Baker Hughes México Data Science Hackathon 2022

Predicting modeling Challenge

Predictive team A

March 2022

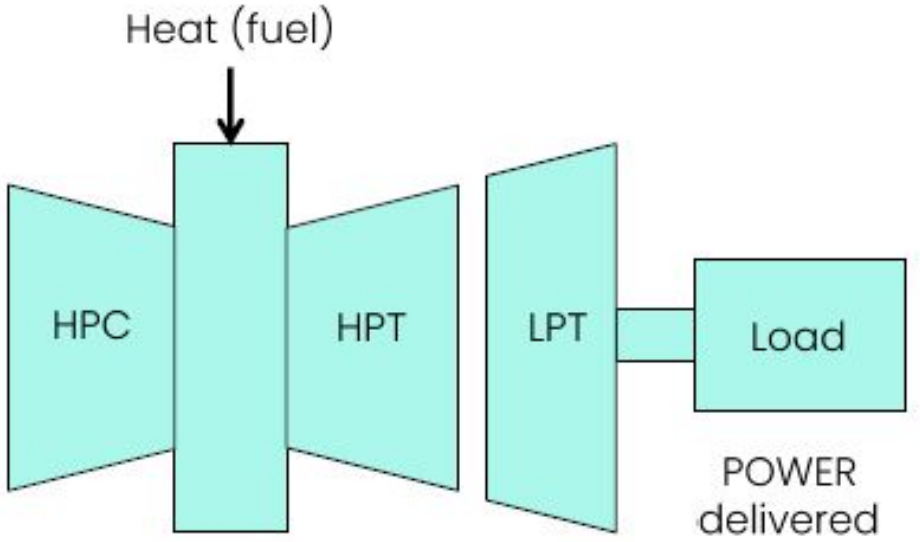
Executive summary

- Exploratory data analysis
- Model selection by Root Mean Square Mean
- Linear regression model
- Model behaviour as expected
- Assumptions of variables made with the linear regression model.
- Code & Results

Problem description

- In this competition we had to be able to develop a model for an Aeroderivative Gas Turbine to predict the POWER (kW) output from the Low Pressure Turbine (LPT).

- For this porpoise we handled synthetic data that simulates the behavior of Gas Turbine engines based at different locations worldwide.



Exploratory data analysis

- **General data overview**
  - Unusual 0 values
  - NaN values

```
In [70]:     .
Out[70]:
       T_AMB     P_AMB    CMP_SPEED  ...         RH       WAR          POWER
0   1.450440  0.843522     0.000000  ...  81.237441  0.000041            NaN
1   2.761142  0.843856  7870.729713  ...  74.311313  0.000041  13332.692409
2   9.270325  0.843413  9898.625866  ...  47.897182  0.000041  13026.684965
3  14.293265  0.844249  9850.791469  ...  34.400729  0.000041  12773.507042
4  12.875213  0.843663  9828.508458  ...  37.537882  0.000041  12768.092781

[5 rows x 12 columns]
```

- First, we guest it was a sensor's problem.
- Then we realize that the zeros correspond to the variable CMP_SPEED and the NaN values where allocated to the turbine's variables.
- Showing that in those days the turbine was put-off

