Open the blackbox in Neural Network using R & Bakken Data

Marshal Wigwe

Introduction

We currently live in the world of “big data” and several operators are beginning to apply data science tools for understanding and optimization of oil and gas production, among other things. In the Bakken and Three Fork, about 88% of the 170 Bbbl. most likely estimate of the OOIP are located in six of the nineteen producing counties in North Dakota, according to NDGS 2010 assessment. These counties are Burke (10.04%), Divide (10.46%), Dunn (11.87%), McKenzie (21.51%), Mountrail (17.10%) and Williams (17.11%). The dataset used in this analysis contains 5755 wells, distributed in these counties as shown in `Table 1`. These wells were completed between 2008 and 2016, with at least one year of production recorded. The use of neural network as a predictive tool is common practice and most times, we tend to treat this tool as a “blackbox”. We provide the “box” some input parameters, and it spills out a result or prediction. However, it is important we understand what goes on under the hood, to open the “blackbox”. In the example shown here, we will predict the first twelve month oil production using well completion and production variables. We would discuss simple models, but first, let us understand our dataset.

Analysis Routine: Data Import

To get started with this analysis, direct your R session to a dedicated working directory which should contain the `bakken` dataset. Remember to convert date variables to date, we used the `lubridate` package for this. Date variables were imported by `read.csv` function as factors. Import data into R and Preprocess the data Preprocessing here implies converting date variables that were imported as factors to date.

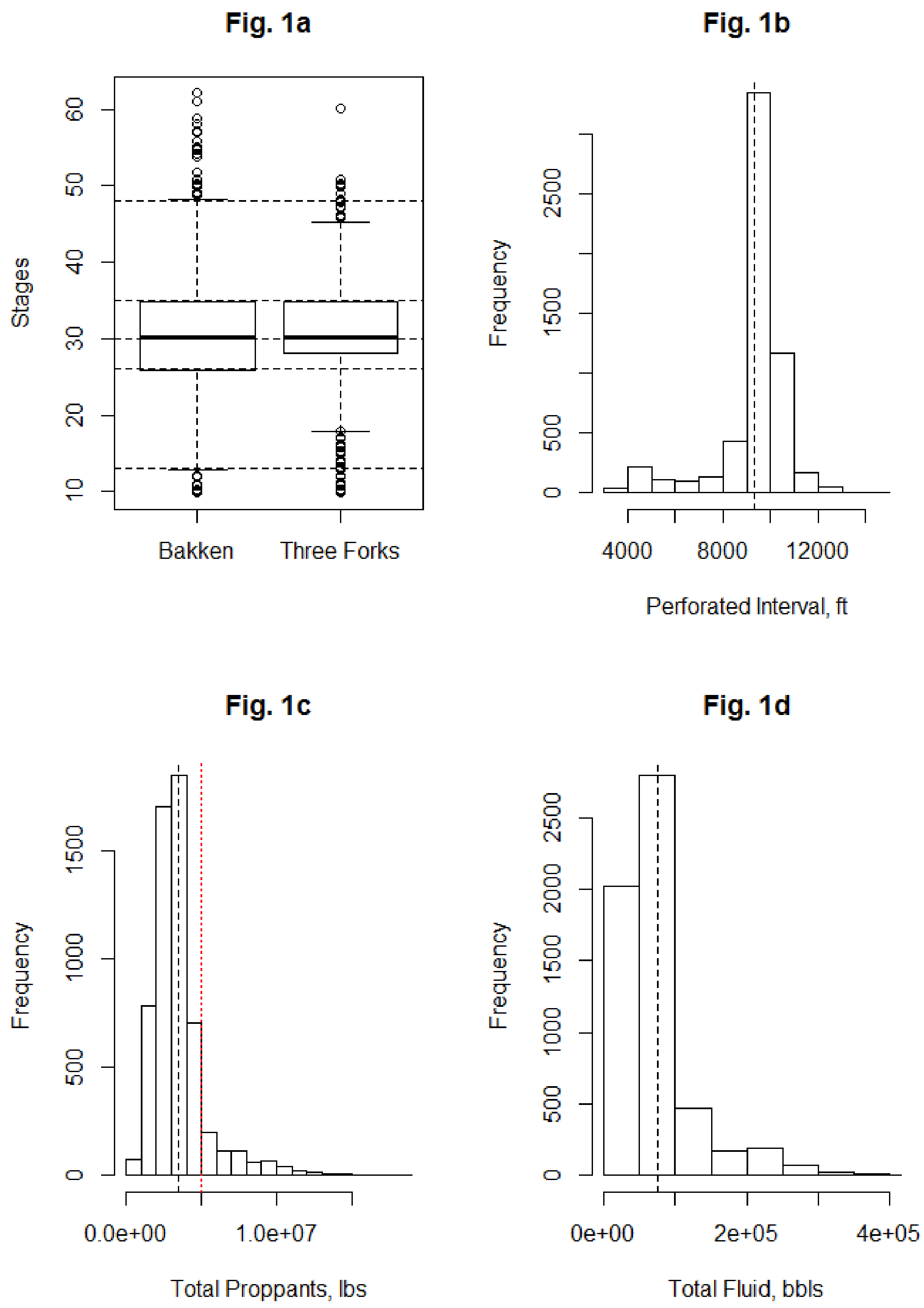
```r
bakken = read.csv("bakken_data.csv")
bakken$completionDate = mdy(bakken$completionDate)
bakken$firstProdDate = mdy(bakken$firstProdDate)

# Table 1: Distribution of wells by County and Formation
table1 = addmargins(table(bakken$County, bakken$targetFormation))
kable(table1, format = "pandoc", caption = "Table 1: Distribution of wells by
County and Formation")
```

