Springboard --- DSC

Capstone Project 3

Time Series Forecasting of Oil and Gas Production

By Jefferson Fernandez

April 2022

1. Introduction

The goal of this project is to develop Time Series Forecasting models that can predict future oil production of two wells. The first well follows the typical production profile an oil well and the second well does not. The machine learning models developed in this project are of interest to the Reservoir Engineers who are interested in finding out whether the well is profitable to keep or is more profitable to sell. The industry standard in predicting the production of an oil or a gas well is Decline Curve Analysis. This analysis is based on fitting different types of curves to the production rate. Machine learning unlike decline curve analysis gives as the option to try multiple types of algorithms from classical to deep learning algorithm and from univariate to multivariate analysis.

The best models for both wells are the ARIMA models. The best model for the F -14 well (Well that has a typical decline curve profile) is the ARIMA model with P = 2, d = 1 and q = 1 (34.4% MAPE). The best model for the F -15 well (Well that does not have a typical decline curve profile) is the ARIMA model with exogenous variables with P = 0, d = 0 and q = 1 (67.9%).

From: https://www.researchgate.net/figure/The-geographic-location-of-the-Volve-field_fig1_349397713

The Volve is a field in the central part of the North Sea, five kilometers north of the Sleipner Øst field. The water depth is 80 meters. Volve was discovered in 1993, and the plan for development and operation (PDO) was approved in 2005. The field was developed with a jack-up processing and drilling facility. The vessel "Navion Saga" was used for storing stabilized oil. Production started in 2008. Volve produced oil from sandstone of Middle Jurassic age in the Hugin Formation. The reservoir is at a depth of 2,700-3,100 meters. The western part of the structure is heavily faulted and communication across the faults is uncertain. The field was produced with water injection for pressure support. The oil was exported by tankers and the rich gas was transported to the Sleipner, a facility for further export. The field was shut down in 2016 and the facility was removed in 2018 (from: nosrkpetroleum.no).

2. Approach

2.1 Data Acquisition and Wrangling

The data from this project is from the open-licensed Volve data set made available by Equinor. The oil production data set from Volve is part of approximately 40,000 files from the Volve field, which was in production from 2008 to 2016. The data has been released to give students and scientists a realistic case to study in the field of petroleum Engineering. The data is from a csv file. Each data point is in daily format. Along with the daily oil production data there are 23 exogenous variables in the file such as Average downhole pressure, Average downhole temperature, Average Choke Size, Average Wellhead Pressure, Bore Gas Volume, Bore Water Volume, Average Downhole Tubing Pressure, etc.

```
df.columns
```

```
Index(['DATEPRD', 'WELL_BORE_CODE', 'NPD_WELL_BORE_CODE', 'NPD_WELL_BORE_NAME',
       'NPD_FIELD_CODE', 'NPD_FIELD_NAME', 'NPD_FACILITY_CODE',
       'NPD_FACILITY_NAME', 'ON_STREAM_HRS', 'AVG_DOWNHOLE_PRESSURE',
       'AVG_DOWNHOLE_TEMPERATURE', 'AVG_DP_TUBING', 'AVG_ANNULUS_PRESS',
       'AVG_CHOKE_SIZE_P', 'AVG_CHOKE_UOM', 'AVG_WHP_P', 'AVG_WHT_P',
       'DP_CHOKE_SIZE', 'BORE_OIL_VOL', 'BORE_GAS_VOL', 'BORE_WAT_VOL',
       'BORE_WI_VOL', 'FLOW_KIND', 'WELL_TYPE'],
      dtype='object')
```

Each well is associated with a unique well bore code, and well bore name. Since All 7 wells are in located in the same field therefore, they have the same facility code, facility name and field code. There are a total of 15,634 records in the dataset. Examining the dataset, we found out that the features are in datetime, object, integer, and float data types. Since we are working on a time series project, we are interested in transforming every relevant feature as numerical, drop irrelevant categorical features and use the production date as the index and transform it as date time data type.

```
df.dtypes
```

```
DATEPRD                        datetime64[ns]
WELL_BORE_CODE                         object
NPD_WELL_BORE_CODE                      int64
NPD_WELL_BORE_NAME                     object
NPD_FIELD_CODE                          int64
NPD_FIELD_NAME                         object
NPD_FACILITY_CODE                       int64
NPD_FACILITY_NAME                      object
ON_STREAM_HRS                         float64
AVG_DOWNHOLE_PRESSURE                 float64
AVG_DOWNHOLE_TEMPERATURE              float64
AVG_DP_TUBING                         float64
AVG_ANNULUS_PRESS                     float64
AVG_CHOKE_SIZE_P                      float64
AVG_CHOKE_UOM                          object
AVG_WHP_P                             float64
AVG_WHT_P                             float64
DP_CHOKE_SIZE                         float64
BORE_OIL_VOL                          float64
BORE_GAS_VOL                          float64
BORE_WAT_VOL                          float64
BORE_WI_VOL                           float64
FLOW_KIND                              object
WELL_TYPE                              object
dtype: object
```