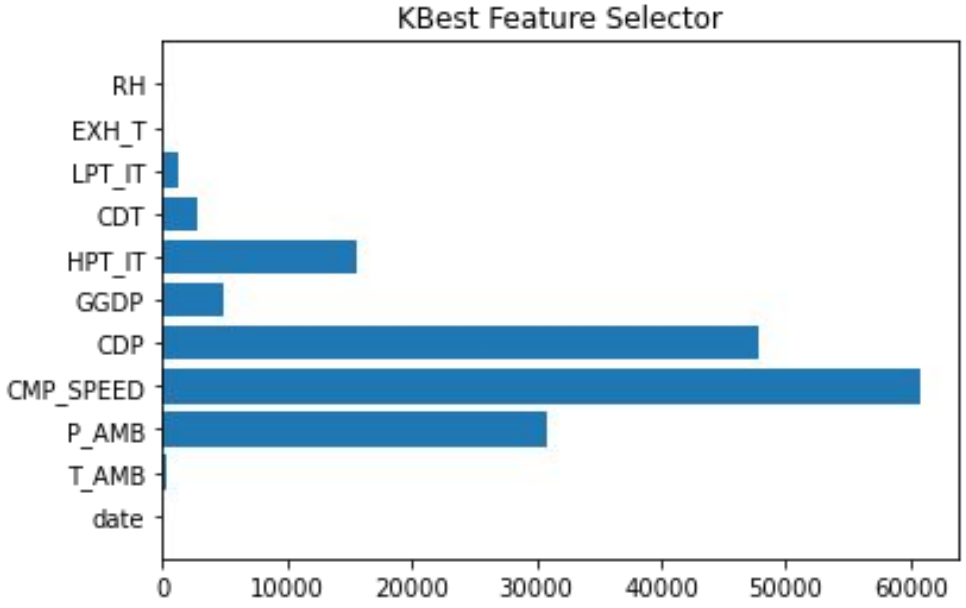
Data preprocessing and feature engineering techniques

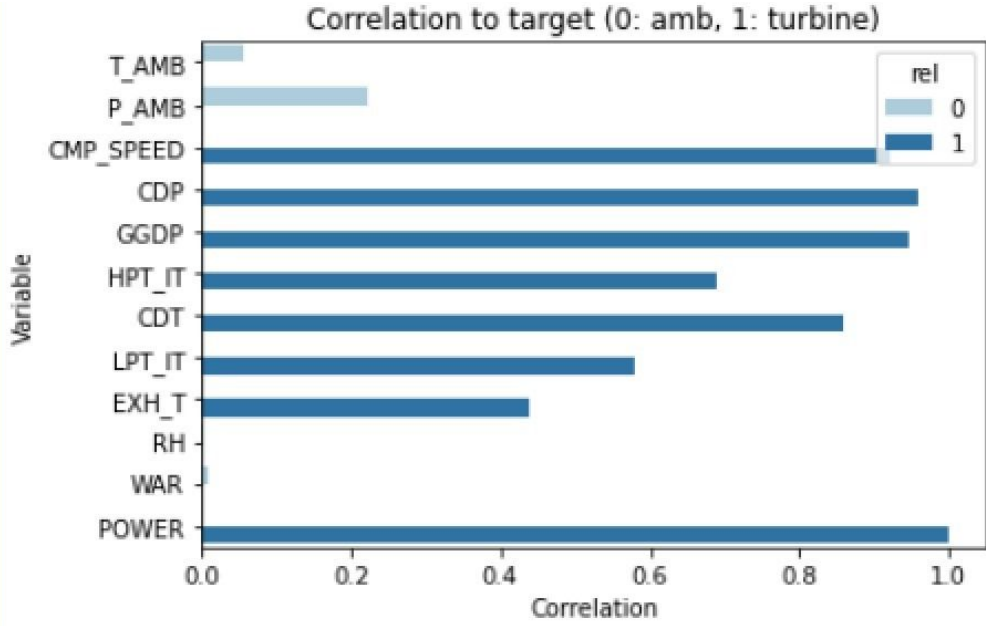
- **Handling missing values**
  - Delete the rows with missing values
  - Replace the missing values with the statistics mean
  - Replace the missing values with zeros

```
# Remove all nan values
df.dropna(inplace = True)
df_test.dropna(inplace = True)

# Replace with mean
for col in df.columns:
    if df[col].dtype == 'float':
        df[col].fillna(df[col].mean(), inplace = True)
for col in df_test.columns:
    if df_test[col].dtype == 'float':
        df_test[col].fillna(df_test[col].mean(), inplace = True)


# Handling missing values with zero
df.fillna(0, inplace = True)
df_test.fillna(0, inplace = True)
```

Data preprocessing and feature engineering techniques

- **Determinate hyperparameters.**





CMP_SPEED, CDP, GGDP, HPT_IT, CDT, LPT_IT, EXH_T.