Table 1: *Distribution of wells by County and Formation*

	BAKKEN	THREE FORKS	Sum
BURKE	87	51	138
DIVIDE	119	186	305
DUNN	658	365	1023
MCKENZIE	1185	678	1863
MOUNTRAIL	850	396	1246
WILLIAMS	843	337	1180
Sum	3742	2013	5755

Distribution of Completion Parameters

Let's take a look at the distribution of the number of stages, total pounds of proppant, total volume of fluid injected and the perforated interval typically used in frac jobs in the Bakken and Three Forks formations (`fig. 1`). The comparative boxplot shows the number of stages. We can observe that on average, operators are using the same application in both formations for the number of stages (30 stages). There is also more variability in number of stages in the bakken compared to the Three Forks. For the perforated interval, the histogram shows that most operators favor a perforated interval in the 8,000 ft. to 11,000 ft. range on the lateral. The distribution of total pounds of proppant used for the frac jobs is as shown. Most frac jobs used less than 5 million pounds of total proppants (the red line).As we shall see later, of the 656 occurrences of application of more than 5 million pounds of total proppants, only 83 cases occurred prior to 2014 (`Fig. 2`). This indicates that the use of large pounds of proppants started becoming popular during the downturn. To summarize this, on average, 75,000 bbls of fluid and 3.5 million pounds of proppants were used for the 30 stage completion of a 9,300 ft. perforated interval between 2008 and 2016.



```
## [1] "Fig. 1: Distribution of completions parameters"
```

How has Completion Parameters changed since 2008

`Fig. 2` shows the variation of completion parameters with time from 2008 - 2016. `Fig. 2a` seems to suggest an increasing trend in the number of frac stages used in completions. There does not appear to be a systematic change in the length of lateral and perforated interval since 2008 (`Fig. 2b`). However, we see an increasing tendency towards perforating the lateral in the 9,000 ft. - 11,000 ft. range. As mentioned previously, we can see that the use of more than 5MM lbs of proppants started becoming popular after 2014 (`Fig. 2c`). Most of the completions prior to 2012 utilized less than 100,000 bbls of total fluid and like the case of total proppants, the use of more than 100,000 bbls of total fluid became increasingly popular from 2013 and well into the downturn (`Fig. 2d`). This tendency to use more proppants and higher fluid volume meant that operators could complete fewer wells with a view to "increasing" production (fig. 2e).

