Keenan Flynn

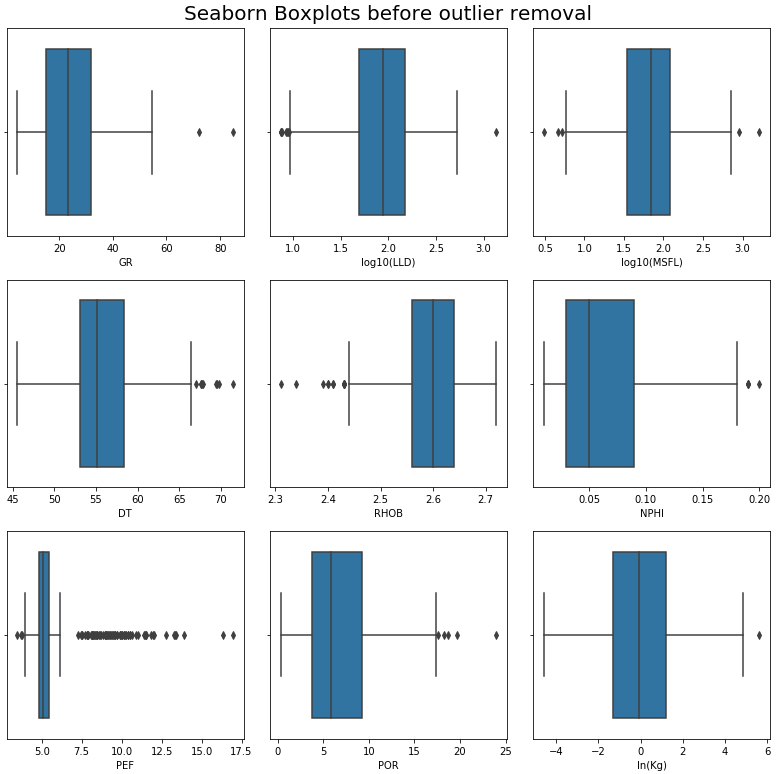
PETE 4161

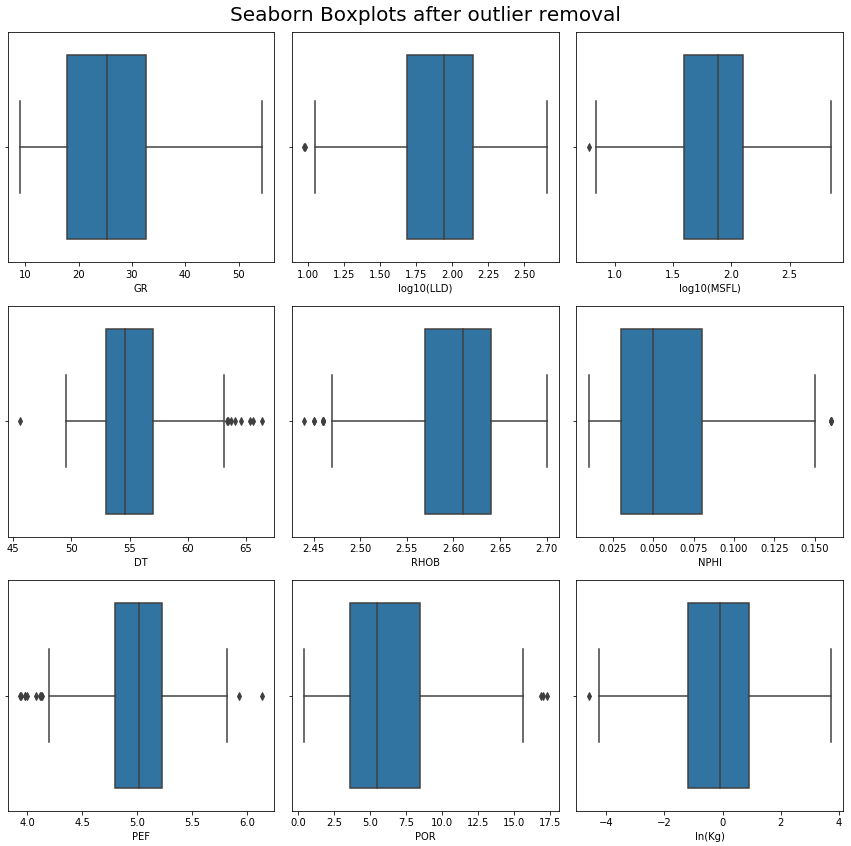
Salt Creek Data Exploration & Reduction, Classification, and Regression

