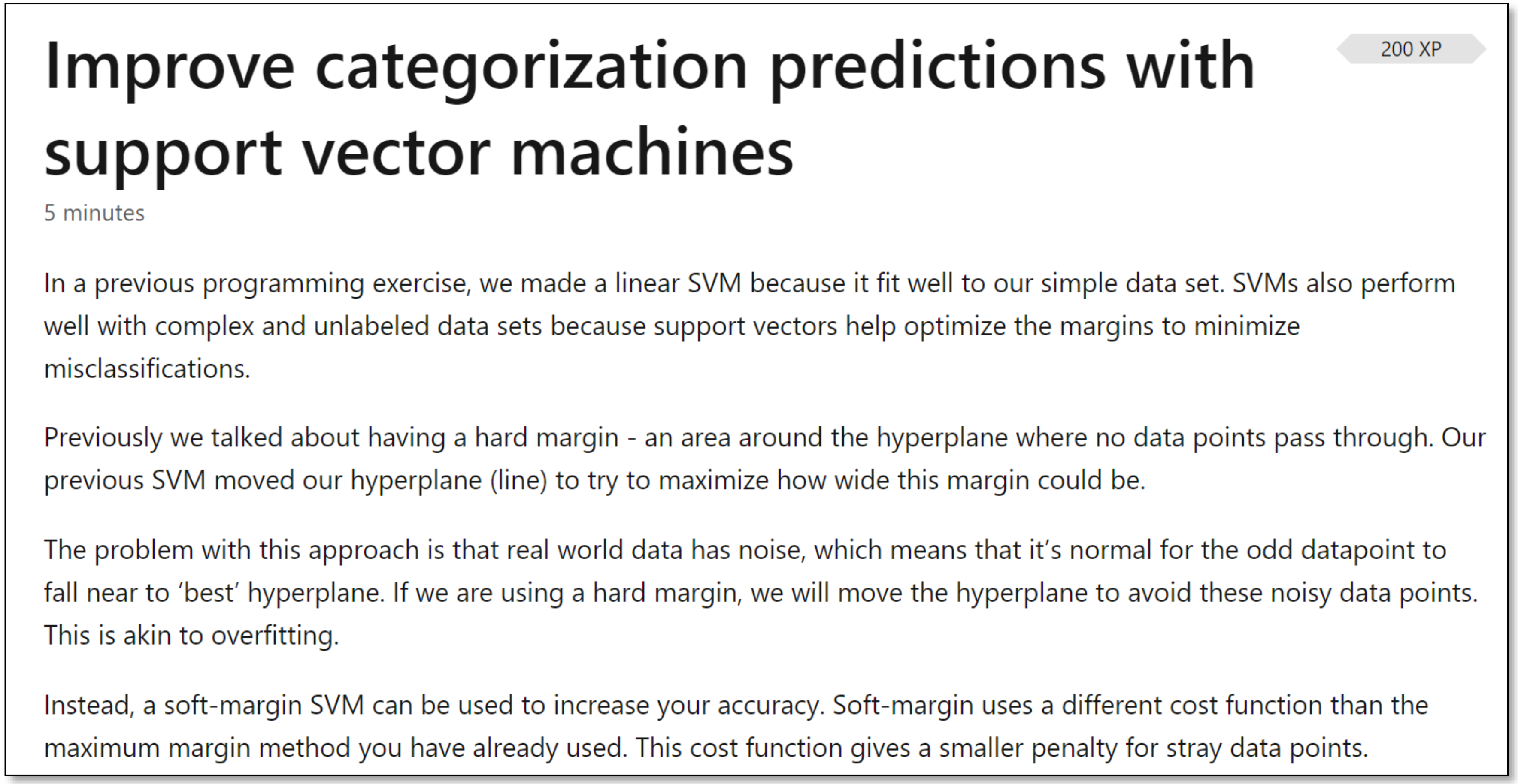
The larger our training set is, the more accurate our model can be overall. The larger our test set is, the more confident we can be in the model's accuracy with new data. So, the more data the better.

Normally, we start with a large dataset and split it into training data and test data. How much data goes to testing and training is different for each application, but it's normal to start out by assigning about 80% of your data to training, and 20% to testing.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/7-testing-your-predictions

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Improve categorization predictions with support vector machines

200 XP

5 minutes

In a previous programming exercise, we made a linear SVM because it fit well to our simple data set. SVMs also perform well with complex and unlabeled data sets because support vectors help optimize the margins to minimize misclassifications.

Previously we talked about having a hard margin - an area around the hyperplane where no data points pass through. Our previous SVM moved our hyperplane (line) to try to maximize how wide this margin could be.

