Dimensionality reduction and unsupervised machine learning have practical applications in the analysis of X-ray fluorescence measurements taken from cutting samples. Dimensionality reduction involves reducing the number of random variables considered by extracting a smaller, informative set of features that capture most of the dataset variation. Meanwhile, clustering, an unsupervised machine learning technique, identifies natural groupings within the data without relying on any pre-defined training data. These groups represent samples with similar characteristics and can be interpreted as unique geochemical facies.

To begin, the initial stage involves importing the necessary tools required for implementing this model.

```
import pandas as pd
import numpy as np

# Machine learning libraries
from sklearn.preprocessing import scale
from sklearn.decomposition import FactorAnalysis
from sklearn.cluster import KMeans

# Visualization libraries
import matplotlib.pylab as plt
import seaborn as sns
```

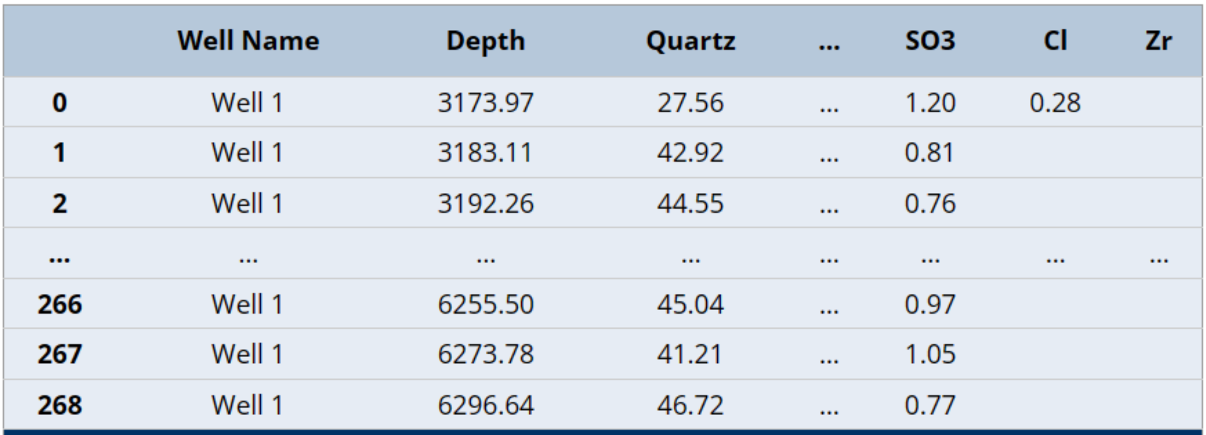
XRF Cuttings Analysis

X-ray fluorescence (XRF) is increasingly prevalent as a wellsite cuttings analysis technique. The utilization of portable XRF devices allows for swift measurement of the elemental composition of cuttings as they are drilled. These devices operate by detecting the fluorescent X-rays emitted when the sample is exposed to an energetic X-ray source. Each element within the sample emits X-rays at specific wavelengths, enabling the measurement of the emitted X-ray spectrum and the subsequent quantification of corresponding element amounts in the sample. By analyzing trends in element concentrations, valuable insights can be gained, such as inferring sediment depositional environments, identifying sources, and indicating conditions favorable for organic material preservation. The collected XRF data is useful for geologic characterization, optimizing well placement, and providing additional guidance for geosteering during drilling operations.

For this tutorial, the dataset comprises X-ray fluorescence (XRF) measurements obtained from cuttings extracted from the lateral section of an unconventional well. The measurements were taken at approximately 10-meter intervals along the wellbore. Each sample in the dataset is associated with 22 measurements, representing the weight percentage of a specific chemical component. To work with this data, the pandas library is employed to read the information from a CSV file and store it in a dataframe. Pandas offers various convenient data structures and tools that are widely used in data science for data manipulation and analysis.

```
geochem_df = pd.read_csv('XRF_dataset.csv')
```

A dataframe is a 2-dimensional labeled data structure in which data is organized into rows and columns. In this context, each row of the dataframe represents a distinct sample, while the columns contain measurements corresponding to each sample. The contents of the dataframe, as shown in Table 1, would typically include the XRF measurements of cuttings from the unconventional well's lateral section.