In the textbook Applied Statistical Modeling and Data Analytics, the dataset ‘Salt-Creek.DAT’ is used as a vehicle to convey various methods of data modeling. In this report, we will similarly be examining and exploring the well logs from Salt Creek field. The data file contains 403 rows and 10 columns of well log data. The columns in order are ‘GR, log10(LLD), log10(MSFL), DT, RHOB, NPHI, PEF, POR, Kg, ln(Kg)’. The file contains the typical suite of well logs: Gamma Ray, Induction, Density, Photoelectric Absorption, and Sonic. These are our independent variables. Dependent porosity and permeability information is also given. Well log analysis is a particularly vital aspect of petroleum engineering as it is one of the main tools used to evaluate economic zones in a well. Historically this examination has been done by specialized professionals who manually interpret the well logs for hydrocarbon bearing strata. In this report, we will use statistical analysis, classification, and regression to gain a computer driven insight into the Salt Creek dataset.

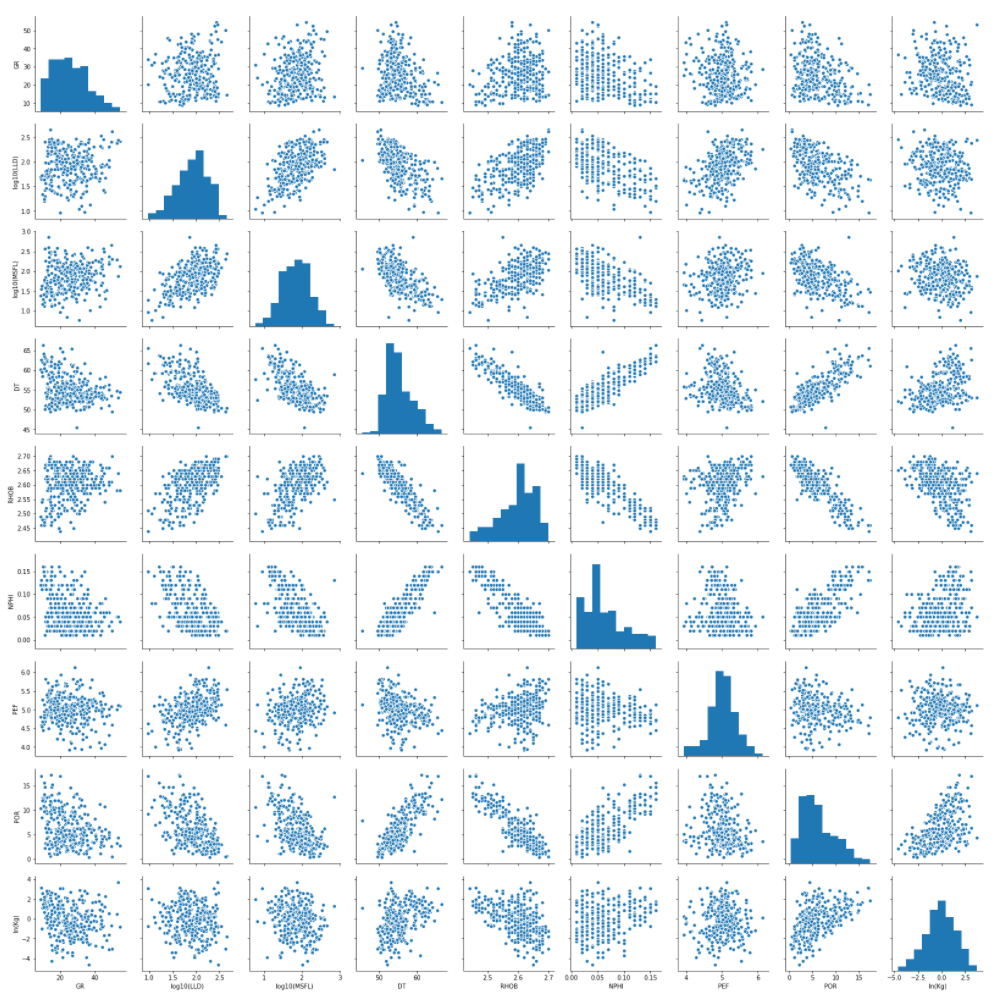
The first step in analyzing the data is to get our data into a usable format. We can do this by reading the DAT file into a Pandas DataFrame object. The datatype of each column is a 64 digit float, which is important to know when performing mathematical operations on the data later. With the datatype information, we can drop irrelevant columns which include the variable ‘Kg’ because we already have this information in the form of ‘ln(Kg)’. I checked to see if there were any duplicate rows, however there were none. There were also no null values.