Selection of feature variables

- CMP_SPEED, CDP, GGDP, HPT_IT, CDT, LPT_IT, EXH_T.

- CMP_SPEED - compressor speed in RPM.
- CDP - compressor discharge pressure, barA.
- CDT - compressor discharge temperature, degC. —
- GGDP - gas generator discharge pressure, barA.
- HPT_IT - High Pressure Turbine (HPT) inlet temperature, degC.
- LPT_IT - Low Pressure Turbine (LPT) inlet temperature, degC.
- EXH_T - exhaust temperature, degC.

Model comparison and model selection techniques

- Based in Root Mean Square Error parameters applied to the test dataset (subset).
- Logistic regression
- Linear regression
- Ridge regression

Model Selection - Linear Regression

Actual vs Predicted data on training dataset

date	Actual	Predicted
2021-07-18	0.000000	-51.594492
2021-10-02	5283.494881	5249.715082
2021-03-09	10749.132285	11160.228757
2021-10-04	17166.465997	16250.215063
2021-05-29	3103.702759	3376.599212

Actual vs Predicted data on test splitted data

date	actual	predicted
2021-10-28	10757.528311	10579.359239
2021-02-03	11769.494848	12475.357093
2021-01-09	12818.381578	13487.561463
2021-04-10	7040.412363	7128.064488
2021-02-20	5584.987785	5620.507935

*Method 1

Model Selection - Linear Regression

Actual vs Predicted data on training dataset

date	Actual	Predicted
2021-07-18	0.000000	-31.763825
2021-10-02	5283.494881	5308.745030
2021-03-09	10749.132285	11233.155887
2021-10-04	17166.465997	16976.987363
2021-05-29	3103.702759	2194.332666

Actual vs Predicted data on test splitted data

date	actual	predicted
2021-10-28	10757.528311	10733.030062
2021-02-03	11769.494848	12433.462689
2021-01-09	12818.381578	13489.899869
2021-04-10	7040.412363	7393.157403
2021-02-20	5584.987785	5647.803598

*Method 2

RMSE For Model Evaluation

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

```
rmse_train = (np.sqrt(mean_squared_error(target_train, target_train_predict)))
rmse_test = (np.sqrt(mean_squared_error(target_test, target_test_predict)))

print("RMSE for training data: {:.4f}".format(rmse_train))
print("RMSE for test splitted data: {:.4f}".format(rmse_test))
```

