# Aurubis Buffalo
# Waiting Time Analysis
# For Coil Production

**-Group 17**

Xinyu Huang, Yuqi Hou, Li Zeng,
Aijing Song, Fanglei Su

Aurubis
Metals for Progress

Aurubis

# Content

## Data Cleaning & Handling

Dealing with Missing Values
Calculating Waiting Time

## Exploratory Data Analysis

Attributes Data Visualization
Sanky Diagram

## Waiting Time Model

Feature Engineering
Model Building:

- Random Forest
- Neural Network
- Linear Regression
- Bayesian Ridge
- Light GBM
- K Nearest Neighbor

Model Selection

## Conclusion

# Part1: Data Cleaning

- Drop Unrelated Attributes with Waiting Time Model
  - INMINL(Minor Location), INSTRN(Instruction)…
- Missing Value
  - Drop the attribute which have more than 50% missing value
  - Other: Listwise deletion if attributes are necessary after feature engineering
- Transform Data Type
  - Some categorical variables represented by number (INSNXU/next unit)
  - Separate categorical variables and numerical variables

----------Attributes with Missing Value----------

| | Total Missing | Percent Missing(%) |
|---|---|---|
| INUCLS | 223275 | 81.493175 |
| INUNIT | 223229 | 81.476385 |
| INMINL | 33543 | 12.242864 |
| INMAJL | 33466 | 12.214760 |
| INFNXO | 29474 | 10.757720 |
| INFNXU | 29188 | 10.653332 |
| INSNXO | 8684 | 3.169574 |
| INSNXU | 8400 | 3.065917 |
| INVCLK | 8017 | 2.926126 |
| INSWMC | 3794 | 1.384773 |
| INFWMC | 3262 | 1.190598 |
| INUOM | 2965 | 1.082196 |
| INFSHF | 1172 | 0.427768 |
| INFHLD | 141 | 0.051464 |
| INFENT | 102 | 0.037229 |
| INSHLD | 80 | 0.029199 |
| INFACT | 74 | 0.027009 |
| INHEAT | 28 | 0.010220 |
| INSENT | 28 | 0.010220 |

Drop Attributes

*Total number of missing values of each attribute &*
*Percentage of missing values of each attribute in descending order*

# Part1: Calculating Waiting Time

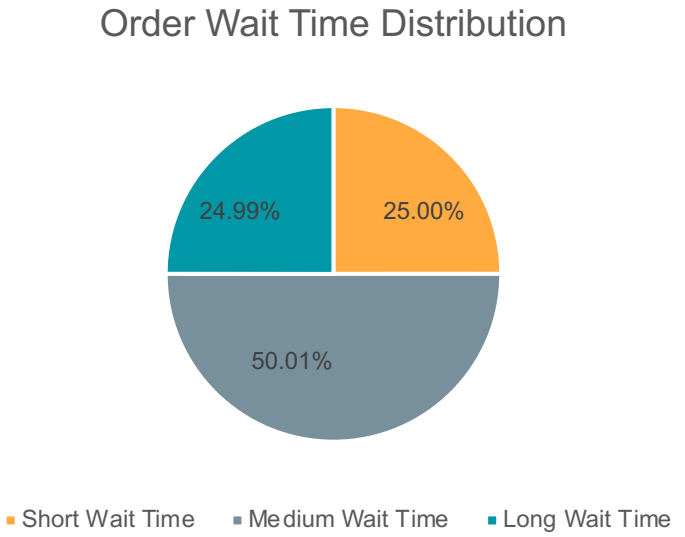
| | Inventory # | Next Operation | Next Unit | Start Timestamp | Finish Timestamp | Waiting Time (In Hour) |
|---|---|---|---|---|---|---|
| | **INVID#** | **INSNXO** | **INSNXU** | **INSTMS** | **INFTMS** | |
| **0** | 420000 | IN | 044 | 2019-06-11 07:40:46.146000 | 2019-06-11 08:44:24.893000 | N/A |
| **1** | 420000 | AN | 147 | 2019-06-11 08:44:24.893000 | 2019-06-11 20:13:02.019742 | |
| **2** | 420000 | IN | 147 | 2019-06-11 20:13:02.019742 | 2019-06-11 21:42:42.162513 | 11.477222 |
| **3** | 420000 | CR | 046 | 2019-06-11 21:42:42.162513 | 2019-06-12 17:42:43.189322 | |
| **4** | 420000 | IN | 046 | 2019-06-12 17:42:43.189322 | 2019-06-12 18:07:29.016738 | 20.000278 |
| **5** | 420000 | AN | 134 | 2019-06-12 18:07:29.016738 | 2019-06-15 06:53:54.290856 | |
| **6** | 420000 | IN | 134 | 2019-06-15 06:53:54.290856 | 2019-06-15 08:28:28.202985 | 60.773611 |