After transforming the features into relevant data types, we are now interested in finding out if there are missing days in the data set. It is important that we do not have any missing dates since the time series algorithms would lead to incorrect predictions. Using a for lop we found out that 5 wells have missing production data. We now must replace the NaN values. We replaced the NaN values for the Volume features with 0 and the rest of the features with the average value for that feature for that well.

```
dic={}
counter=0
for well in wells:
    date_set = set(well[0] + timedelta(x) for x in range((well[-1] - well[0]).days))
    missing = sorted(date_set - set(well))
    dic[well_name[counter]] = missing
    counter+=1
```

Fig: for loop to determine the missing days in the production data

```
dic
```

```
{'F_1': [],
 'F_11': [Timestamp('2013-09-19 00:00:00'),
  Timestamp('2014-01-30 00:00:00'),
  Timestamp('2014-02-07 00:00:00')],
 'F_12': [Timestamp('2008-04-28 00:00:00'),
  Timestamp('2008-08-07 00:00:00'),
  Timestamp('2009-05-05 00:00:00'),
  Timestamp('2009-08-03 00:00:00'),
  Timestamp('2009-09-01 00:00:00'),
  Timestamp('2010-02-28 00:00:00'),
  Timestamp('2010-03-01 00:00:00'),
  Timestamp('2010-03-07 00:00:00'),
  Timestamp('2010-08-16 00:00:00'),
  Timestamp('2010-08-18 00:00:00'),
  Timestamp('2010-08-19 00:00:00'),
  Timestamp('2010-08-24 00:00:00'),
  Timestamp('2011-02-02 00:00:00'),
  Timestamp('2011-02-03 00:00:00'),
  Timestamp('2011-02-04 00:00:00'),
```

Fig: Dictionary that contains all the missing dates.

NPD_WELL_BORE_NAME	AVG_DOWNHOLE_PRESSURE	AVG_DOWNHOLE_TEMPERATURE	AVG_DP_TUBING
15/9-F-1 C	246.666036	104.925303	192.540223
15/9-F-11	233.962353	104.332832	182.861664
15/9-F-12	80.729069	33.292082	84.765923
15/9-F-14	233.074651	95.133791	192.653088
15/9-F-15 D	226.034939	104.645505	186.151842
15/9-F-4	0.000000	0.000000	0.000000
15/9-F-5	0.000000	0.000000	0.000000

Fig: Average value for each feature for each well

At the end of our data wrangling exercise, we produced a data frame with no missing dates, no NAN values, and contains only the relevant features that we included in the modelling process and includes all 7 wells.

```
df.dtypes
```

```
DATEPRD                     datetime64[ns]
NPD_WELL_BORE_NAME                  object
AVG_DOWNHOLE_PRESSURE              float64
AVG_DOWNHOLE_TEMPERATURE           float64
AVG_DP_TUBING                      float64
AVG_ANNULUS_PRESS                  float64
AVG_CHOKE_SIZE_P                   float64
AVG_WHP_P                          float64
AVG_WHT_P                          float64
DP_CHOKE_SIZE                      float64
BORE_OIL_VOL                       float64
BORE_GAS_VOL                       float64
BORE_WAT_VOL                       float64
BORE_WI_VOL                        float64
dtype: object
```

Fig: In the final Dataframe all numerical data are of the same data type.

2.2 Storytelling and Inferential Statistics

Plotting the daily oil production shows that the production data is erratic and will make it hard for the algorithm to make predictions and thus the need to down sample the data set from daily to weekly production. Unfortunately, even after down sampling the data set to weekly production the data was still erratic and as a result, we finally down sampled the data to monthly.