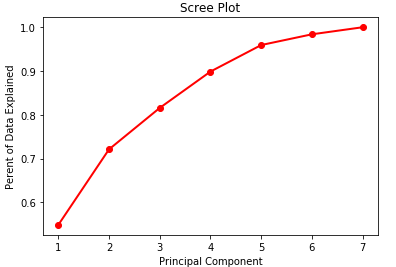
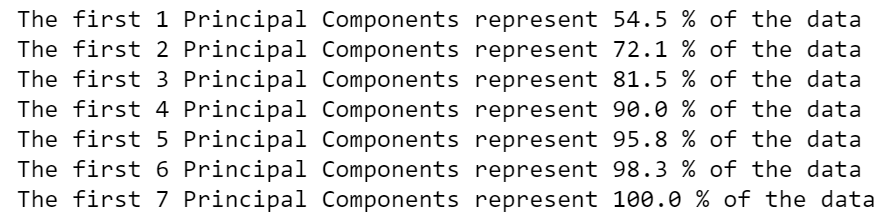
The next step in exploratory data analysis was to remove outliers. We can visualize what kind of outliers we have with boxplots and we can determine which data to remove using the Interquartile range method (IQR). The first and third quartile was taken of the aggregate data, and IQR calculated as third quartile minus first quartile. If a sample of data was not within +/- 1.5 times the IQR, it was removed. 1.5 is a common number to use in this method, but if we want fewer or more outliers removed, we can tune this number. In this case, 403 rows of data were initially in the DataFrame, and after removal of outliers there were 286.





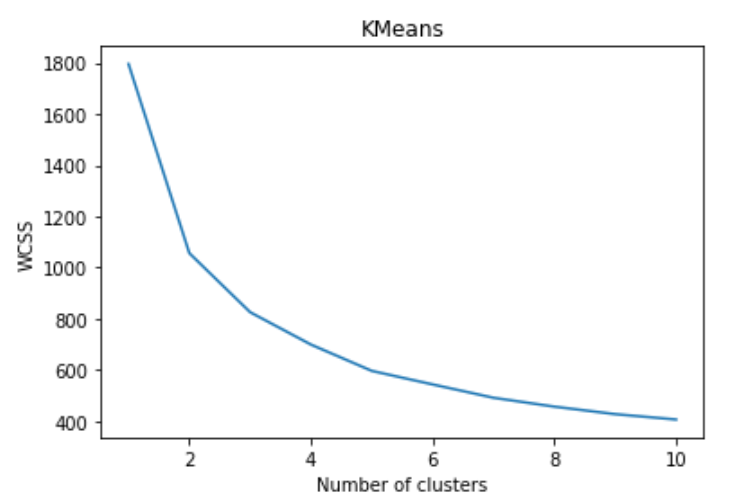
The final step in data exploration is visualizing correlations between the data. With Seaborn’s pairplot we can see relationships between variables. In the case of salt creek, when examining the diagonal of the pair plot, all distributions are either gaussian or log normal.



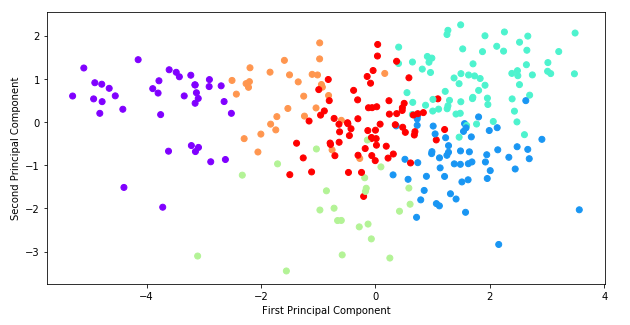
The salt creek data is now ready for dimensional reduction using the PCA method. This method is used rather than dropping columns because of the minimal variance lost. After standardizing the data, a covariance matrix was found. This eigenvalues and eigenvectors of this “C” matrix were then found and sorted so that we can examine the total variance explained of each eigenvector/eigenvalue pair or principal component. The results of this examination are as follows, in the form of text and in the graphical form of a scree plot.

To determine how many principal components to keep, I used a combination of the elbow method in the case of the scree plot and an abstract minumum value of 90% variance explained. This led me to choose the first four principal components as the components to keep, with the first 2 components explaining about two-thirds of all data. The data can now be classified.

K-means clustering was used to classify all 4 principal component data into groups. This algroithm was provided through scikit learn’s cluster library. The number of clusters chosen was 6. This was chosen using WCSS method. WCSS plotted against the number of clusters.