	Well Name	Depth	Quartz	...	SO3	Cl	Zr
0	Well 1	3173.97	27.56	...	1.20	0.28	
1	Well 1	3183.11	42.92	...	0.81		
2	Well 1	3192.26	44.55	...	0.76		
...	...	...	...	...	...	...	...
266	Well 1	6255.50	45.04	...	0.97		
267	Well 1	6273.78	41.21	...	1.05		
268	Well 1	6296.64	46.72	...	0.77		

Table 1. Partial contents of the dataframe after loading the dataset. The entire dataset consists of 269 cuttings samples and 22 measurements of each sample.

In the context of machine learning, the term "feature" pertains to the attributes of the objects under consideration, which are utilized to describe, cluster, and classify them. In this specific scenario, the objects being studied are cuttings samples, and the features consist of the 22 X-ray fluorescence (XRF) measurements. To enhance the dataset, we can perform feature engineering, which involves leveraging domain knowledge to create new features that aid machine learning algorithms in discerning patterns within the data.

In geochemistry, elements are employed as proxies, providing valuable indications of the physical, chemical, or biological events that occurred during the formation of rocks. Ratios of specific elements can be used to infer various geological processes and conditions

In the context of geochemical analysis, the relative strength of various effects can be indicated through the examination of specific ratios. The focus here is on three specific ratios: Si/Zr, Si/Al, and Zr/Al.

1. Si/Zr ratio: This ratio is utilized to reveal the relative proportions of biogenic silica and terrestrial detrital inputs. Variations in the Si/Zr ratio can provide insights into the contributions of biogenic silica (e.g., from plankton) and terrestrial detritus (e.g., from erosion) in the sample.

2. Si/Al ratio: The Si/Al ratio serves as a proxy for biogenic silica to aluminous clay. By examining this ratio, one can gain information about the balance between biogenic silica and aluminous clay minerals in the sample, which is valuable in understanding sedimentary processes.

3. Zr/Al ratio: This ratio acts as a proxy for terrigenous input. The chemical behavior of Zr suggests that the Zr/Al ratio can be used as an indicator of grain size variations in the sample. It provides insights into the relative abundance of terrigenous materials, which can aid in inferring sedimentary environments and provenance.

By analyzing these ratios, geoscientists can draw valuable conclusions about the physical, chemical, and biological processes that influenced rock formation and the properties of the samples under study.

```
geochem_df['Si/Zr'] = geochem_df['SiO2'] / geochem_df['Zr']
geochem_df['Si/Al'] = geochem_df['SiO2'] / geochem_df['Al2O3']
geochem_df['Zr/Al'] = geochem_df['Zr'] / geochem_df['Al2O3']
```

Dimensionality Reduction

Multivariate datasets are abundant with variables, offering the ability to explain complex behavior that cannot be captured through a single observation. Multivariate methods enable us to analyze changes in multiple observations simultaneously. Considering the abundance of observations, it is likely that the observed changes are linked to a smaller number of underlying causes. Dimensionality reduction is a technique used to leverage correlations in the observed data to uncover a more concise underlying model that explains the observed variation.

Exploratory Factor Analysis (EFA) is a method employed to decrease the number of variables by identifying latent factors underlying the dataset. These factors cannot be directly measured but can only be inferred through measuring observable properties. For instance, in the case of a geochemical dataset, a "shaliness" factor may be associated with high readings of silicon dioxide, calcium, and quartz. EFA assumes that the observations are a linear combination of the underlying factors, along with some Gaussian noise.

Principal Component Analysis (PCA) is another dimensionality reduction technique related to EFA. PCA identifies components as weighted linear combinations of the observations and assumes that these components account for all of the observed variance in the data, without being influenced by error or noise. Unlike PCA, EFA considers the presence of variance due to error.