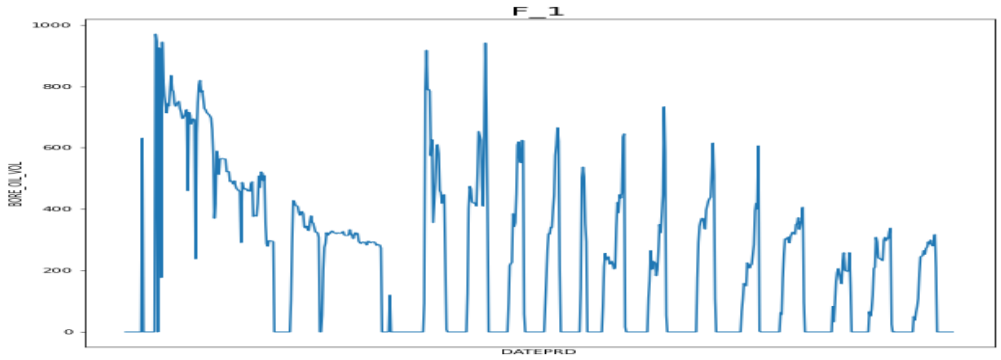

Fig: Daily Oil Production F_1 Well

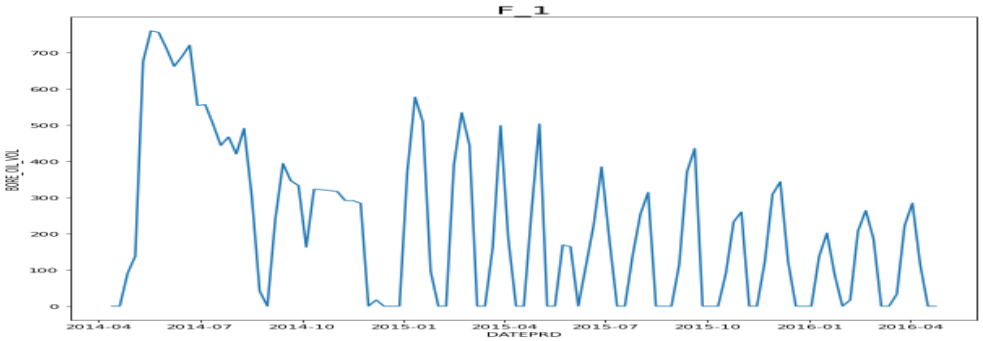

Fig: Weekly Oil Production F_1 Well

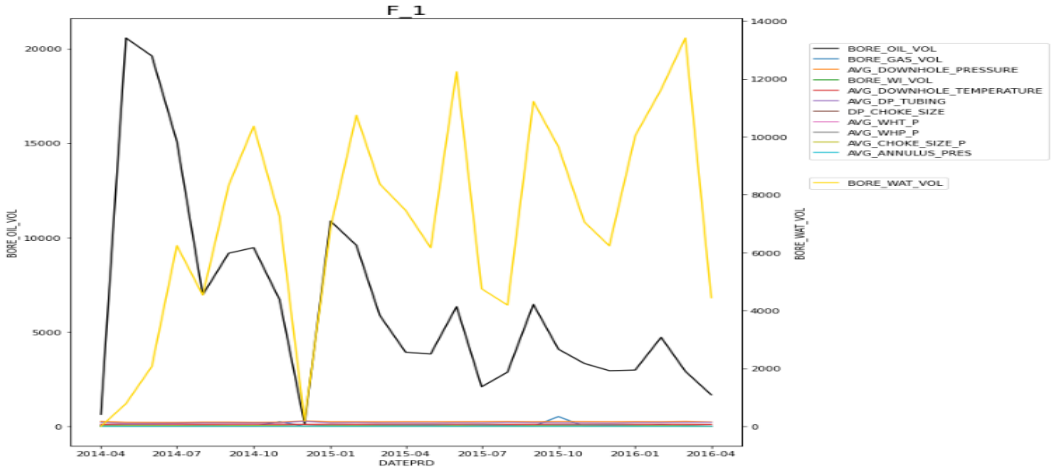

Fig: Monthly oil Production F_1 Well

Plotting the oil productions for each well shows that the F_4 well did not produce any oil but served as a water injection well on the other hand the F_5 well produced oil only for a short period of time and served as a water injection well for most of its life. The wells did not come online at the same dates. Since we are not interested in forecasting the production of wells that served as water injections wells, we ended up dropping these wells from the data frame which are the F_4 and F_5 wells.

