Capstone Project 1: In Depth Analysis

After initial data exploration, an in-depth Machine Learning algorithm testing was performed.

1. Separated Different Zoning by Petrophysics Chracteristic

Petrophysics-based zonation for both training and test set based on GR, ZDEN, CNC. Compared to test set, I found the training set showed similar spatial patterns, to take advantage of the spatial correlation. I performed inter-well correlation. Per-zone estimation has been proved to be a good preprocessing method for well-log interpretation (Pan et al., in press; Pan et al., 2019).

For this dataset, there are 5 zones in the training dataset, as shown in the following figures with different colors representing different zones:

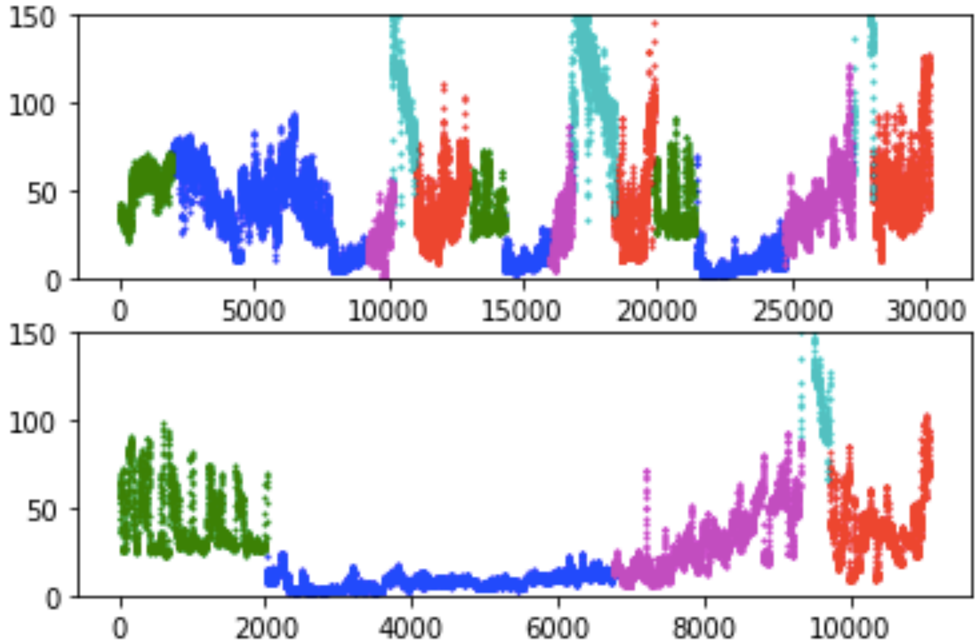

Figure 1: Top figure shows training data rezoning. Bottom figure shows test data rezoning distribution.

Then the cross plots of different well logs is shown below. as expected, different zones show different correlations and spans of data.

2. Preprocess data to reduce impact of miscorrelation between index and depth

Preprocessing: 10000 data points in the test well should roughly correspond to 5000 ft, which is too long for a well, thus I think it could be resampled. The artifacts in the cross plots (continuous dots) also indicate the existence of resampling. To alieviate the aliasing problem caused by resampling, and try to restore the true correlation, I calculated gradients to differentiate real data from interpolated data, and use medidan filter to capture the general trends of different well logs. I also perform logrithmic transformation to resistivity logs to avoid large weight on resisitivity logs.

Figure 2: Cross plot of different logs shows clear different patterns of different zones.

```
df1['HRD_log']=np.log(df1.HRD.values)
df2['HRD_log']=np.log(df2.HRD.values)
df1['HRM_log']=np.log(df1.HRM.values)
df2['HRM_log']=np.log(df2.HRM.values)

# Calculate the gradients of input logs
# Interpolated well logs should have similar gradients
for i in df1.keys():
    df1[i+'grad']=np.gradient(df1[i].values)
for i in df2.keys():
    df2[i+'grad']=np.gradient(df2[i].values)
#Observe the patterns of logs
```

Figure 3: This code snippet shows how to perform gradient calculation on different well logs.

3. Model Building with Deep Learning Approach

I Perform sonic logs estimation for each zone. Caliper log is not used because there is no strong correlation between caliper log and sonic logs, sonic logs is sensitive to the density and porosity of the formation, thus the gradients of logs that measures the density and porosity of the formation are used as input features.

There are two more processing steps were adopted before using the ML model: (1)Min and Max scalar was applied to the data for normalization. (2) Median filter was applied to reduce the amount of transient noises existing in data.