Before applying EFA, it is crucial to standardize the features by rescaling them to have zero mean and unit variance. This standardization step is essential for many machine learning algorithms. If measurements were made using different scales (e.g., ppm and wt%), without accounting for the scale differences, it may lead to skewed factor analysis and inaccurate weights assigned to each factor. The scikit-learn library provides tools in the preprocessing module to facilitate the rescaling of the dataset, ensuring a consistent and appropriate application of EFA.

```
from sklearn.preprocessing import scale
data = geochem_df.ix[:, 2:]
data = scale(data)
```

EFA requires that the number of factors to be extracted is specified a priori. Often, it is not immediately obvious how many factors should be specified. Many authors have proposed rules over the years. One simple approach (known as the Kaiser criterion) involves looking at eigenvalues of the data’s covariance matrix and counting the number above a threshold value (typically 1.0). The following code snippet will compute and plot the eigenvalues

```
covar_matrix = np.cov(data, rowvar=False)
eigenvalues = np.linalg.eig(covar_matrix)[0]
plt.plot(eigenvalues, 'o-')
plt.axhline(y=1.0, color='r', linestyle='--')
```

The resulting plot is shown in Figure 1. There are 6 eigenvalues greater than 1.0 (dashed red line), suggesting there are 6 relevant factors to be extracted.

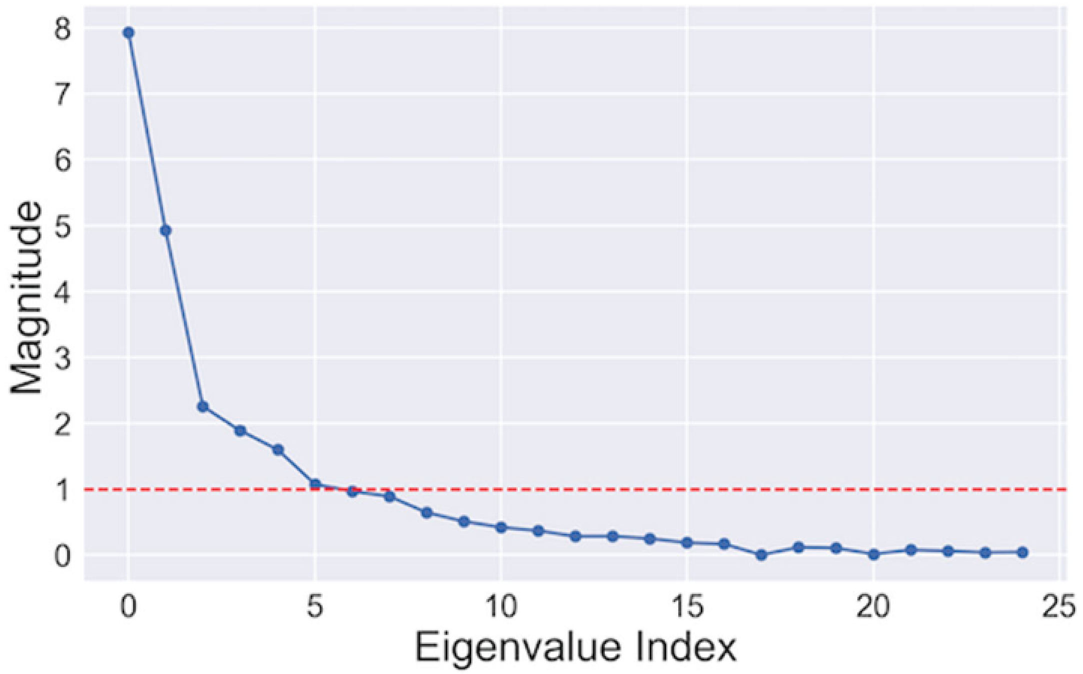


Figure 1. Eigenvalues of covariance matrix.