Fig. 2a

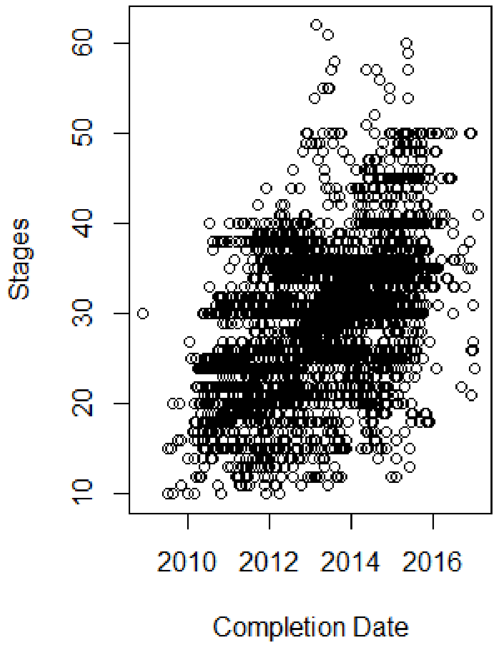
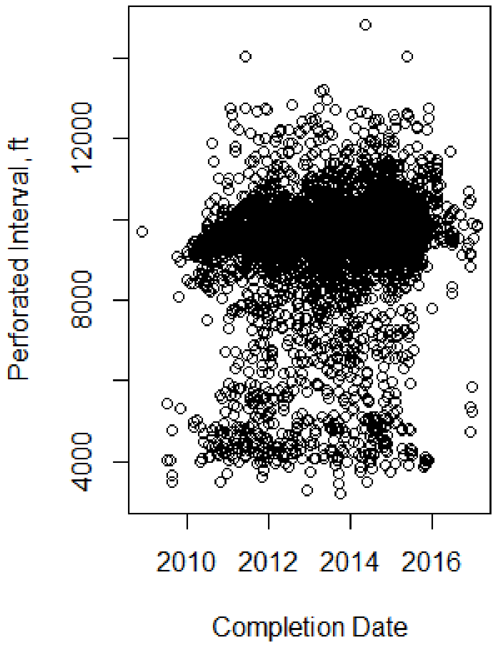
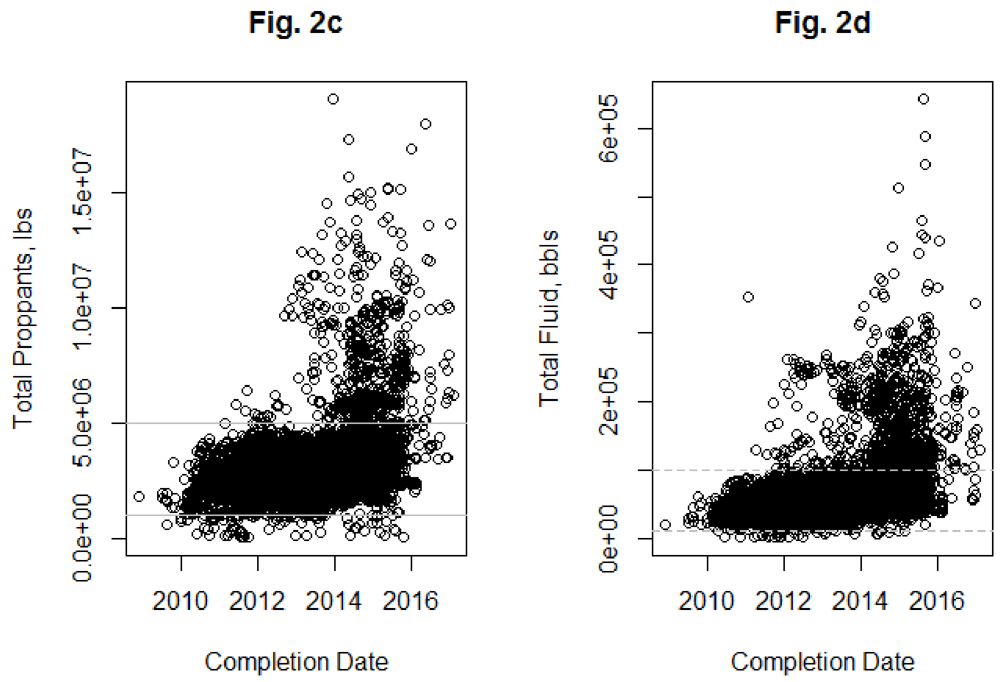


Fig. 2b

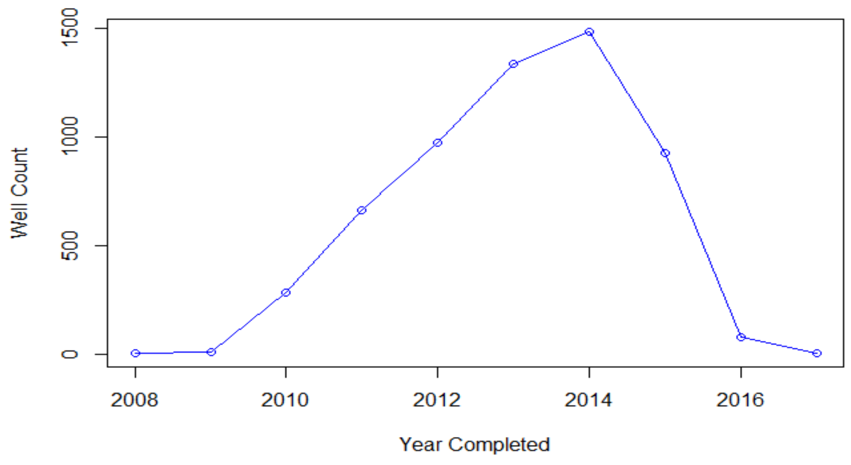




```
## [1] "Fig. 2: Distribution of completions parameters cont'd"
```

Looking at other well parameters

- Aggregate the number of wells completed by year.



```
## [1] "Fig. 2e: Wells completed by year"
```