Another important observation from the heatmaps is the fact that the correlations between the variables and the wells are not the same. For example, there is a positive correlation between downhole pressure for some wells and negative for the rest of the wells. For some wells there is a stronger correlation between the target variable and the lag variable and for the rest of the well the correlation is stronger between the target variable and the original variables.

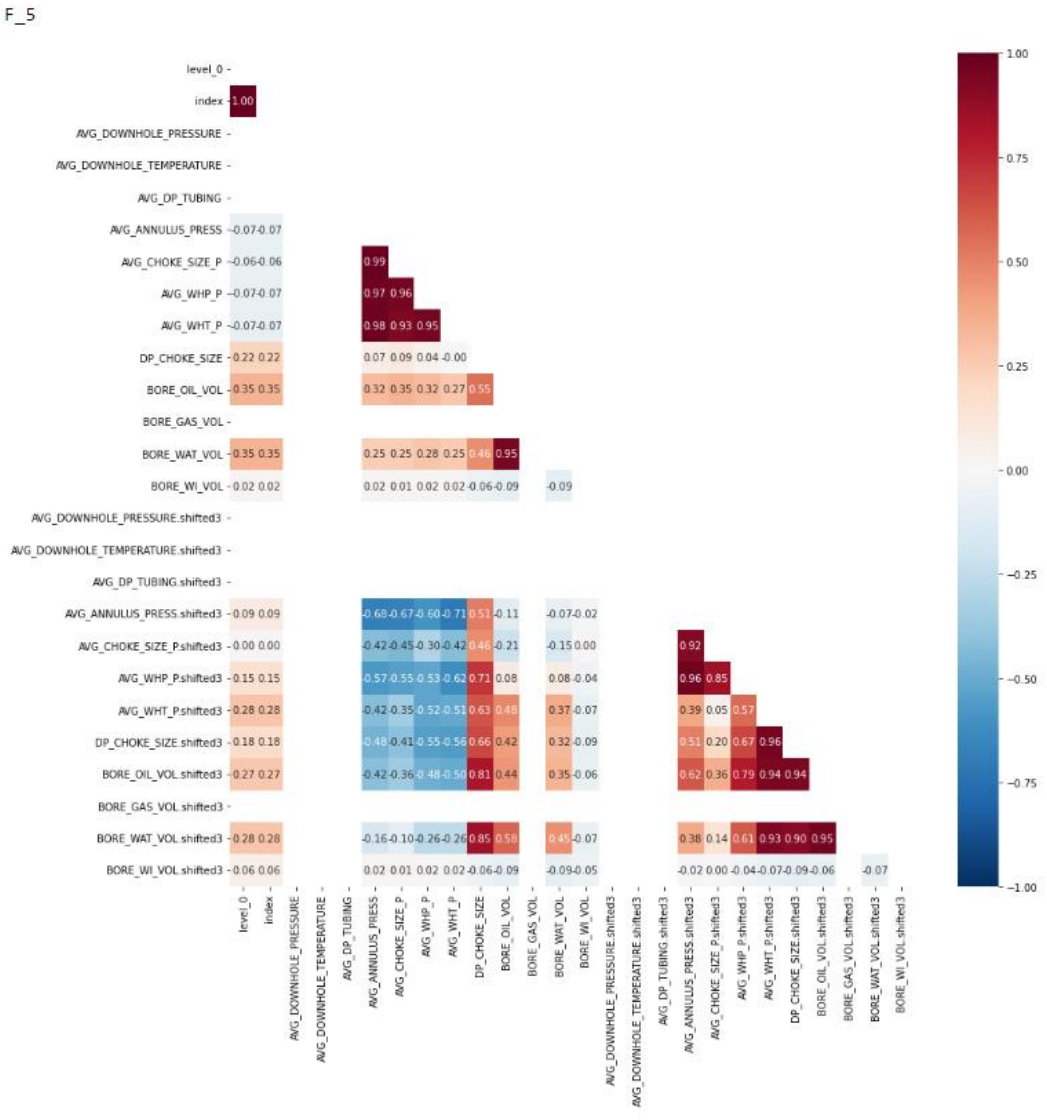

Fig: Heat Map Correlation F_5 Well

2.3 Base Modelling – Classical Forecasting

We began our modelling by picking our two wells that we picked based on their production profiles. The F_14 well is a well that follows the production profile of a typical well on the other hand the F4_15 well is the second well that we picked because it does not follow a typical decline curve.