**Example:** 2019-06-11 20:13:02 - 2019-06-11 08:44:24 = 11.477 h

- Only focus on IN status (INSNXO/Next Operation)
- Machine Waiting Time =
  Start timestamp of this IN operation (INSTMS of this row) –
  Finish timestamp of last IN operation (INFTMS of last row)

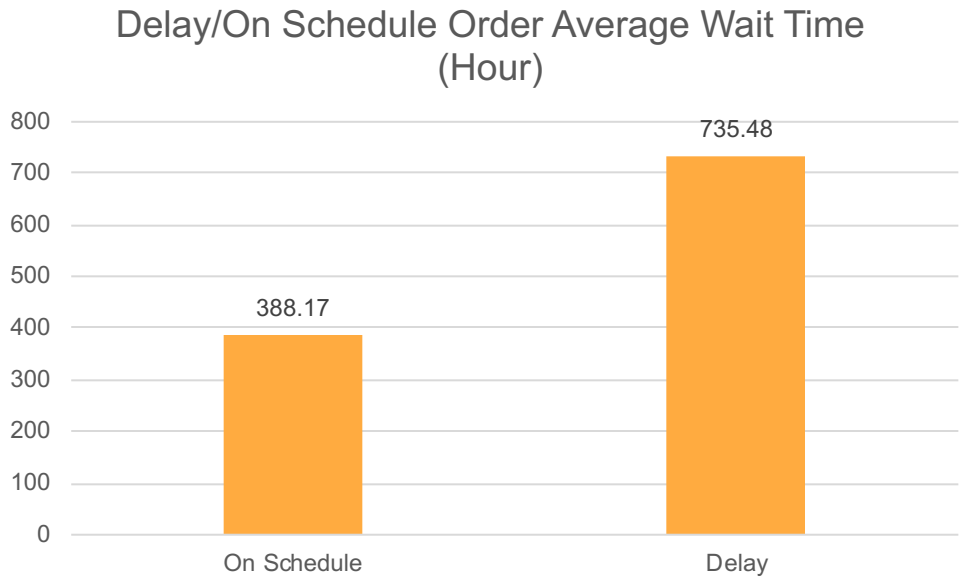
## Order Perspective

| | Wait Time (Hour) |
|---|---|
| Average | 59.06 |
| Max | 4379.16 |
| Upper Quartile | 61.01 |
| Median | 38.145 |
| Lower Quartile | 22.195 |
| Min | 0 |




- Short wait time: wait time < lower quartile
- Medium wait time: lower quartile ≤ wait time ≤ upper quartile
- Long wait time: wait time > upper quartile

