Microsoft

Data Science and Machine Learning Essentials

Module 5 Key Points

Chapter 19: Recommendation Models

Key Points

- You can create recommenders using regression, classification, or clustering models; but a common approach is to create a filter-based recommender that uses matrix factorization. Azure ML supports the **Matchbox Recommender** model, which uses this technique.
- In Azure ML, you can use the **Recommender Split** mode of the **Split** module to prepare data for a recommender.
- After splitting the data, you can use a **Train Matchbox Recommender** module to train a recommender.
- After training the recommender, you can use the **Score Matchbox Recommender** to generate predictions. You can generate the following kinds of prediction:
    - **Item Recommendation**: Predicts recommended items based on a given user.
    - **Related Items**: Predicts recommended items based on a given item.
    - **Rating Prediction**: Predicts ratings for given users and items.
    - **Related Users**: Predicts users based on a given user.
- Use the **Evaluate Recommender** module to evaluate recommender performance. You can evaluate the model based on the following metrics:
    - *Normalized Discounted Cumulative Gain* (NDCG) for item recommendation, related items, and related users.
    - *Mean Absolute Error* (MAE) and *Root Mean Squared Error* (RSME) for rating prediction.

Further Reading

- **Train Matchbox Recommender** module: https://msdn.microsoft.com/en-us/library/azure/dn905987.aspx
- **Split** module: https://msdn.microsoft.com/en-us/library/azure/dn905969.aspx
- **Score Matchbox Recommender** module: https://msdn.microsoft.com/en-us/library/azure/dn905970.aspx
- **Evaluate Recommender** module: https://msdn.microsoft.com/en-us/library/azure/dn905954.aspx

Chapter 20: Introduction to Jupyter Notebooks in Azure ML

Key Points

- Jupyter notebooks provide an interactive environment for exploring data and collaborating with other data scientists.
- Azure ML supports Jupyter notebooks for Python scripts, with support for other languages likely to follow.

Further Reading

- Jupyter Project: http://jupyter.org/
- Jupyter documentation: http://jupyter-notebook.readthedocs.org/latest/
- Introducing Jupyter in Azure ML: http://aka.ms/mlnotebook

Chapter 21: Publishing Azure ML Models

Key Points

- After you have created a model in an Azure ML experiment, you can publish it as a web service. This creates a web service input based on the schema of your initial dataset, and a web service output based on the results from a **Score Model** module.
- Many modules in an Azure ML experiment are automatically converted to transformations in a web service. However, you should consider the modules in your experiment carefully, and remove any that are useful when training a model from existing data, but which might case incorrect results or errors when used in a web service that accepts a single row as input. In particular, you should carefully test any custom R or Python code in your web service before using it in production.
- After publishing a model as a web service, client applications can access it through a RESTful interface (where JavaScript Notation, or *JSON*, documents are exchanged over HTTP). To do this, they need to specify the appropriate endpoint URL for the web service and the secure key required to access it.
- Each web service supports two endpoints:
    - **Request-Response Service**: Use this for synchronous prediction for single or multiple data rows.
    - **Batch Execution Service**: Use this for asynchronous prediction for high-volume batches of data.

Further Reading

- Deploy an Azure Machine Learning Web Service: https://azure.microsoft.com/en-gb/documentation/articles/machine-learning-publish-a-machine-learning-web-service/