- Looking at injection rate and pressure

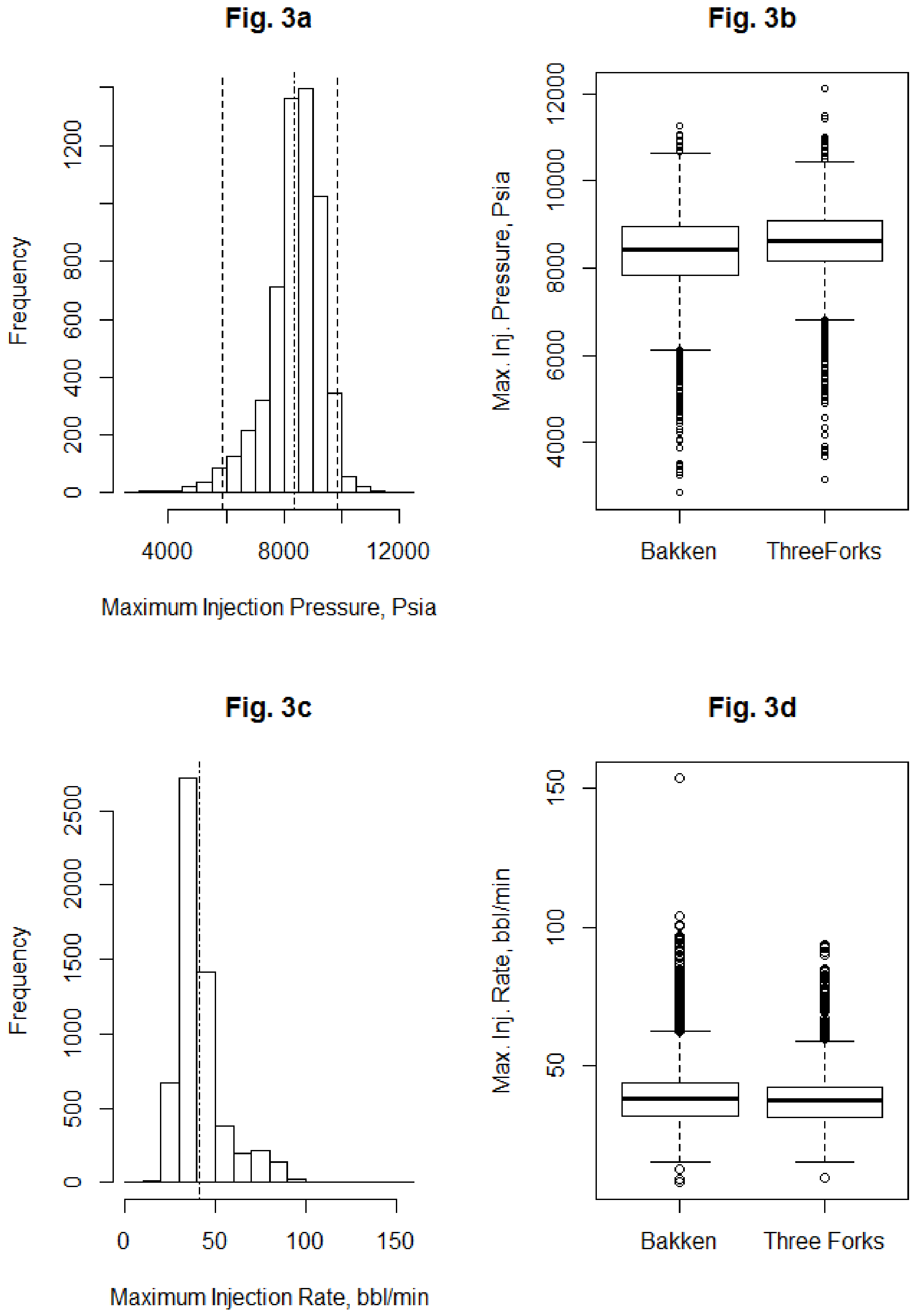


Fig. 3a

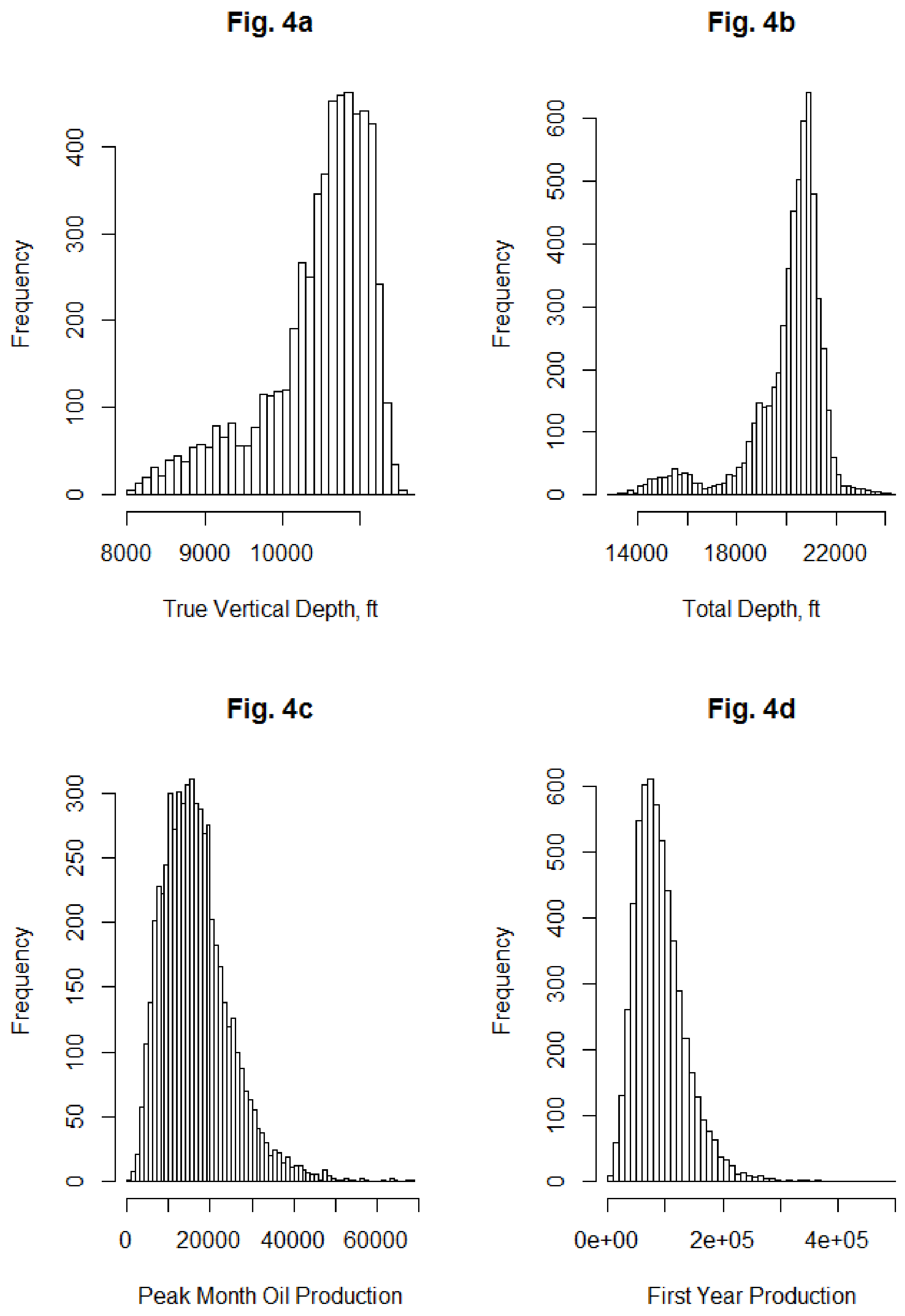
Fig. 3b

Fig. 3c

Fig. 3d

```
## [1] "Fig. 3: Distribution of