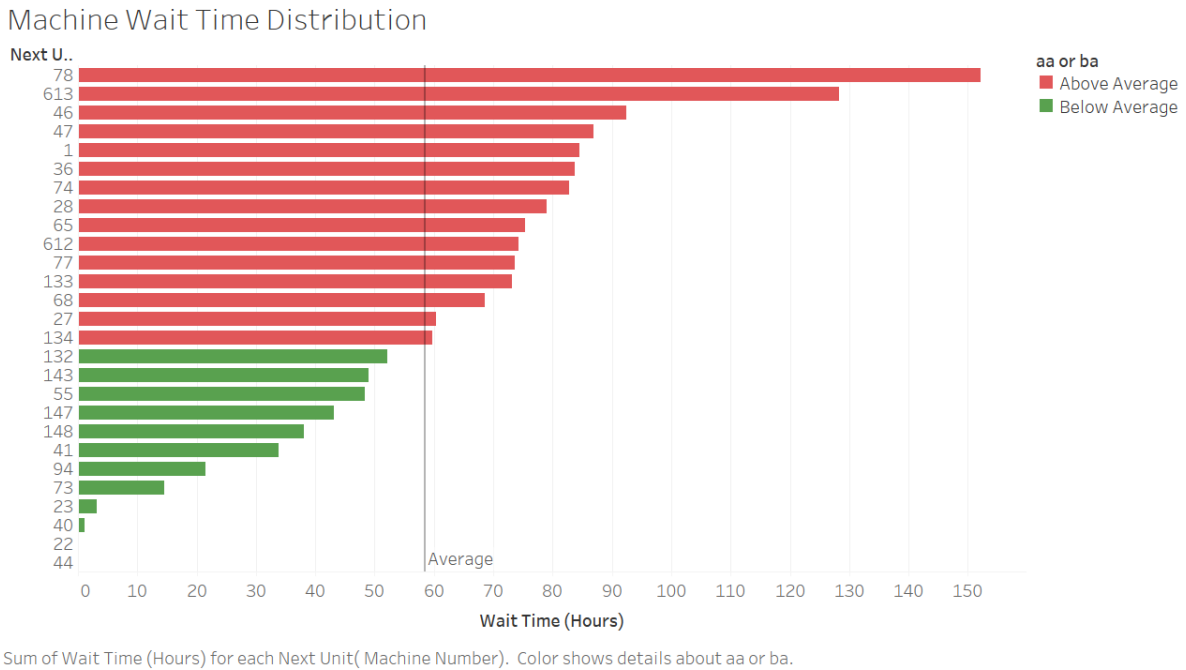
- **Order Perspective (Con'd)**




**The wait time of on scheduled orders is almost half of wait time of delay orders**

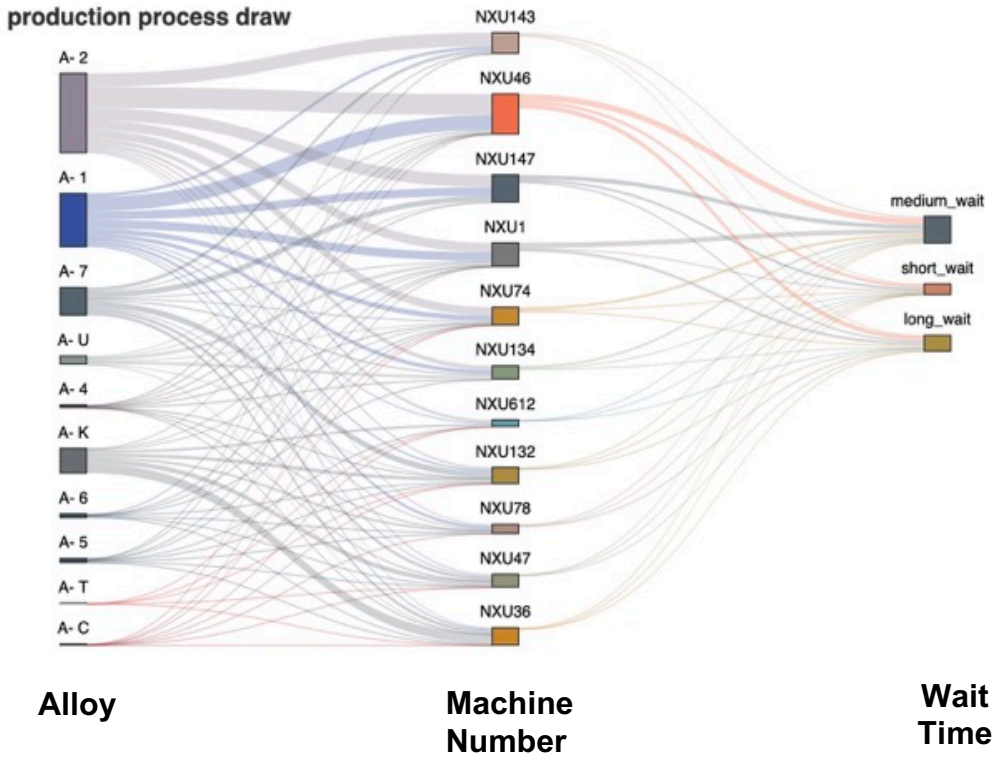
**Analysis and prediction on wait time can help to solve order delay problem**

