Microsoft

Data Science and Machine Learning Essentials

Module 4 Key Points

Chapter 16: Regression Modeling

Key Points

- You can use the **Sweep Parameters** module to train a regression model based on a random selection of parameter settings until the best combination is found.
- You can use the **Cross Validation** module to perform cross-validation of a regression model. This helps identify and eliminate issues caused by disparity between the training data and test data, and ensures a more generalized model. When evaluating the results of cross-validation, a model that will generalize well is indicated by similar metrics across all of the folds with a small range of mean values and a low standard deviation.
- You can evaluate the performance of a regression model by visualizing the residuals (errors) using R or Python and comparing them to the label and features. In a model that is a good fit for the data, and in which the information in the features is being exploited effectively, the residuals should be randomly distributed with no evident structure in relation to features or the label.
- If after evaluating model performance and feature importance for a regression model, you find that you can't improve its accuracy through feature engineering or pruning; you can try a different kind of regression model. For example, if you have built a linear model (such as Bayesian linear regression), and you have identified a number of non-linear relationships between features and the label, you might have more success with a non-linear model (such as Decision forest regression.)

Further Reading

- **Regression Models**: https://msdn.microsoft.com/en-us/library/azure/dn905922.aspx
- **Sweep Parameters**: https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx
- **Cross Validate**: https://msdn.microsoft.com/en-us/library/azure/dn905852.aspx

Chapter 17: Classification Modeling

Key Points

- The process for creating classification models is similar to that for regression models.

- As with regression models, you can use the **Evaluate Model** module to determine model performance, you can use the **Sweep Parameters** module to train a classification module based on a random selection of parameter settings, and you can use the **Cross Validate** module to ensure that the function and coefficients in the model are generalizable.
- The metrics for evaluating classification model performance are based on statistics for *true positive* (TP), *true negative* (TF), *false positive* (FP), and *false negative* (FN) predictions. These metrics include:
    - Accuracy = TP + TN / (TP + TN + FP + FN)
    - Precision (or positive predictive value) = TP / (TP + FP)
    - Recall = TP / (TP + FN)
    - F1 = Precision * Recall / (Precision + Recall)

Further Reading

- Classification Models: https://msdn.microsoft.com/en-us/library/azure/dn905808.aspx

Chapter 18: Unsupervised Learning Models

Key Points

- Most clustering models are based on K-Means clustering algorithms (which iteratively identify a center point for each cluster, and move the points until the clusters consist of tightly grouped entities in widely separated clusters) or hierarchical clustering algorithms, which start by grouping every entity into its own cluster and then combining each cluster with its closest neighbor until the required number of clusters is identified.
- You can create K-Means clusters in Azure ML by using the **K-Means Clustering** module and the **Train Clustering Model** module.
- You can visualize a principal component diagram of the clusters in the **Results dataset** of the **Train Clustering Model** module. More sophisticated visualizations can be created using R or Python.
- You can create hierarchical clustering models with R or Python.

Further Reading

- Clustering: https://msdn.microsoft.com/en-us/library/azure/dn905908.aspx