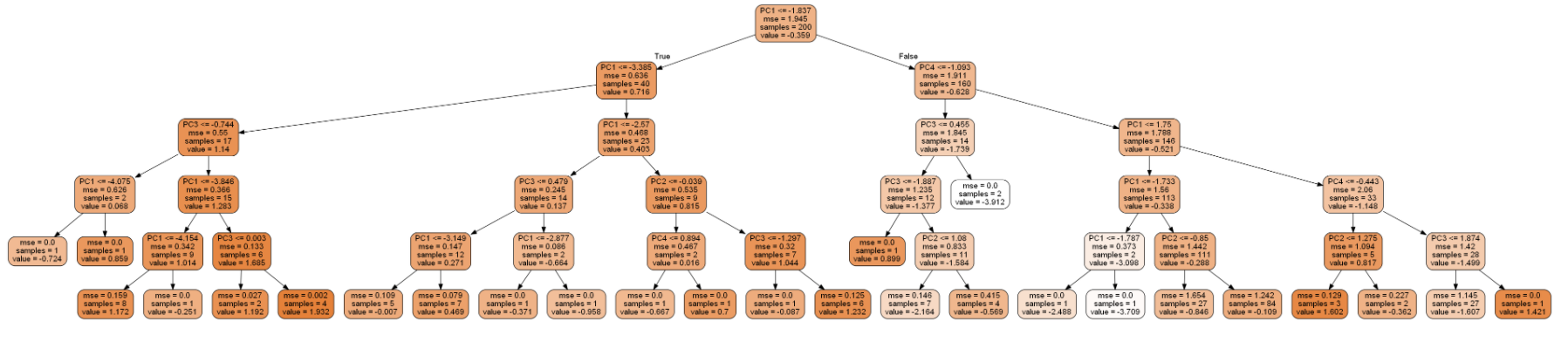


The evaluation of how many clusters to choose was done using the elbow method and 6 clusters were chosen. The 6 clusters were visualized along the first and second principal components.

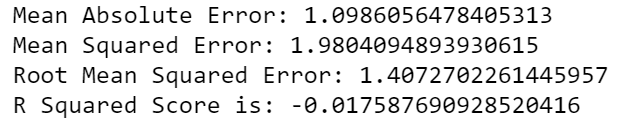


The principal components of the Salt Creek set were finally used to predict permeability (or ‘ln(Kg)’) by means of a regression tree. The tree was created using scikit learn’s ‘DecisionTreeRegressor’ function. A regression tree was chosen because our predictive values are continuous and quantitative rather than qualitative. To achieve qualitative predictions the ‘DecisionTreeClassifier’ function could replace the regression tree function and the code would work without error.

The data was split into 30% test and 70% training data. Our ‘x’ data is the input from our well logs and the ‘y’ is the variable we want to predict, being either porosity or permeability. Our criterion for the split is mean square error and our hyperparameter is a max depth of 5. The result is the tree below.



We can use the tree to get an array of predicted ‘y’ values. These predicted values were then evaluated using 4 different metrics: Mean absolute error, Mean Squared error, Root mean square error and R squared.



In conclusion, the unsupervised (Kmeans clustering) and supervised (regression tree) methods described above have created a model that can both classify data into groups and predict values given inputs that have been rigorously cleaned and transformed. The next step in this process would be to further tune hyperparameters by use of ensemble methods and further refine the clustering, whether improving the Kmeans algorithm or testing Kmeans against other types of clustering.

Intro

* Hey everyone, name Keenan
* Thanks for listening to my presentation summarizing the 2 projects we did for this class

Project 1

* The first project was an examination of salt creek data set
  + Midwestern Texas
* This project encompass
  + Data Exploration
  + Dimensional reduction
  + Unsupervised Learning in form of clustering
  + Classification and Regression

Salt Creek

* The salt creek set consists of well log info from 7 wells in Permian Basin
* Highly heterogenous carbonate reservoir
  + In theory see statistical significance between the wells
* 7 well logs -> name
  + #GR -> Gamma Ray Log
  + #LLD -> Laterolog Deep
  + #MSFL -> Micro Resistivity
  + #DT -> Sonic Log
  + #RHOB -> Density log
  + #NPHI -> Neutron density
  + #PEF -> photoelectric absorption
* Measured perms/porosity for each data point

Data Exploration 1

* Data Exploration is first step in preprocessing the data to fit in our learning models
* Much work already done, each column is 64 digit float, there are no duplicates in the data
* Remove outliers, visualized through seaborn boxplots
* PEF shows the most radical transformation with many outliers removed
* Outlier removal done through IQR method
  + First and Third quartile were found for the entire dataset, the difference=IQR
  + If a datapoint lays a distance of 1.5\*IQR behind the first quartile or in front of the third quartile it was removed
  + 1.5 industry standard