Injection Rate and Pressure"
```

- Looking at TVD, TD, Peak Oil and 12 month oil



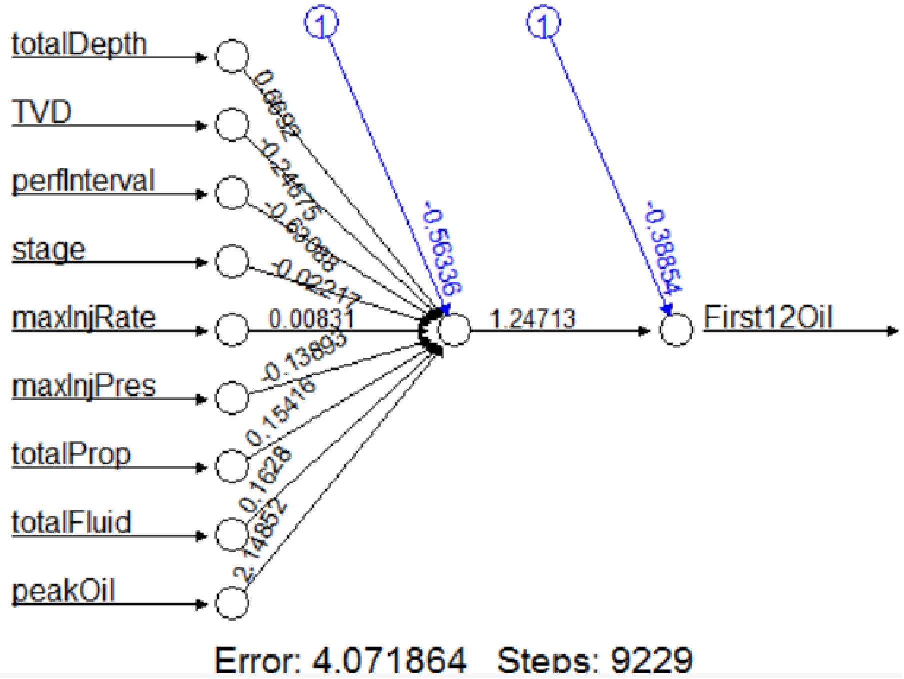
```
## [1] "Fig. 4: Distribution of Drilling and Production Variables"
```