## Machine Perspective




- Wait time for machine 78 is the longest
- Wait time for machine 22 and 44 is the shortest
- Average wait time for all machine is 58.5 hours

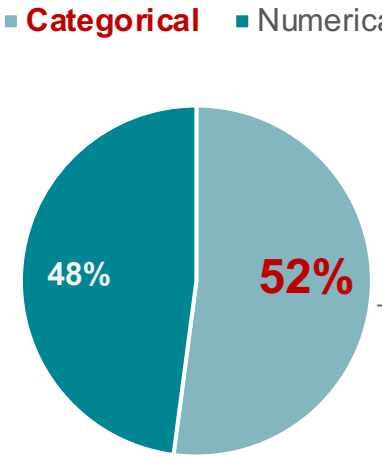
## Process Analysis (Sankey Plot)




- Most of alloy type of coils are A-1, A-2, A-K and A-7 (alloy type)
- A-2 coils are more likely to go to the machine 143 (short wait)
- A-1 coils are more likely to go to the machine 46 (long wait)
- A-7 coils are more likely to go to machine 132 and 147 (short wait)
- A-K coils are more likely to go to machine 36 (long wait)

# Part3: Why We Need Target Encoding to Deal with Features

## Categorical VS. Numerical Attributes




**Difficulty 1**

**Too Many subcategories (>=10) in some categorical variables**

- Some variables (i.e., INALLY) can use the first digit to represent but for most of them, hard to manipulate;
- Need more instructions about the meaning of those categories to manipulate!

**Difficulty 2**

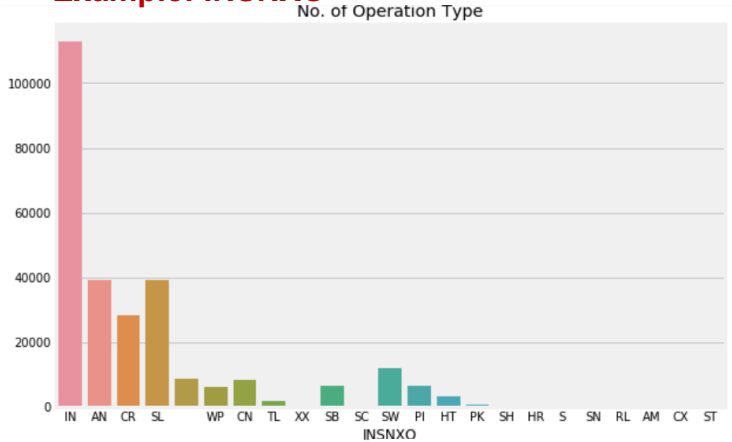
**Traditional Encoding Methods are NOT Suitable**

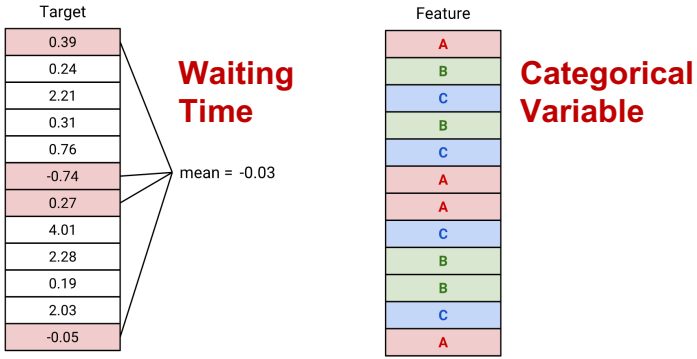
- One-hot encoding will create too many dimensions, but we have only 270,000 operation data;
- Integer encoding does NOT work because there is NO ordered relationship between each other.