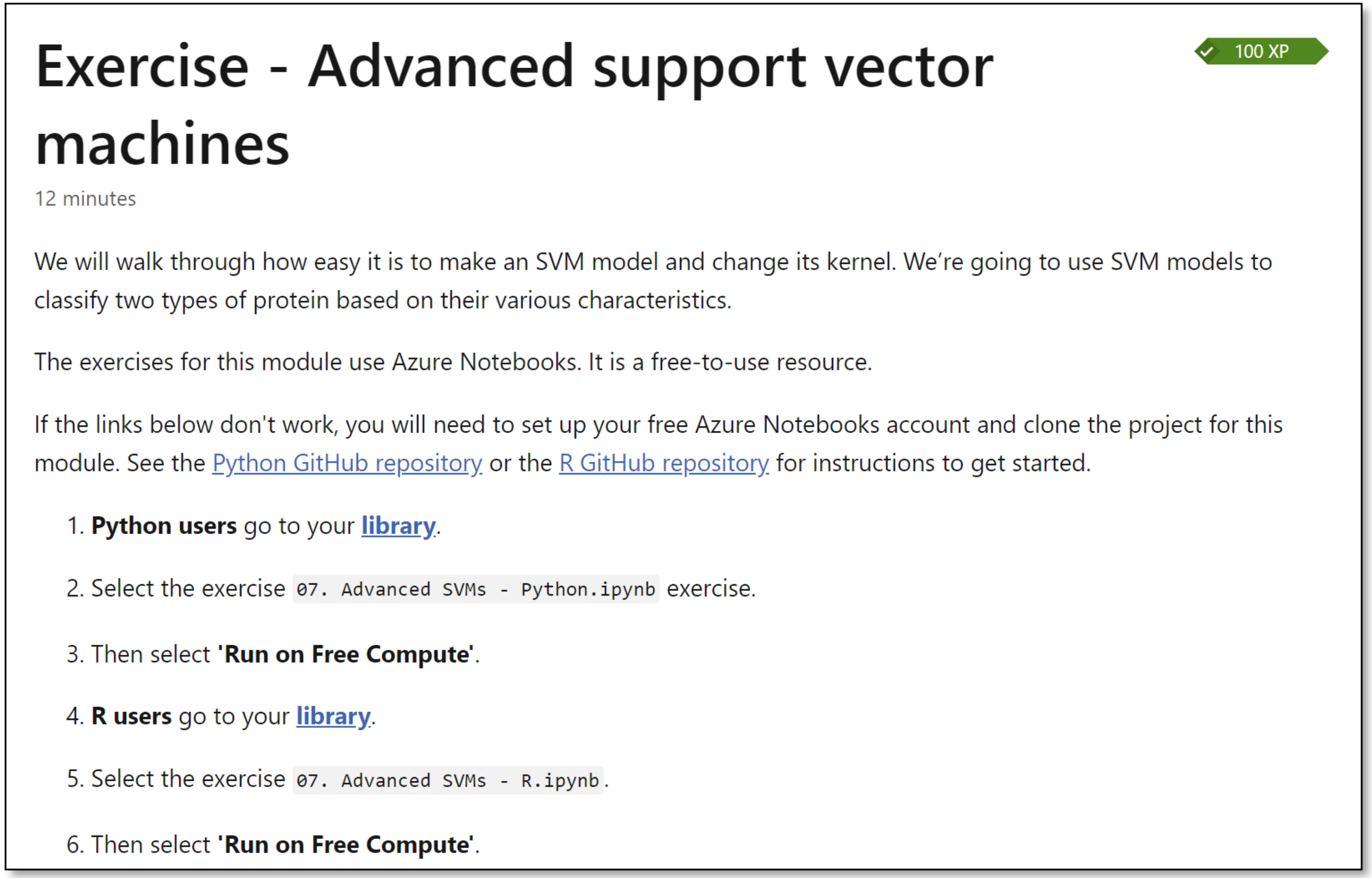
The problem with this approach is that real world data has noise, which means that it's normal for the odd datapoint to fall near to 'best' hyperplane. If we are using a hard margin, we will move the hyperplane to avoid these noisy data points. This is akin to overfitting.

Instead, a soft-margin SVM can be used to increase your accuracy. Soft-margin uses a different cost function than the maximum margin method you have already used. This cost function gives a smaller penalty for stray data points.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/8-advanced-classification-with-svms

http://faculty.marshall.usc.edu/gareth-james/ISL/

Predict Categories



Exercise - Advanced support vector machines

12 minutes

We will walk through how easy it is to make an SVM model and change its kernel. We’re going to use SVM models to classify two types of protein based on their various characteristics.

The exercises for this module use Azure Notebooks. It is a free-to-use resource.

If the links below don't work, you will need to set up your free Azure Notebooks account and clone the project for this module. See the Python GitHub repository or the R GitHub repository for instructions to get started.

1. **Python users** go to your **library**.
2. Select the exercise `07. Advanced SVMs - Python.ipynb` exercise.
3. Then select **'Run on Free Compute'**.
4. **R users** go to your **library**.
5. Select the exercise `07. Advanced SVMs - R.ipynb`.
6. Then select **'Run on Free Compute'**.

https://docs.microsoft.com/en-us/learn/modules/introduction-to-classification/9-exercise-svms

http://faculty.marshall.usc.edu/gareth-james/ISL/