Data Exploration 3

* Finally pairwise relationships were evaluated using seaborns pairplot
* Scatter plots show relationships between variables and diagonal histograms show relationship within variables
* We can see that outlier removal got us to a point where we have some good looking gaussian or lognormal distributions in each attribute with a few hiccups here and there

Dimensional Reduction 1

* Before the data can be modeled in machine learning, we must reduce the dimensions.
* First the data was standardized by subtracting the mean and dividing by stan dev for each point
* Next an Eigendecompositon was performed using numpy linear algebra suite, giving us 7 principal components
* Each of these components describes a certain amount of variance of the data and the components that describe the least were dropped

Dimensional Reduction 2

* Here you can see the process that was used to drop the extra PCs.
* I chose an arbitrary value of 90% of total variance that my PCs needed to describe which correlated to exactly 4 PCs
* I also used a scree plot, however it didn’t have any point where there was a clear elbow so it was less useful

Unsupervised Learning 1

* With an array of the PCs, we can now begin machine learning
* First the data was put through K means clustering, which is unsupervised
* To choose the right amount of clusters we want to minimize the WCSS score or the within cluster sum of squares
* We can visualize this by making another elbow plot. This time our plot gives us a pretty good elbow and through testing I decided that 6 was the right amount of clusters

Unsupervised Learning 2

* Once num clusters known, we can simply use 2 lines of code to split the data
* Each data point is now assigned a label
* Scatter plot all data with the color representing a cluster
* Each cluster represents an electrofacies
* The reason there is overlap is because the labels were created from clustering all 4 Principal Components and this plot is only projecting 2 of the four PCs

Supervised Learning 1

* With defined classes, we can now classify
* The classification method I chose was a Random Forest of 100 decision trees
  + Better because than a single decision tree because that could overfit our data
* Each tree works by asking a logical question on each level
* For example, if x value less than t1……… electrofacies R1

Supervised Learning 2

* This is what one of the trees looks like
* First line = logical expression
* Second line is the split criterion, measures quality of split
* Third line = num samples
* Cluster numbers of all those samples
* Unrestricted depth of levels

Supervised Learning 3

* These are the results of the classification
* In this run it only classified 4 data points falsely
* Precision, recall, f1 all above 90%
* Total Accuracy score of the model averages above 90%

Regression 1

* Very similar to the classification random forest, I used a regression forest to try to predict perm
* Input values were the principal components, with expected outcomes to be permeabilities
* First line = split
* Second measures quality of split
* Samples
* Scores for the model
  + R2 is very low and I couldn’t even get it to be positive, so theres some work to be done still with more time

Project 2

* Examination of time series data from the Volve Field
  + Off coast of Norway
* In this project I did some……..

Data Exploration

* 16 wells in the dataset, conveniently they each have 3197 days of production.
  + Starting on January 1, 2008
  + Ending on Oct 1, 2016
* For most wells, a majority of Production days were zeros, so I removed any zero days of production to find………
* In this project I will fit models to well P-F-14 and try to validate my results on P-F-12

Data Exploration 2

* For each well there was about 50 attributes
  + Daily Oil, water, gas
  + Cum Oil water gas
  + Daily liquid, cum liquid
  + Pressure data
* I chose to focus on just 2 of these attributes
  + Days since January 1, 2008 and Daily Oil rate
* Both wells have similar production
  + Upwards slope while coming online, downward concave after reaching peak
* You can see the effects of removing zero days of production
  + Bit of a smoother line

Supervised Learning

* For supervised learning I did a persistence model
* The persistence forecast is where the observation from the prior time step (t-1) is used to predict the observation at the current time step (t).
* Good in sections where it is flat…..show…. little error
* Not so good in sections with large increase/decrease
* Drawback of this is limited horizon of just 1
* High r2 in my case

LSTM

* LSTM is supposed to be able to have a longer horizon than other types on models that aren’t covered under machine learning
* It is recurrent neural network
  + Learn/remember over long sequences
* Used LSTM that
* Model was fit with about an 80/20 split
* To get our data into formatting for LSTM
  + Transform data into a supervised learning problem
    - Input of t-1 and output of t1
  + Transform data to stationary
    - Removes the trend from data by differencing the data
  + Finally scale the data so that it lies between -1 and 1
* Mean square error is lower than persistence model by about 15
* Only forecasting 1 value ahead

Validate

Regression 1

* Very similar to