The classical time series algorithms we employed in this project are Autoregressive models, Moving Average Models, Autoregressive Integrated Moving Average and Autoregressive Integrated Moving Average with Exogeneous Variables. We select the model for each well based on the lowest Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). The AIC can be termed as a measure of the goodness of fit of any estimated statistical model. The BIC is a type of model selection among a class of parametric models with different numbers of parameters.

The metric that we used to calculate the performing model is the Mean absolute percentage error. MAPE is a measure of prediction accuracy of a forecasting method in statistics. MAPE is commonly used as a loss function for regression problems and in model evaluation, because of its very intuitive interpretation in terms of relative error.

	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
AR_order_2_difference_2	13.148332	15.850624	11.989556	103.963873	36.238096
AR_order_6_difference_2	21.022414	23.479713	19.968694	85.472249	37.485768
MA_order_4_difference_2	54.757805	49.942689	56.822585	263.435749	106.239707
ARIMA_p_order_6_difference_2_q_order_0	11.374029	14.131526	10.191580	108.130674	35.956952
ARIMAX_p_order_1_difference_2_q_order_1	12.106721	14.841422	10.934048	106.410007	36.073050

Fig: F_14 Classical Models Performance

	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
AR_order_2_difference_2	4.461194	1.102241	36.980994	473.401118	128.986387
AR_order_0_difference_0	0.170149	3.380181	30.907868	447.979071	120.609317
MA_order_1_difference_0	4.816648	1.446266	37.447105	475.352255	129.765569
MA_order_3_difference_0	4.212378	0.861426	36.654719	472.035334	128.440964
ARIMA_p_order_0_difference_0_q_order_2	66.759857	67.828695	56.411873	82.459479	68.364976
ARIMAX_p_order_0_difference_0_q_order_1	67.440455	68.487408	57.304347	78.723592	67.988950

Fig: F_15 Classical Models Performance

The best models for both wells are the ARIMAX models and as expected the simple models performed significantly worse than the ARIMAX models. Also as expected the F-14 well which followed a typical decline curve was easier to model than the F-15 model which did not follow a typical decline curve (35% MAPE vs 65% MAPE for the ARIMAX Models).

2.4 Extended Modelling – Advanced Packages & Deep Learning

In this section of the project, we used Advanced Packages and Deep Learning Methods specifically developed for time series forecasting. For the advanced packages we used Auto-ARIMA,KATS, KATS, SKTIME & Silverkite. For Deep Learning we used Tensorflow/Keras to develop Fully Connected Neural Networks, Recurrent Neural Network and Convolutional Neural Network.

The purpose of using these packages and frameworks is to go beyond classical forecasting and more importantly make better prediction in the context of lower MAPE scores. One big advantage of using these special tools is to enable us to create prediction without having to pre-process the time series before fitting, for example we there is no need to do differencing with these packages.

The Auto-ARIMA package allows us to reduce the work in finding the optimal ARIMA model by allowing us to go straight to modelling without first differencing the model to eliminate the non-stationarity in the time series and plotting the PACF and ACF. The Auto-ARIMA package automatically generates the optimal p, d, and q values by giving the model a range of p, d, and q to evaluate. The best model is then chosen with the lowest AIC or BI score.

The SKTIME package provides an easy-to-use, flexible, and modular open-source framework for a wide range of time series machine learning tasks. It offers scikit-learn compatible interfaces and model composition tools, with the goal to make the ecosystem more usable and interoperable. SKTIME features a unified interface for multiple time series learning tasks. It supports forecasting, time series classification and time series regression. The sktime.forecasting module contains algorithms and composition tools for forecasting. In this project we used the Theta model, TBATS model and Polynomial Trend. Each model has its own strengths and weaknesses, and we used all three algorithms to show the flexibility of the SKTIME package.

