Chapter 1

1. Can you tell the difference between machine learning and traditional programming (rule-based automation)?

In traditional programming, the computer follows a set of predefined rules to process the input data and produce the outcome. In machine learning, the computer tries to mimic human thinking. It interacts with the input data, expected outcome, and environment, and it derives patterns that are represented by one or more mathematical models. The models are then used to interact with future input data and to generate outcomes. Unlike in automation, the computer in a machine learning setting doesn't receive explicit and instructive coding.

2. What's overfitting and how do we avoid it?

Reaching the right fit model is the goal of a machine learning task. What if the model overfits? Overfitting means a model fits the existing observations too well but fails to predict future new observations.

- cross validation

  When the training size is very large, it's often sufficient to split it into training, validation, and testing (three subsets) and conduct a performance check on the latter two. Cross-validation is less preferable in this case since it's computationally costly to train a model for each single round. But if you can afford it, there's no reason not to use cross-validation. When the size isn't so large, cross-validation is definitely a good choice.

- Regularization

  Another way of preventing overfitting is regularization. Recall that the unnecessary complexity of the model is a source of overfitting. Regularization adds extra parameters to the error function we're trying to minimize, in order to penalize complex models.

- feature selection and dimensionality reduction

  We typically represent data as a grid of numbers (a matrix). Each column represents a variable, which we call a feature in machine learning. In supervised learning, one of the variables is actually not a feature, but the label that we're trying to predict. And in supervised learning, each row is an example that we can use for training or testing.

3. Name two feature engineering approaches.

- Polynomial transformation
- Binning

4. Name two ways to combine multiple models.

- Voting and averaging
- Bagging

Chapter 2

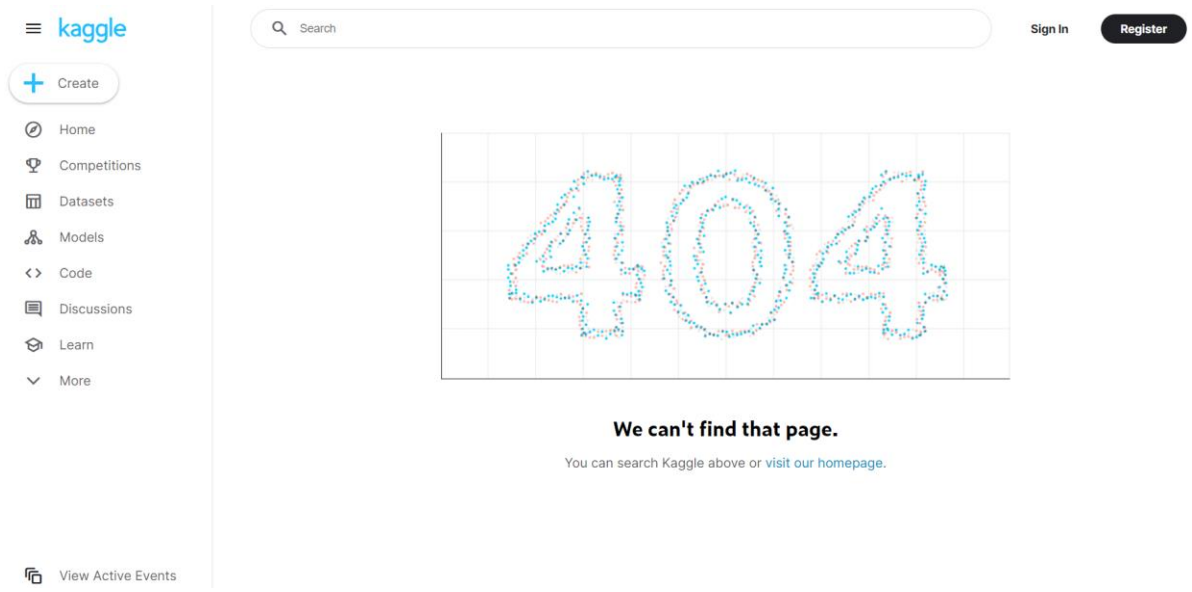
1. **As mentioned earlier, we extracted user-movie relationships only from the movie rating data where most ratings are unknown. Can you also utilize data from the files movies.dat and users.dat?**

Yes, any file with .dat extension can be used for the model.

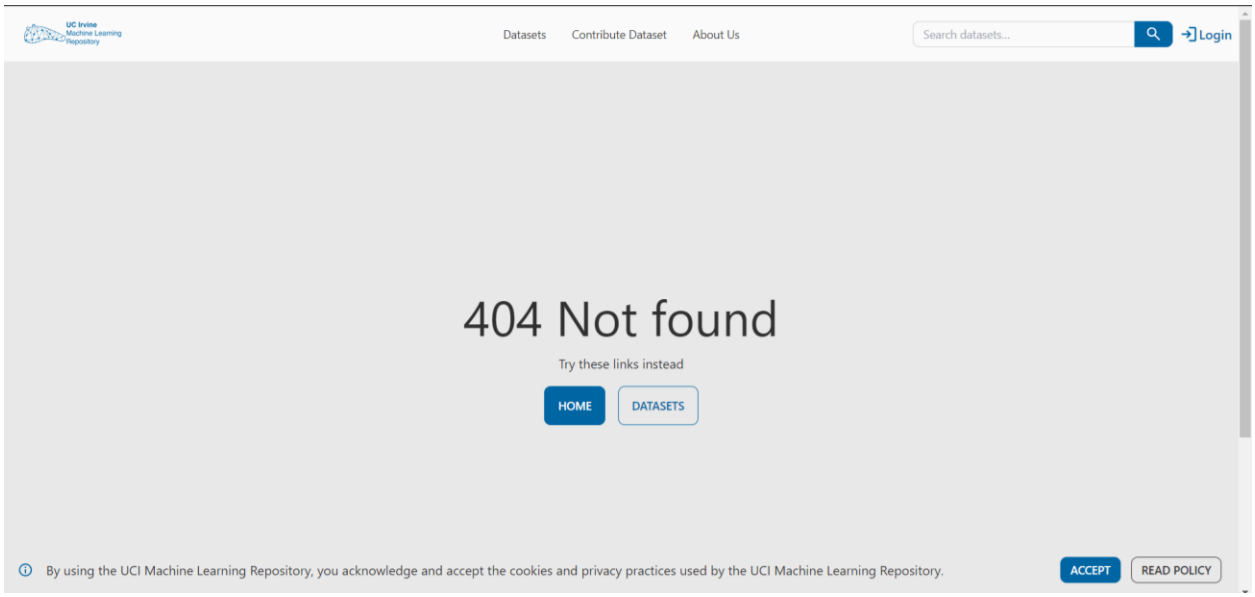
2. Practice makes perfect—another great project to deepen your understanding could be heart disease classification. The dataset can be downloaded directly at https://www.kaggle.com/ronitf/heart-disease-uci, or from the original page at https://archive.ics.uci.edu/ml/datasets/Heart+Disease.

The research dataset (the link of which is included in the book) is not available.

2.1. https://www.kaggle.com/ronitf/heart-disease-uci



2.2. https://archive.ics.uci.edu/ml/datasets/Heart+Disease.



3. Don't forget to fine-tune the model you obtained from Exercise 2 using the techniques you learned in this chapter. What is the best AUC it achieves?

Due to the unavailability of the data set, it is not possible to train the model.