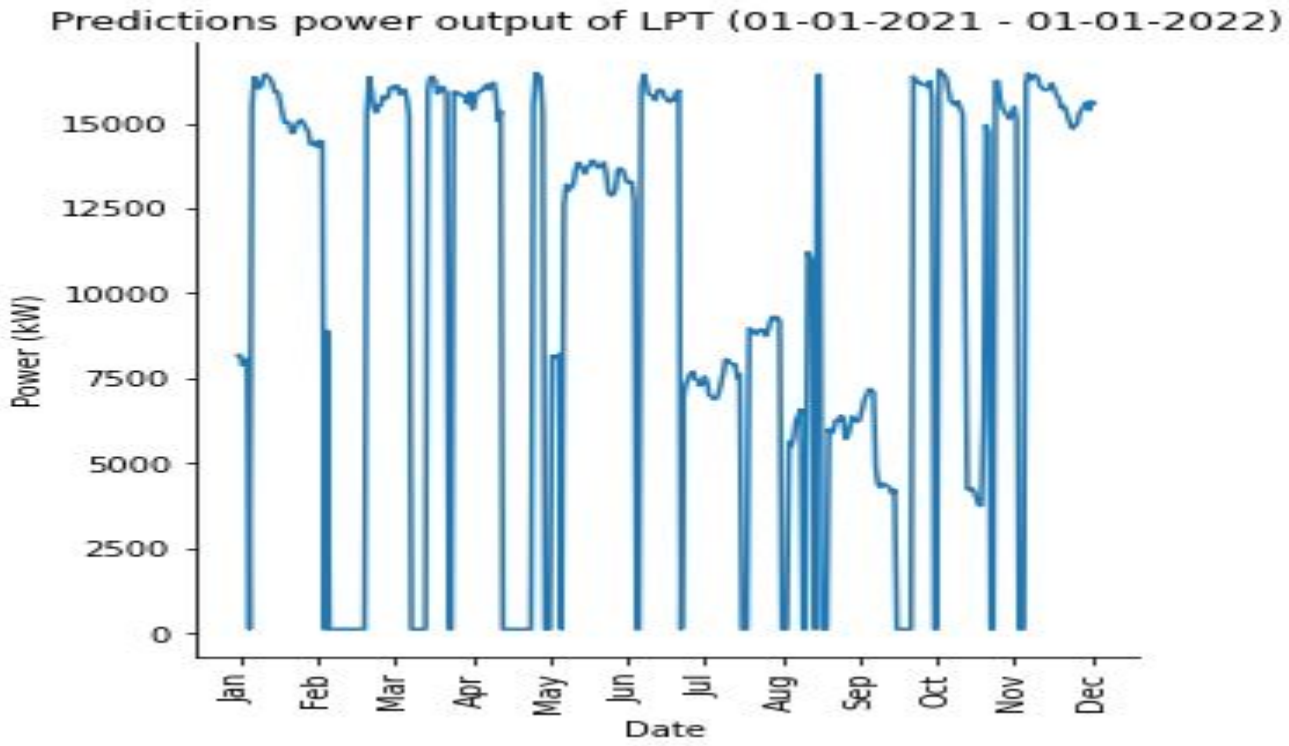
```
RMSE for training data: 354.4378
RMSE for test splitted data: 363.4620
```