# Part3: Target Encoding for Categorical Variables

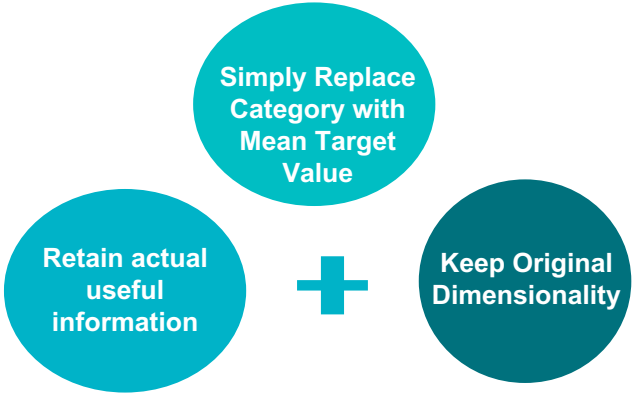
## How We Implement Target Encoding?

**Example: INSNXO**




## Advantages of Target Encoding




**Why is Mean encoding Great for our Dataset?**

- Can embody the target in the label
- could prove to be a **much simpler** alternative
- tend to group the similar classes

# Part3: New Feature Creation

## 1 INALLY

| Value | Frequency |
|---|---|
| 260 | 19043 |
| 1453 | 9895 |
| KLF5 | 8255 |
| 7151 | 5053 |
| 1102 | 3798 |
| 268 | 3389 |
| 7036 | 3332 |
| 122 | 3057 |
| UNIL | 2328 |
| 2608 | 2073 |
| 1921 | 2071 |
| 220 | 1803 |
| 1103 | 1606 |
| 1257 | 1599 |
| 110 | 1509 |
| 6476 | 1390 |
| 230 | 1186 |
| 226 | 1077 |
| 510 | 980 |

Use the first digit → Much closer to the business meaning

## 2 count_steps

| | INSENT | INSNXO | count_steps |
|---|---|---|---|
| 2 | 13599 | IN | 1 |
| 4 | 13599 | IN | 1 |
| 6 | 13599 | IN | 1 |
| 10 | 13599 | IN | 3 |
| 12 | 13599 | IN | 1 |
| ... | ... | ... | ... |
| 273906 | 25979 | IN | 2 |
| 273913 | 25979 | IN | 2 |
| 273916 | 25979 | IN | 2 |
| 273918 | 25979 | IN | 1 |
| 273933 | 26870 | IN | 3 |

| | INSENT | INSNXO |
|---|---|---|
| 0 | 13599 | IN |
| 1 | 13599 | AN |
| 2 | 13599 | IN |

Count Ignored Steps between Two 'IN' Status

## 3 Yield_Loss

| | INSGWT | INFGWT | yield loss |
|---|---|---|---|
| 0 | 20674 | 20485 | -0.009142 |
| 1 | 20485 | 20485 | 0.000000 |
| 2 | 20485 | 20485 | 0.000000 |
| 3 | 20485 | 20485 | 0.000000 |
| 4 | 20485 | 18913 | -0.076739 |
| ... | ... | ... | ... |
| 273975 | 1 | 1 | 0.000000 |
| 273976 | 1 | 1 | 0.000000 |
| 273977 | 1 | 1 | 0.000000 |
| 273978 | 1 | 1 | 0.000000 |
| 273979 | 1 | 1 | 0.000000 |

Yield loss=(start weight-finish weight)/start weight

# Part3: Select Suitable Features for Model Building

## Adopt Three Methods for Feature Engineering

### 1 Filter Method

- **Simplicity**: uses ranking technique and rank ordering method for variable selection; not including any mining algorithm;
- **Example:** Pearson's Correlation, Anova;

