Feature Scaling

Feature scaling is a method used to normalize the range of independent variables or features of data.
The next step is to scale the crude oil prices between (0, 1) to avoid intensive computation. Common methods include **Standardization** and **Normalization**.

Normalization	Standardization
$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$	$x_{stand} = \frac{x - \text{mean}(x)}{\text{Std}(x)}$

LSTM's are sensitive to the scale of the data so we apply MinMax scaler.

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
data_oil=scaler.fit_transform(np.array(data_oil).reshape(-1,1))
```

Feature engineering is the process of creating features based on knowledge about current features and the required task. Feature creation and feature engineering are the most important tasks in machine learning because they impact model performance. Deep learning also impacts model performance, but to a lesser extent.
You can change features or create features from other features. You must clearly understand the data to do this step. Work with a subject matter expert, if necessary. This activity might also require that you find new data sources. Two key activities are feature extraction and capturing feature relationships.
Feature extraction
You might find that columns in your data aren't useful, possibly because they're too granular. For example, a time stamp is unlikely to be useful while the time of day, or day of the week, might be. In text analytics, *feature extraction* is creating vectors from the raw text strings. Feature extraction might involve creating a column that indicates whether particular phrases are mentioned. Or, you can create columns for each word (Bag-of-Words) and their values are a representation of their frequency (term frequency–inverse document frequency - TF or TF-IDF).
Capturing feature relationships
Rather than expect the model to find relationships between two features, the relationships can be explicitly called out. You're then helping your algorithm focus on what you, or the subject matter expert you're working with, know is important. This focus might be the sum, the difference, the product, or the quotient. For example, a machine learning model might not easily find the connection between the longitude and latitude

values of two addresses. However, by providing the distance between the two, you better enable the model to derive patterns.

Other techniques

Many other techniques exist. Because feature engineering is an art in itself, this list isn't exhaustive.

- One-hot-encoding: Categorical integer features are transformed into “one-hot” vectors. In relational terms, this transformation results in extra columns: one column for each distinct category.
- Time-to-Frequency transformation: Time-series, and sometimes also sequence data, is recorded in the time domain and can be transformed into the frequency domain; for example, by using FFT (Fast Fourier Transformation).
- Month-From-Date: Creating an extra feature that contains the month independent from data captures seasonal aspects. Sometimes further discretization into quarters also helps.
- Aggregate-on-Target: Aggregating fields on the target variable or even other fields can improve performance. For example, you can count the number of data points per postal code or take the median of all values by geographical region.