Feature Preparation, Selection and Engineering: Takeaways

Syntax

- Returning the descriptive statistics of a data frame:

```
df = pd.DataFrame({ 'object': ['a', 'b', 'c'],
               'numeric': [1, 2, 3],
               'categorical': pd.Categorical(['d','e','f'])
                })
df.describe(include='all')
```

- Rescaling data:

```
from sklearn.preprocessing import minmax_scale
columns = ["column one", "column two"]
data[columns] = minmax_scale(data[columns])
```

- Returning a NumPy array of coefficients from a LogisticRegression model:

```
lr = LogisticRegression()
lr.fit(train_X,train_y)
coefficients = lr.coef_
```

- Creating a horizontal bar plot:

```
ordered_feature_importance = feature_importance.abs().sort_values()
ordered_feature_importance.plot.barh()
plt.show()
```

- Creating bins:

```
pd.cut(np.array([1, 7, 5, 4, 6, 3]), 3)
```

- Extracting groups from a match of a regular expression:

```
s = Series(['a1', 'b2', 'c3'])
s.str.extract(r'([ab])(\d)')
```

- Producing a correlation matrix:

```
import seaborn as sns
correlations = train.corr()
sns.heatmap(correlations)
plt.show()
```

- Using recursive feature elimination for feature selection:

```
from sklearn.feature_selection import RFECV
lr = LogisticRegression()
selector = RFECV(lr,cv=10)
selector.fit(all_X,all_y)
optimized_columns = all_X.columns[selector.support_]
```

Concepts

- We can focus on two main areas to boost the accuracy of our predictions:
    - Improving the features we train our model on.
    - Improving the model itself.
- Feature selection involves selecting features that are incorporated into the model. Feature selection is important because it helps to exclude features which are not good predictors or features that are closely related to each other.
- A model that is overfitting fits the training data too closely and is unlikely to predict well on unseen data.
- A model that is well-fit captures the underlying pattern in the data without the detailed noise found in the training set. The key to creating a well-fit model is to select the right balance of features.
- Rescaling transforms features by scaling each feature to a given range.
- Feature engineering is the practice of creating new features from existing data. Feature engineering results in a lot of accuracy boosts.
- A common technique to engineer a feature is called binning. Binning is when you take a continuous feature and separate it out into several ranges to create a categorical feature.
- Collinearity occurs where more than one feature contains data that are similar. If you have some columns that are collinear, you may get great results on your test data set, but then the model performs worse on unseen data.

Resources

- Documentation for cross validation score
- Documentation for recursive feature elimination with cross validation
- Mastering feature engineering