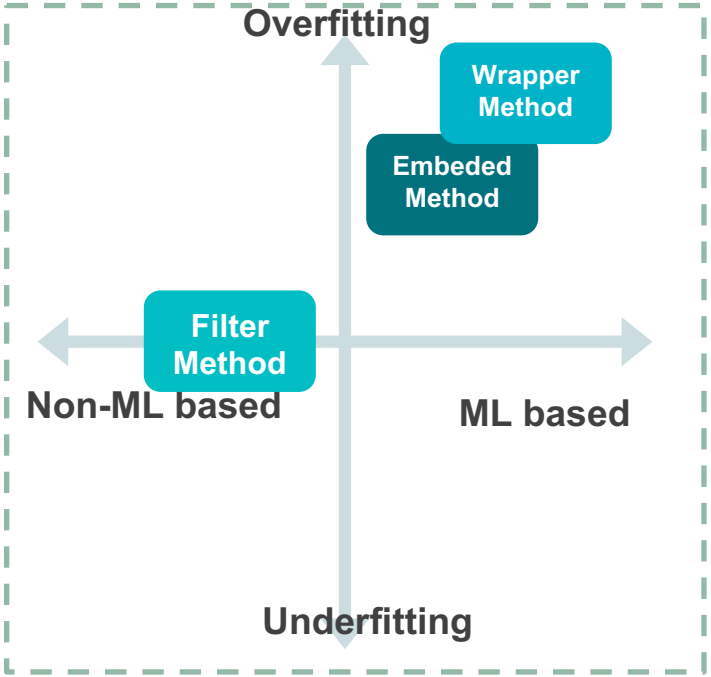
### 2 Embeded Method

- **Iterative:** take care of each iteration of the model training process;
- **Penalization:** L1, L2;

### 3 Wrapper Method

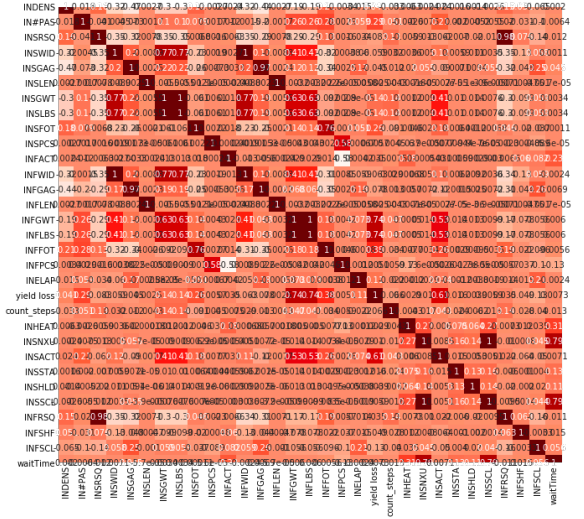
- **ML Based :** needs one machine learning algorithm and uses its performance as evaluation criteria;
- **Example**: forward selection, backward elimination;

## Comparison of Three Methods



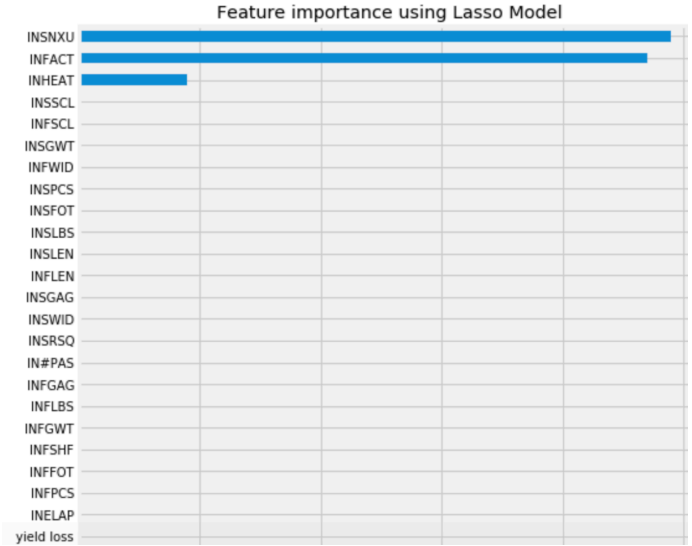
# Part3: Results from Three Methods and Final Feature Selection

## 1. Filter Method




```
INSNXU       0.794730
INSSCL       0.794671
waitTime     1.000000
Name: waitTime, dtype: float64
```