After preprocessing a simple fully connected ANN network, with very few nodes, was applied. The first layer has 24 nodes, second layer 8 nodes, and then connected to a dense layer before dropout.

```
RMSE of test data (#1 DTC): 4.77
RMSE of test data (#2 DTS): 17.28
Overall RMSE = 12.68
```

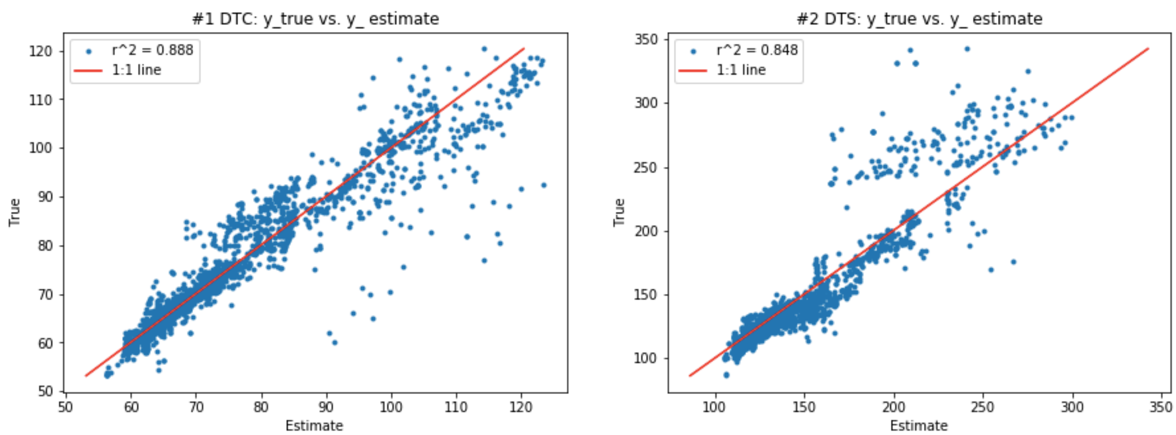

Figure 4: Comparison of predicted data and test data shows good agreement.

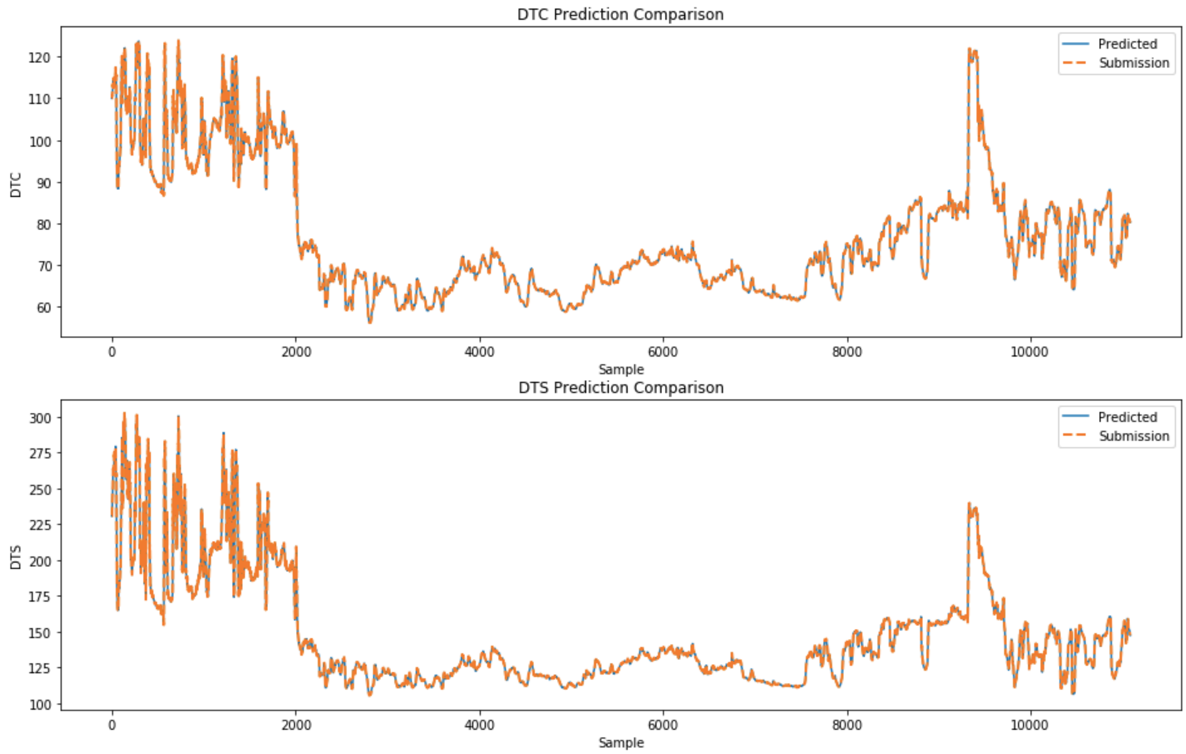

Figure 5: Prediction results plot.