KATS which stand for Kits for analyzing time series is a toolkit to analyze time series data, a lightweight easy-to-use, and generalizable framework to perform time series

analysis that is being developed by Facebook Time series analysis is an essential component of Data Science and Engineering at industry, from understanding the key statistics an characteristics, detecting regressions and anomalies, to forecasting future trends. KATS aims to provide the one-stop shop for time series analysis, including detection, forecasting, feature extraction/embedding, multivariate analysis, etc. In this project we implemented the ARIMA, fbProphet, Theta, Harmonic Regression, LSTM, and Ensemble algorithms, each of which has different strength and weaknesses, and we used all these algortihms to show the flexibility of the fbProphet framework.

The Silverkite algorithm is part of the Greykite library, an open-source Python library developed to support LinkedIn's forecasting needs. Silverkite is its main forecasting algorithm. It is fast, accurate, and intuitive, making it suitable for interactive and automated forecasting at scale. The Silverkite algorithm works well on time series with (potentially time-varying) trends and seasonality, repeated events/holidays, and/or short-range effects.

Tensorflow is the most popular deep learning package available coupled with Keras which makes deep learning more accessible by making it easier to build deep learning models. In this project we developed 3 different neural network models which are the following 1. Multi-level perceptron 2. Long Short-Term Memory 3. Convolutional Network. An MLP model is a fully connected class of feedforward artificial neural network.

The term MLP is used ambiguously, sometimes loosely to mean any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons(with threshold activation. Multilayer perceptron's are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer. The MLPs can be used for time series forecasting by taking multiple observations at prior time steps, called lag observations, and using them as input features and predicting one or more time-steps from those observations. In this notebook we created MLP models with the following characteristics.

- 3 layers
- Rectified Linear Units as the activation function for all layers.
- Mean squared error as the loss function
- Adam optimization algorithm as the optimizer

Long short-term memory or LSTM is an artificial recurrent neural network (RNN) architecture that unlike standard feedforward neural networks has feedback connection/s. LSTMS are well-suited to classifying, processing, and making predictions based on time series data. The LSTM has an internal memory allowing it to accumulate

internal state as it reads across the steps of given input sequence. In this notebook we created LSTM models with the following characteristics

- 3 layers
- Rectified Linear Units as the activation function for all layers.
- Mean squared error as the loss function
- Adam optimization algorithm as the optimizer

3. Findings

	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
AR_order_2_difference_2	13.148332	15.850624	11.989556	103.963873	36.238096
AR_order_6_difference_2	21.022414	23.479713	19.968694	85.472249	37.485768
MA_order_4_difference_2	54.757805	49.942689	56.822585	263.435749	106.239707
ARIMA_p_order_6_difference_2_q_order_0	11.374029	14.131526	10.191580	108.130674	35.956952
ARIMAX_p_order_1_difference_2_q_order_1	12.106721	14.841422	10.934048	106.410007	36.073050

Fig. Base Models Performance for F – 14 Well

	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
AR_order_2_difference_2	4.461194	1.102241	36.980994	473.401118	128.986387
AR_order_0_difference_0	0.170149	3.380181	30.907868	447.979071	120.609317
MA_order_1_difference_0	4.816648	1.446266	37.447105	475.352255	129.765569
MA_order_3_difference_0	4.212378	0.861426	36.654719	472.035334	128.440964
ARIMA_p_order_0_difference_0_q_order_2	66.759857	67.828695	56.411873	82.459479	68.364976
ARIMAX_p_order_0_difference_0_q_order_2	67.440455	68.487408	57.304347	78.723592	67.988950

Fig. Base Models Performance for F-15 Well

	Model	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
0	Auto-ARIMA	1.510069	4.574472	0.196015	131.295359	34.393979
1	KATS-ARIMAX	19.059558	15.355154	20.648052	179.601406	58.666043
2	KATS-fbProphet	220.715370	210.736671	224.994358	653.173199	327.404899
3	KATS-Theta	5.773030	2.482021	7.184255	148.399106	40.959603
4	KATS-Quadratic	17.869888	20.425275	16.774107	92.875692	36.986241
5	KATS-LSTM	19.047857	21.566592	17.967793	90.109331	37.172893
6	KATS-Ensemble	6.048429	8.971627	4.794926	120.637399	35.113095
7	SKTIME-ThetaForecaster	0.747198	3.835338	0.577034	133.086896	34.561616
8	SKTIME-TBATS	23.689575	19.841113	25.339843	190.474613	64.836286
9	SKTIME-PolynomialTrend	11.910353	14.651163	10.735060	106.871162	36.041935
10	Silverkite	9.421965	6.017424	10.881874	156.968326	45.822397