## 2. Embeded Method




INSNXU
INFACT
INHEAT

## 3. Wrapper Method

Backward Elimination

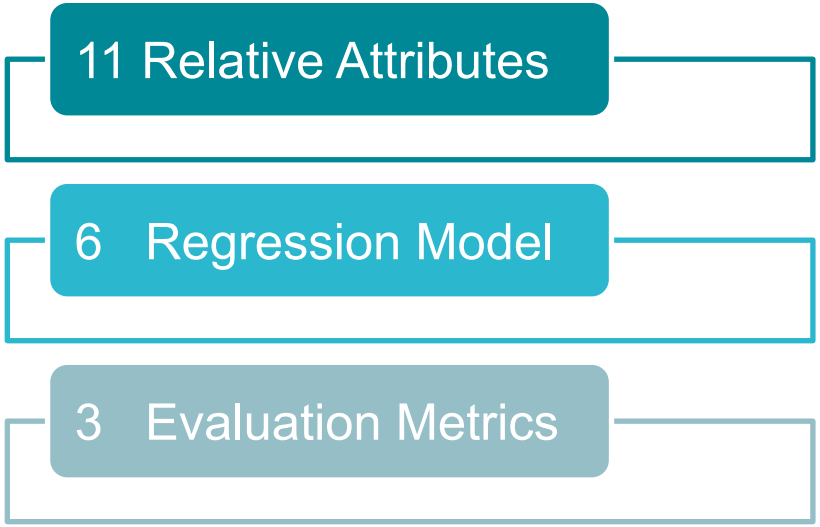
+

Recursive Feature Elimination

INDENS INSGAG INSPCS INFAG
INFPCS INSSCL IN#PAS INALLY

**Appeared two times!**

# Part4: Model Selection



# 11 relative attributes After Target Encoding

| | INDENS | IN#PAS | INSGAG | INSPCS | INFACT | INFGAG | INFPCS | INHEAT | INALLY | INSNXU | INSSCL | INFSCL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2** | 0.317 | 0 | 0.0350 | 1 | 2.154609e+07 | 0.0350 | 1 | 140431.222222 | 2 | 8.408858e+06 | 2.673828e+05 | 8.405765e+06 |
| **4** | 0.317 | 2 | 0.0350 | 1 | 2.154609e+07 | 0.0163 | 1 | 140431.222222 | 2 | 5.901128e+06 | 6.691793e+06 | 4.602240e+06 |
| **6** | 0.317 | 0 | 0.0163 | 1 | 2.154609e+07 | 0.0163 | 1 | 140431.222222 | 2 | 1.720098e+07 | 1.016231e+07 | 8.405765e+06 |
| **10** | 0.317 | 0 | 0.0163 | 1 | 2.154609e+07 | 0.0163 | 1 | 140431.222222 | 2 | 1.720098e+07 | 6.691793e+06 | 8.405765e+06 |
| **12** | 0.317 | 0 | 0.0163 | 1 | 3.436142e+05 | 0.0163 | 1 | 140431.222222 | 2 | 3.335057e+05 | 6.691793e+06 | 8.405765e+06 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **273906** | 0.323 | 0 | 0.0120 | 1 | 3.436142e+05 | 0.0120 | 1 | 200894.777778 | 1 | 6.916634e+05 | 7.165718e+06 | 3.697288e+05 |
| **273913** | 0.323 | 0 | 0.0120 | 1 | 2.154609e+07 | 0.0120 | 1 | 200894.777778 | 1 | 2.431705e+05 | 6.691793e+06 | 3.697288e+05 |
| **273916** | 0.323 | 1 | 0.0120 | 1 | 2.154609e+07 | 0.0120 | 1 | 200894.777778 | 1 | 3.850504e+05 | 7.165718e+06 | 3.697288e+05 |
| **273918** | 0.323 | 0 | 0.0120 | 1 | 3.436142e+05 | 0.0120 | 1 | 200894.777778 | 1 | 6.916634e+05 | 7.165718e+06 | 3.697288e+05 |
| **273933** | 0.323 | 0 | 0.0100 | 1 | 2.154609e+07 | 0.0100 | 1 | 321797.000000 | 7 | 3.286763e+05 | 7.165718e+06 | 3.697288e+05 |

# Part4: Model Selection

## 3 Metrics for Model Performance Evaluation

**MEAN ABSOLUTE ERROR**

Sum of absolute differences between our true waiting time and predicted value.
Range=[ 0, ∞]  Lower MAE = model fits better