Variables Selected by Method 1

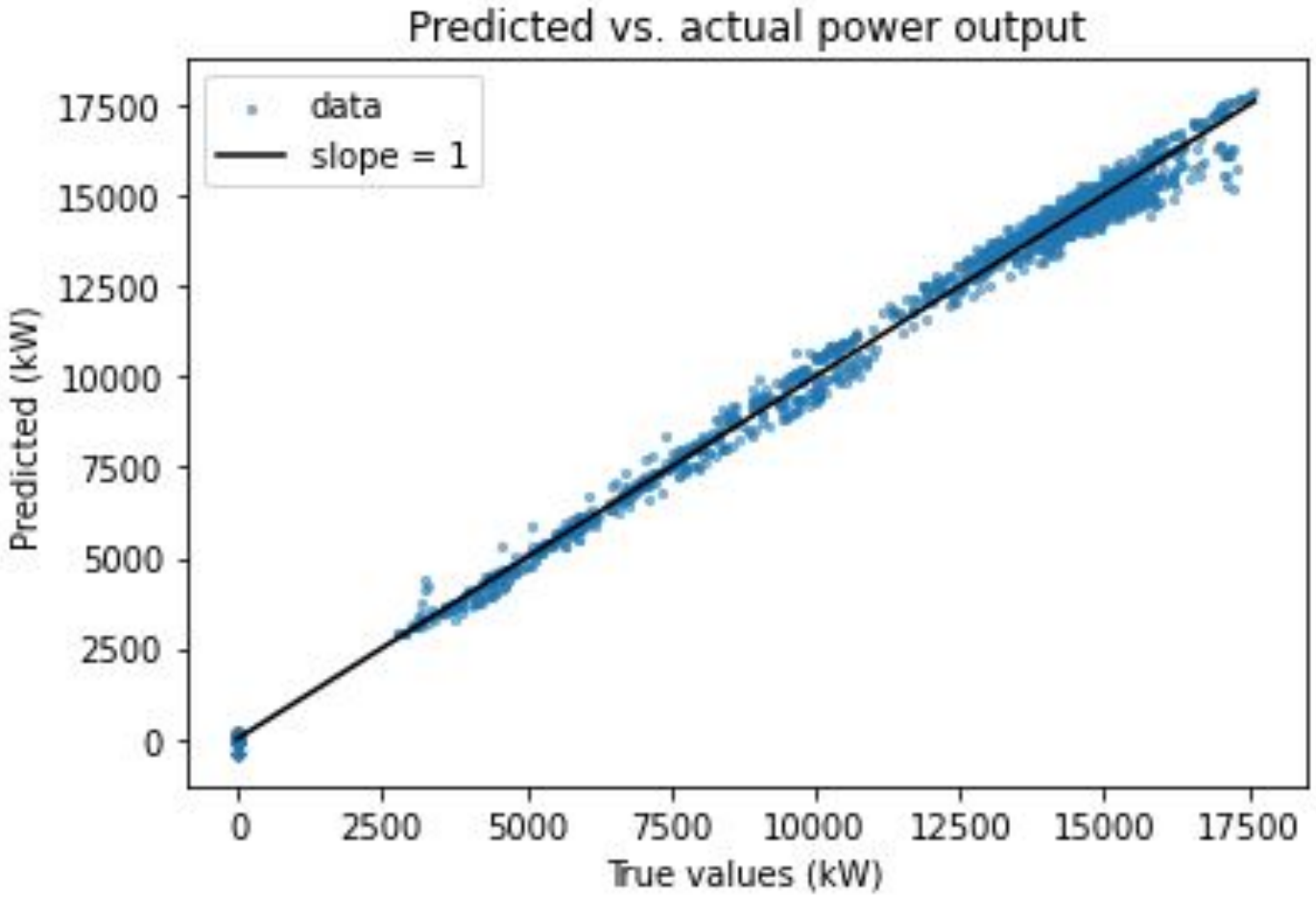
```
RMSE for training data: 457.8888
RMSE for test splitted data: 482.0964
```

Variables Selected by Method 2

Results


Predictions power output of LPT (01-01-2021 - 01-01-2022)
Power (kW)
15000
12500
10000
7500
5000
2500
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Date

Expected vs. predictions



Model coefficients and model eval.

- Interception → -3080.509774297314
- Coefficients → -0.78558, 189.443, 3008, 260.384, -562.218, 306.611
- Independent variables → CMP_SPEED, CDP, GGDP, HPT_IT, LPT_IT, EXH_T
- POWER = -3080.509774297314 - 0.78558CMP_SPEED + 189.443CDP + 3008GGDP + 260.384HPT_IT - 562.218LPT_IT + 306.611EXH_T

R2: 0.9965

References

- Pathak, M. (2018). Joining DataFrames in Pandas Tutorial. 12/03/22, de Datacamp. Sitio web: https://www.datacamp.com/community/tutorials/joining-dataframes-pandas
- Ashok, P. (2020). What is ridge Regression?. 12/03/22, de Great Learning. Sitio Web: Ridge Regression Definition & Examples | What is Ridge Regression? (mygreatlearning.com)
- Ohri, A. (2017). 10 Popular Regression Algorithms In Machine Learning Of 2022. 12/03/22, de Jigsaw. Sitio Web: 10 Popular Regression Algorithms In Machine Learning Of 2022 (jigsawacademy.com)
- Meghna, P. (2021). Pandas dropna() - Drop Null/NA Values from DataFrame. 12/02/22, JournalDev. Sitio Web: Pandas dropna() - Drop Null/NA Values from DataFrame - JournalDev
- Pandas. Versión 1.4.1. Sitio web: pandas documentation — pandas 1.4.1 documentation (pydata.org)
- Scikit. Versión 1.0.2. Sitio web: scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation

Q&A SESSION

Personal experience