The scikit-learn library contains a factor analysis module that can be used to extract the 6 factors. This is done by creating a factor analysis model and fitting the model to the data.

```
fa_model = FactorAnalysis(n_components = 6)
fa_model.fit(data)
factor_data = fa_model.transform(data)
```

Interpreting the factors

The factors can now be examined to interpret the underlying properties they represent. The factor loadings describe the relationship of each measurement to the underlying factor. Each loading score represents the correlation between the factor and the observed variables. The loading scores vary between -1 and 1. A positive value means that a measurement is associated with the presence of an underlying factor. A negative value suggests that a measurement indicates the absence of a factor. The factor loadings can be easily extracted from the factor model and plotted to show the loadings associated with a given factor.

```
loading = fa_model.components_
component_names = geochem_df.columns.values[2:]
loading_df = pd.DataFrame(loading, columns=component_names)

# plot the 5th factor
loading_df.ix[4].plot(kind='bar')
```

Figure 2 shows the factor loadings associated with the fifth factor. In this case, the factor is associated with high values of plagioclase, illite/smectite/mica, pyrite, and organic material, and the absence of calcite and MgO. We could interpret this factor as the organic-rich clay content. Similar interpretations can be given to the other factors by observing their loading scores.

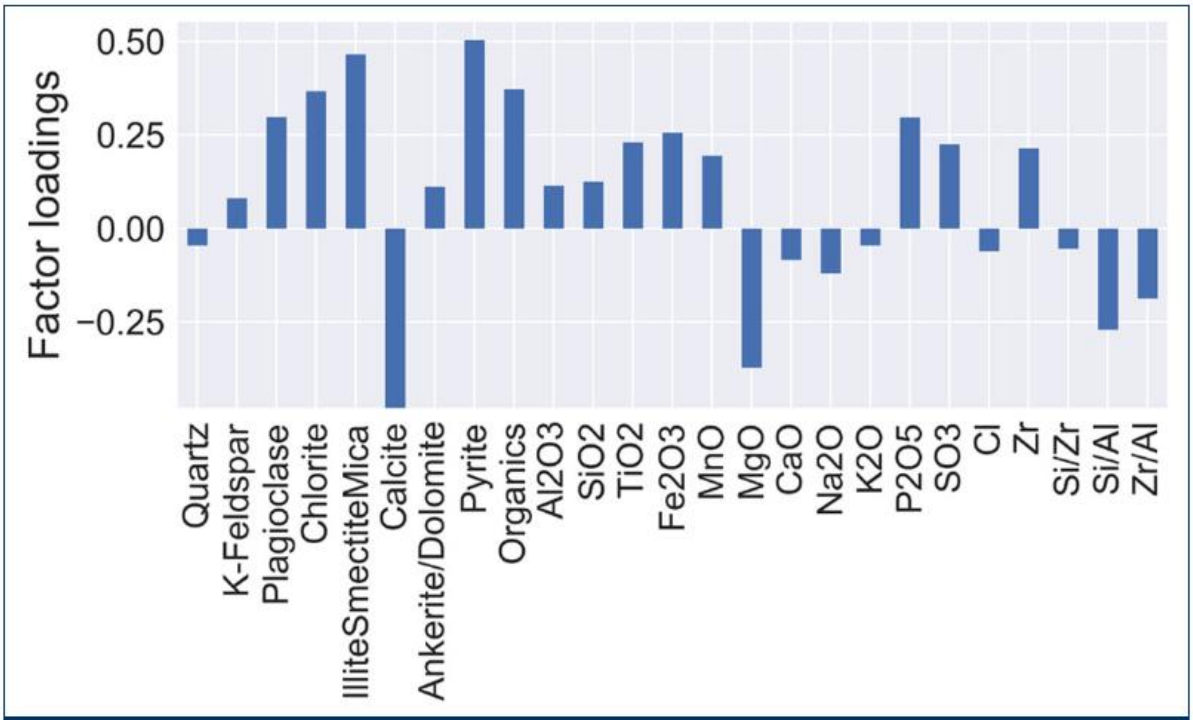


Figure 2. Factor loadings associated with the 5th factor.