Application of Neural Network

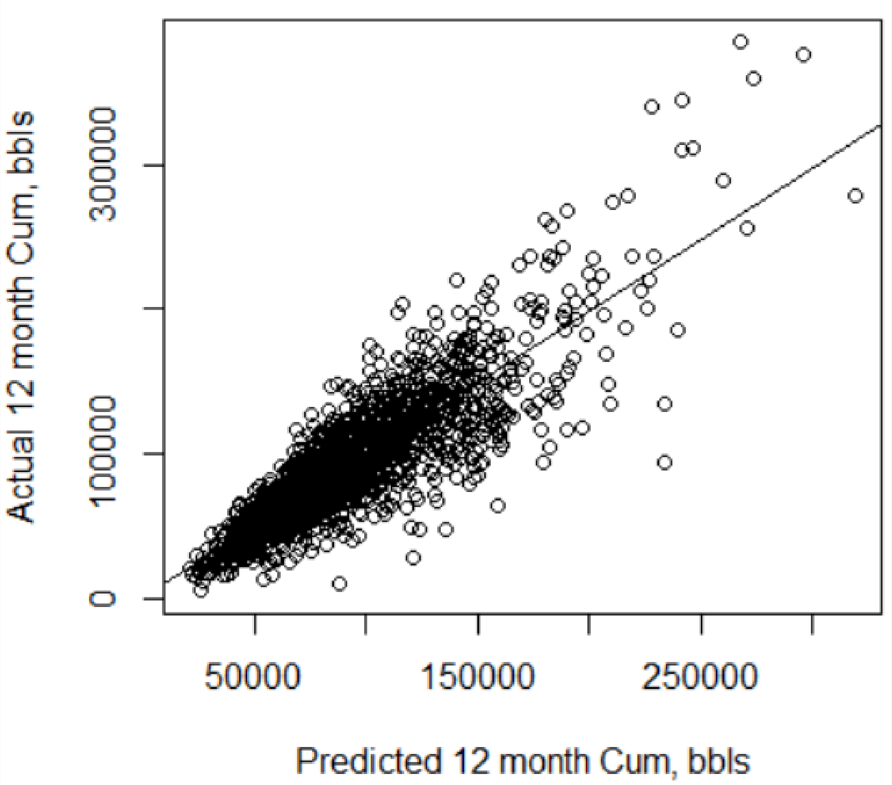
With the on-going growth in the world of “big data”, several data science techniques are currently being used to evaluate the performance of oil and gas wells. Among commonly used techniques are linear and multiple regression, multivariate analysis (includes PCA, cluster and factor analysis, etc), decision trees, and the family of AI, which includes neural network, deep learning, etc. Neural network regression model belongs to a class of functions called universal approximators, just like polynomial and Fourier functions. These functions can come really close to estimating the true mean function if correctly applied. NN is used to estimate the conditional mean functions that are highly non-linear. Hence, NN regression aims to approximate the true (unknown) regression function $f(x)$, using a general non-linear function, say $g(x)$. NN regression uses the logistic function $1/(1 + exp^{-x})$ as activation function. Iterative techniques that are similar to those used to obtain maximum likelihood, are used to estimate the parameters of the model.

Prediction of Well Performance using NN - One Node

Application of Neural Network to model First Year Production using one hidden layer with one node. NN uses numeric data for analysis, hence there is the need to subset our data to extract the variables that are important to our analysis. These variables are measured in different units, as a result, it is good practice to standardize the data. We would need to re-scale the final result back to the original scale.



```
## [1] "Fig. 5: Neural Network Model using one hidden layer"
```



```
## [1] "Fig. 6: NN Predicted vs Actual 12 month Oil - one hidden layer"

##
## Call:
## lm(formula = data$First12Oil[-idx] ~ pred_data)
##
## Residuals:
##         Min          1Q      Median          3Q         Max
## -137596.266  -11442.009    -312.973   11271.275  118957.347
##
## Coefficients:
##                  Estimate    Std. Error  t value            Pr(>|t|)
## (Intercept)  636.84827044 1396.26709117  0.45611             0.64837
## pred_data      0.98915592    0.01387732 71.27862 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22748.06 on 1725 degrees of freedom
## Multiple R-squared:  0.7465338,  Adjusted R-squared:  0.7463869
## F-statistic: 5080.642 on 1 and 1725 DF,  p-value: < 0.000000000000000222
```

Now let’s compute the predicted values “by hand” The aim of this is to extract the coefficients calculated by the NN model and explicitly specify an equation for the direct calculation of first 12 month cum oil $g(y|x)$.