Fig. Specialized Packages Models Performance for F -14 Well

	Model	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
0	Auto-ARIMA	4.816648	1.446266	37.447105	475.352254	129.765569
1	KATS-ARIMAX	13.906891	16.675217	12.894743	372.576303	104.013289
2	KATS-fbProphet	21.227239	23.760179	3.295499	332.393961	95.169219
3	KATS-Theta	10.098965	12.989735	17.888115	393.478508	108.613831
4	KATS-Quadratic	22.532051	18.592030	60.677488	572.594413	168.598996
5	KATS-LSTM	79.468521	73.697706	135.338843	885.126122	293.407798
6	KATS-Ensemble	15.663102	18.374957	10.591807	362.936234	101.891525
7	SKTIME-ThetaForecaster	15.289324	18.013198	11.081945	364.987950	102.343104
8	SKTIME-TBATS	1.690734	4.851871	28.913911	439.632385	118.772225
9	SKTIME-PolynomialTrend	23.111286	19.152640	61.437044	575.773908	169.868720
10	Silverkite	4.815842	1.445486	37.446047	475.347826	129.763800

Fig. Specialized Packages Models Performance for F -15 Well

	Model	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
0	DNN	6.614768	3.297570	8.037224	150.375859	42.081355
1	LSTM	11.385554	7.919917	12.871661	161.579649	48.439195
2	CNN	6.721903	3.401371	8.145788	150.627456	42.224130

Fig. Deep Learning Packages Models Performance for F -14 Well

	Model	2016-04-01 00:00:00	2016-05-01 00:00:00	2016-06-01 00:00:00	2016-07-01 00:00:00	MAPE
0	DNN	9.534387	12.443311	18.628451	396.577549	109.295925
1	LSTM	39.626016	41.567343	20.830999	231.400671	83.356258
2	CNN	13.171067	15.963054	13.859636	376.615338	104.902274

Fig. Deep Learning Packages Models Performance for F -14 Well

The best models for both wells are the ARIMAX models and as expected the simple models performed significantly worse than the ARIMAX models. Also as expected the F-14 well which followed a typical decline curve was easier to model than the F-15 model which did not follow a typical decline curve (35% MAPE vs 65% MAPE for the ARIMAX Models).

Our project shows that when it comes to time series forecasting, classical time series forecasting models can match and outperform the specialized time series packages and deep learning models, as can be seen from our models it is the last month for both wells that is significantly throwing off our MAPE scores. Machine learning gives the petroleum engineer an additional tool in tackling the problem of determining oil and gas production.

3. Conclusions and Future Work

The industry standard in determining the Estimated Ultimate Recovery of a well is based is what is called Decline Curve Analysis. DCA is a very involved process which requires the Engineer to know certain parameters about the well. Decline Curve Analysis has several shortcomings, including that it can underestimate oil reserves, underestimate production rates, and overestimate performance. With machine learning we could speed up the process of determining EUR for hundreds of wells without much pre-processing work that DCA requires and without need a lot of a priori knowledge about the geology and other factors about the wells.

4. Recommendations for Clients

The Petroleum Engineer should consider using machine learning in the determination of EUR not only because it requires less work, but the results are comparable and exceeds the performance of DCA. Another important advantage of using machine learning over DCA is that the latter usually requires proprietary software that cost tens of thousands of dollars in licensing fees whereas python is free to use. Here are our recommendations to further improve the MAPE score for future EUR forecasting.

1. Apply machine learning in EUR determination for wells with daily production data

2. Collect relevant exogenous variables such as downhole pressure, choke size, etc. to further improve MAPE scores.

3. Explore the possibility of using machine learning in determining field wide oil production forecasting.

5. Consulted Resources

The following resources for this project.

- https://machinelearningmastery.com/
- Springboard Materials
- Machine Learning Pocket Reference by Matt Harrison
- Hands-On Machine Leaning with Scikit-Learn Keras and TensorFlow
- Practical Statistics for Data Scientist by Bruce and Bruce
- https://www.tensorflow.org/
- https://facebookresearch.github.io/Kats/
- https://sktime.org/
- https://www.equinor.com/energy/volve-data-sharing
- Forecasting: Principles and Practice
- https://www.equinor.com/energy/volve-data-sharing