Clustering

Cluster analysis is a suitable approach to achieve this grouping task. It aims to cluster or group samples together based on their similarity, where samples within the same cluster share similar characteristics compared to those in other clusters. Cluster analysis belongs to the category of unsupervised machine learning techniques, as it does not require labeled training data to guide the model.

The K-Means algorithm is a widely used clustering method. It works by attempting to partition the data into k groups (clusters) of equal variance. The algorithm iteratively finds the optimal cluster centroids by minimizing the distance between each data point and the closest centroid. Similar to EFA, K-Means requires that the number of clusters is specified before running the algorithm. There are a number of approaches to finding the optimal number of clusters. The goal is to choose the minimum number of clusters that accurately partition the dataset. These range from the relatively simple 'elbow method' to more rigorous techniques involving the Bayesian information criterion and optimizing the Gaussian nature of each. The following code demonstrates the 'elbow method' applied to this dataset. The sum of the squared distance of each point to the nearest cluster centroid (called inertia in scikit-learn) is plotted for an increasing number of clusters. As the number of clusters is increased and better fit the data, error is decreased. The elbow of the curve represents the point of diminishing returns where increasing the number of clusters does not reduce the error appreciably. Figure 3 suggests that about 6 clusters would be adequate for this dataset.

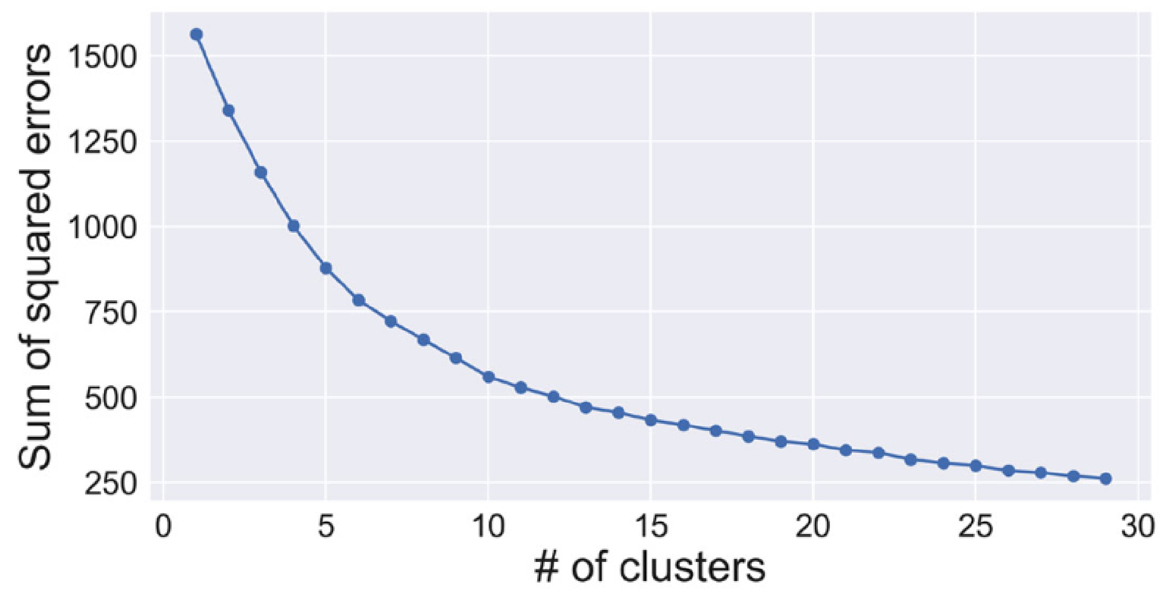


Figure 3. Mean squared error vs. number of clusters for the XRF dataset.

```
inertias = []
means = []
maxK = 30
for k in range(1, maxK):
    kmeans = KMeans(n_clusters=k, random_state=0).fit(factor_data)
    means.append(k)
    inertias.append(kmeans.inertia_)
plt.plot(means, inertias, 'o-')
```