Mathematical equations

Depending on the number of neurons/nodes in each hidden layer, we have as many logistic functions added together. To understand what goes on under the hood, from the plot of the model shown above, the NN uses one logistic function $1/(1+exp^{-x})$. With two nodes, we add a second logistic function and so on. $h_1(x)$ here, is a linear function of the nine variables in the model.

$$g(y|x) = \gamma_0 + \frac{\gamma_1}{1+exp^{-h_1(x)}} \ldots\ldots (1)$$

The result of the analysis above can be summarized in equation form as follows:

$$h_1(x) \\ = -0.5634 + 0.6692 * totalDepth - 0.2468 * TVD - 0.6309 * perfInterval - 0.0222 \\ * stage + 0.0083 * maxInjRate - 0.1389 * manInjPres + 0.1542 * totalProp + 0.1628 \\ * totalFluid + 2.1485 * peakOil \ldots\ldots (2)$$

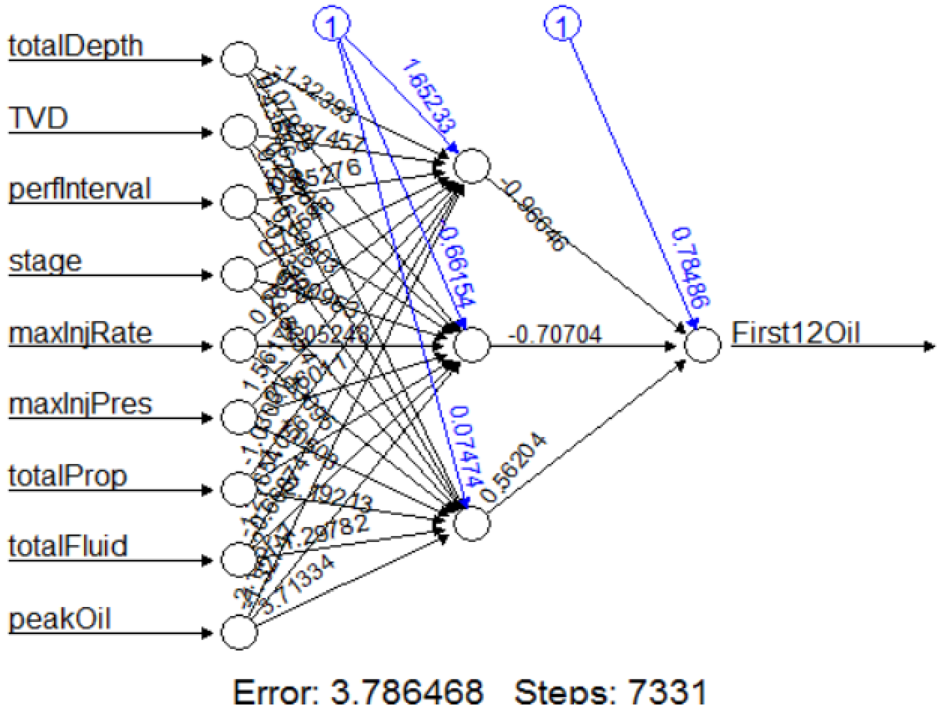
$$g(y|x) = -0.3885 + \frac{1.2471}{1+exp^{-h_1(x)}} \ldots\ldots (3)$$

Putting the NN model in equation form makes the picture shown in the plot clearer. The values shown on the connecting lines and to the left of the node are the coefficients/weights ($\beta's$) of the variables, while the values shown to the right of the node are the $\gamma's$

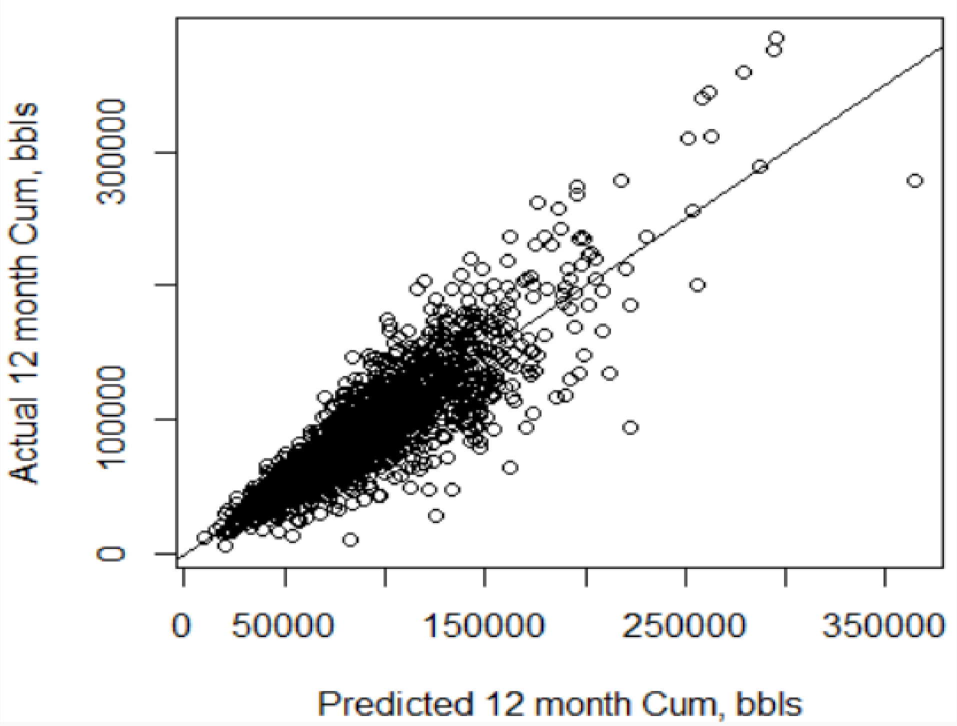
With two and three nodes, we add second and third logistic functions as shown below for three nodes. The linear functions $h_1(x)$, $h_2(x)$ and $h_3(x)$ are different functions.