The K-means algorithm in scikit-learn is used to cluster the reduced dataset. Similar to the factor analysis, this is done by creating a K-means model and fitting the factor dataset.

```
kmeans = KMeans(n_clusters=6, random_state=0)
kmeans.fit(factor_data)
# add the cluster ids to the dataset
geochem_df['Cluster'] = kmeans.labels_ + 1
```

Interpreting the clusters

Each sample in the dataset has now been assigned to one of six clusters. If we are going to interpret these clusters as geochemical facies, it is useful to inspect the geochemical signature of each cluster. Figure 4 contains a series of box plots that show the distribution of a small subset of measurements across each of the 6 clusters. The box plots depict 5 descriptive statistics; the horizontal lines of the colored rectangle show the first quartile, median, and third quartile. The arms show the minimum and maximum. Outliers are shown as black diamonds.

```
sns.boxplot(x='Cluster', y='Calcite', data=geochem_df)
```

Figure 4A indicates that Cluster 1 is characterized by a relatively high (and variable) Illite/Smectite/Mica component, and the highest organic content (4B). Cluster 2 has the highest calcite component (4C) and cluster 6 is associated with the highest MgO concentration. Figure 4 only shows the response of 4 of the 25 measurements, but this can be done for each measurement to build up a geologic interpretation of each cluster.

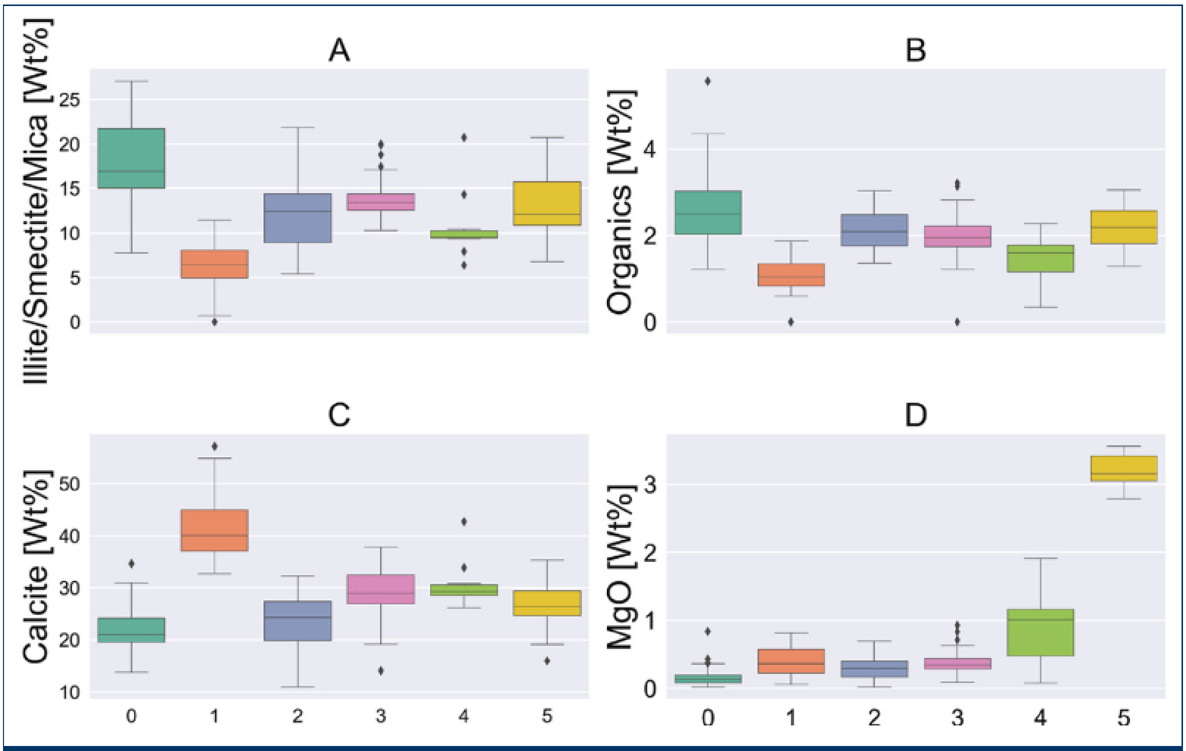


Figure 4. Distribution of illite/smectite/mica (a), organic material (b), calcite (c) and MgO (d) across the clusters.

Visualizing results

Now the cuttings sample have been organized into 6 geochemical facies (clusters). We can visualize the classification in a well log style plot to better understand how the facies are ordered vertically in a well. The right column of Figure 5 shows the clusters assigned to each sample using a unique color, indexed by measured depth (MD). The other columns show 4 of the corresponding geochemical measurements. Similar plots could be made for the other wells in the dataset and used to identify common intervals.