$$g(y|x) = \gamma_0 + \frac{\gamma_1}{1+exp^{-h_1(x)}} + \frac{\gamma_2}{1+exp^{-h_2(x)}} + \frac{\gamma_3}{1+exp^{-h_3(x)}} \ldots\ldots (4)$$

Evaluation of Well Performance using NN - three nodes.



```
## [1] "Fig. 7: Neural Network Model using three hidden layers"
```



```
## [1] "Fig. 8: NN Predicted vs Actual 12 month Oil - three hidden layer"
```

```
##
## Call:
## lm(formula = data$First12Oil[-idx] ~ pred_data)
##
## Residuals:
##          Min          1Q      Median          3Q         Max
## -128787.896  -11210.393    -307.033   11049.186   88210.386
##
## Coefficients:
##                   Estimate    Std. Error  t value             Pr(>|t|)
## (Intercept) -494.13484315 1345.27850143 -0.36731              0.71343
## pred_data      1.00373574    0.01340248 74.89179 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21913.66 on 1725 degrees of freedom
## Multiple R-squared:  0.764787,   Adjusted R-squared:  0.7646507
## F-statistic:  5608.78 on 1 and 1725 DF,  p-value: < 0.00000000000000022204
```

Put the Solution in Mathematical form

- For node 1

$$h_1(x) = 1.6523 - 1.3239 * totalDepth + 0.8746 * TVD + 0.8528 * perfInterval + 0.7155 * stage + 0.2655 * maxInjRate + 1.5617 * manInjPres - 1.0303 * totalProp - 1.2765 * totalFluid - 2.1462 * peakOil \ldots\ldots(5)$$

- For node 2

$$h_2(x) = -0.6615 + 0.0793 * totalDepth + 0.2999 * TVD + 0.0390 * perfInterval + 0.2095 * stage - 1.0525 * maxInjRate + 0.0601 * manInjPres - 1.0560 * totalProp - 0.06687 * totalFluid - 1.3275 * peakOil \ldots\ldots(6)$$

- For node 3

$$h_3(x) = 0.0747 + 0.4366 * totalDepth + 0.5246 * TVD - 0.5329 * perfInterval + 0.8535 * stage - 1.1309 * maxInjRate + 1.0508 * manInjPres - 2.1921 * totalProp - 1.2978 * totalFluid + 3.7133 * peakOil \ldots\ldots(7)$$

- Putting it all together

$$g(y|x) = 0.7849 - \frac{0.9665}{1 + exp^{-h_1(x)}} - \frac{0.7070}{1 + exp^{-h_2(x)}} + \frac{0.5620}{1 + exp^{-h_3(x)}} \ldots\ldots\ldots\ldots(8)$$

These equations can get really complex if we consider several nodes or at worse, multiple hidden layers.

Conclusion

We have demonstrated that it is possible to get the mathematical forms of the result of neural network model output. It is also obvious why we tend to simply rely on the model to generate predicted values rather than have this good-looking but sometimes complicated NN models as shown in the equations above. First, it is simpler to just predict the values of the first 12 month oil using the `compute` function in `r`, rather than go through the lenghty process of extracting all the coefficients of the model as we have done. Second, since the model works with some random number generating process used for splitting the model into training and testing sets, there is a need to use `set.seed` function to force the model to pick the same random samples each time we run it. Otherwise, the model will use different sample sets every time it is run, and the equations generated above will be different each time. With `set.seed` the results of this article are also easily reproduced.

Github Page

This analysis can be viewed on my github repository. Readers can reproduce the work and play with the data as well. To view the plot of the neural network, you will need to run the `chunck` `import_data6` and `import_data8`. Goto githib

Packages used

- `rmarkdown`, lubridate`,knitr, and`neuralnet`.

Reference

Nordeng, S. H., & Helms, L. D. (2010). Bakken Source System–Three Forks Formation Assessment. NDGS

About the Author

Marshal Wigwe is a graduate student at Texas Tech University. He is currently working on his PHD in Reservoir Engineering, with focus in Big Data Analytics. His PHD Advisor is Dr. Marshall Watson, TTU PE Department Chair. He also serves on the board of SPE-PB Young Professionals as the Communications Chair and participates in several volunteer programs. Connect with him on LinkenIn