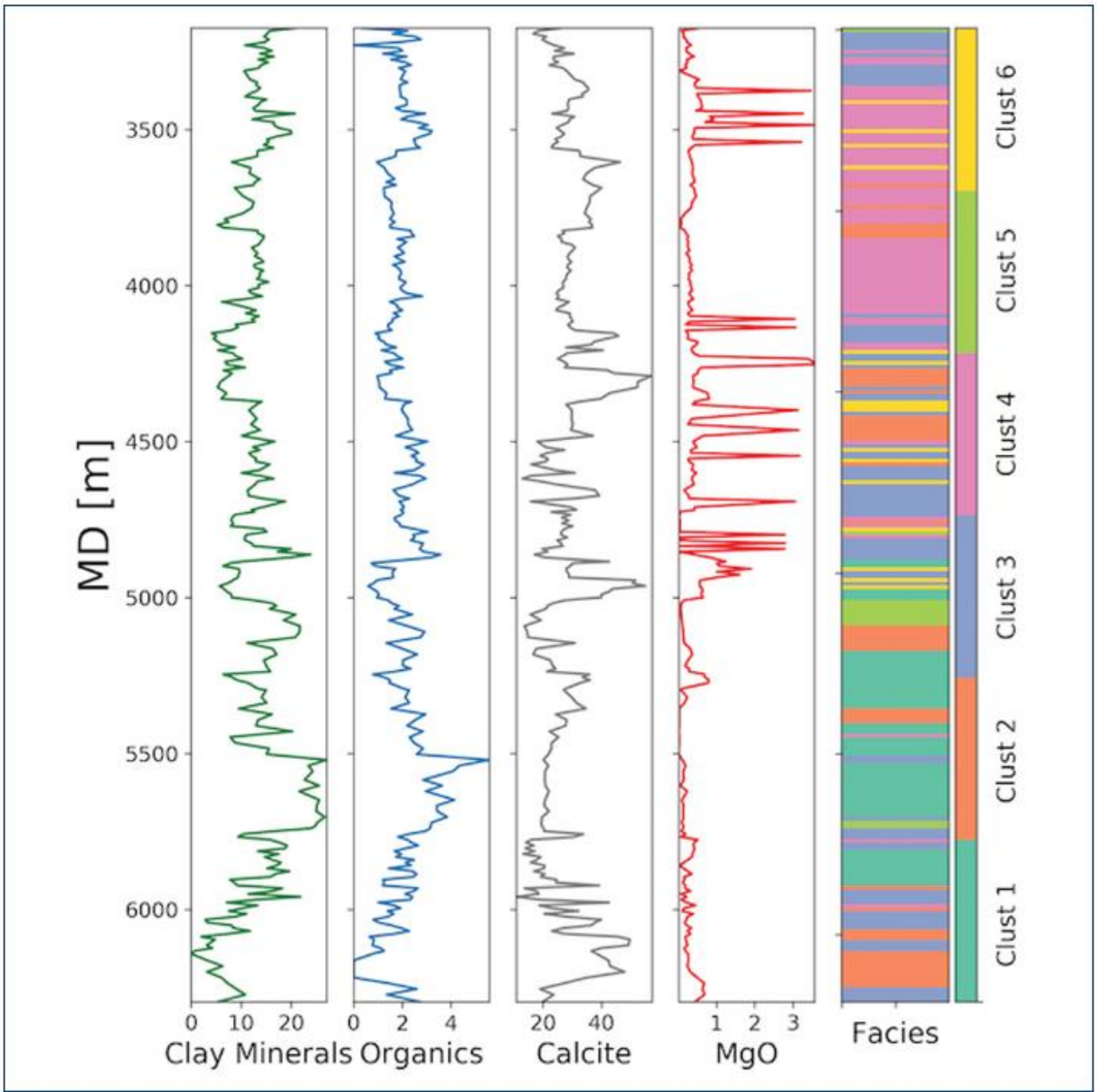


Figure 5. Log style plot showing 4 geochemical measurements and cluster assignments.

One application of this analysis is providing data that can be used for geosteering horizontal wells. This is particularly useful in areas that lack a distinctive gamma ray signature. Wellsite classification of cuttings sample could be used to interpret a well's path through an existing chemo-stratigraphic framework. To build a visualization of this framework, it is helpful to plot the geochemical facies along the well path. Figure 6 shows the trajectory (TVD vs. MD) of the well, with the different facies colored using the same scheme as Figure 5. This can be used to build a pseudo-vertical profile and help identify specific zones as the well porpoises up and down along its length.

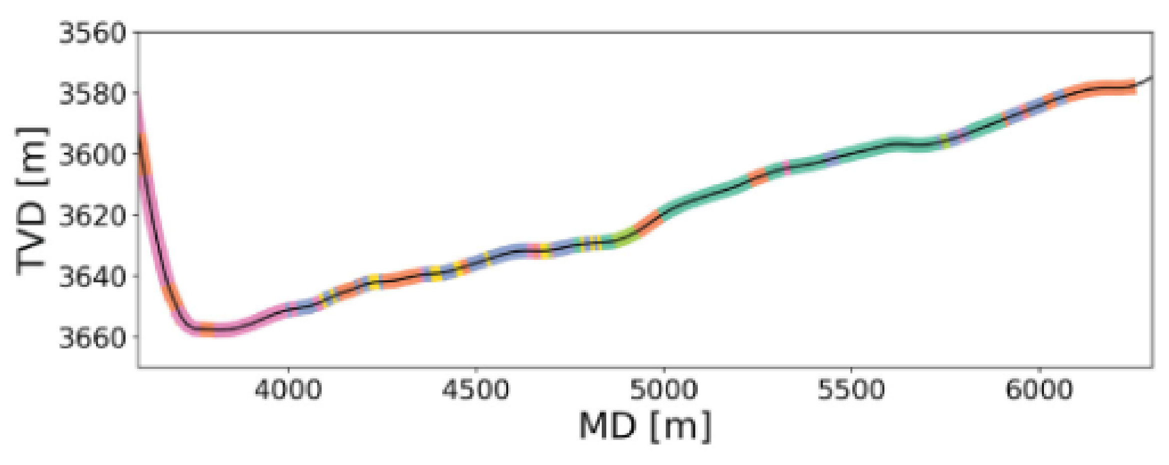


Figure 6. Geochemical facies assignments plotted along well trajectory.

This tutorial has demonstrated how dimensionality reduction and unsupervised machine learning can be used to understand and analyze XRF measurements of cuttings to determine geochemical facies. Exploratory factor analysis yields insight into the underlying rock properties that are changing across the reservoir. K-means clustering is used to organize similar samples into a smaller number of groups that can be interpreted as geochemical facies. This can be used to correlate formation tops between wells and provide data necessary for geosteering.