**R SQUARED**

Measure of how close our actual data are to the fitted regression line.
Range=[0,1]  Closer to 1 = model fits better

**ROOT MEAN SQUARED ERROR**

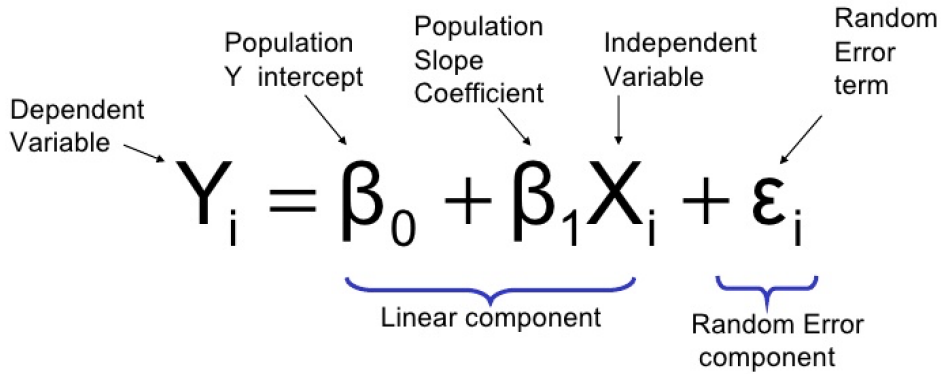
Square root of variance of residuals(prediction errors).
Range=[ 0, ∞]  Lower RMSE = model fits better

# Part4: Model Selection

- Linear Regression
- K-Nearest Neighbors
- Bayesian Ridge Regression
- LightGBM
- Random Forest
- Neural Network

# Linear Regression

- **Linear Regression** is model based on **supervised learning**.
- In this case, it performs the task to predict waiting time of each coil based on given 11 attributes.




| MAE | R2 | RMSE |
|---|---|---|
| 3772.11 | 0.98869 | 14963.68 |

# K Nearest Neighbors Regression

- Store all available cases and predict average waiting time of K nearest neighbors based on a similarity measure with almost **no learning process**.

**Distance functions**

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

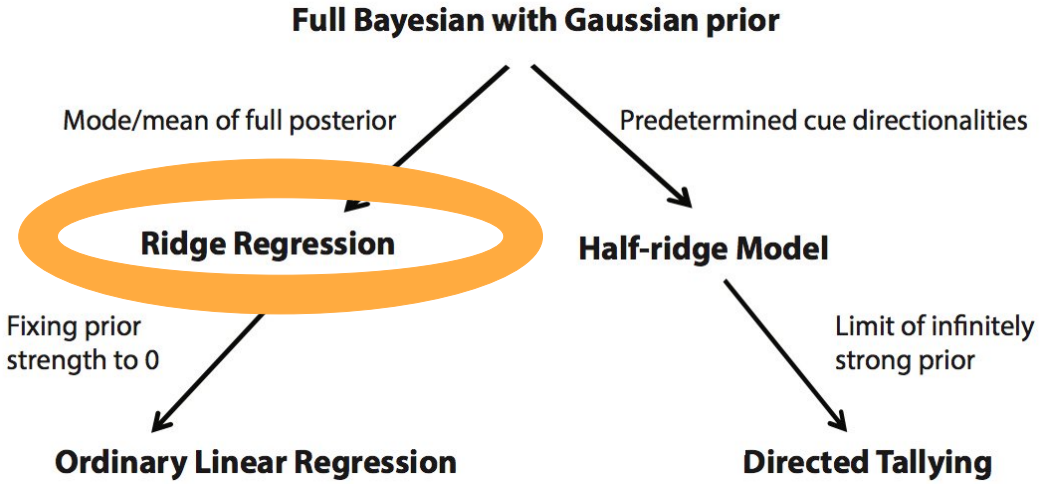
Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

***KNN algorithm***

| MAE | R2 | RMSE |
|---|---|---|
| 1830.62 | 0.6235 | 167685.23 |

# Bayesian Ridge Regression

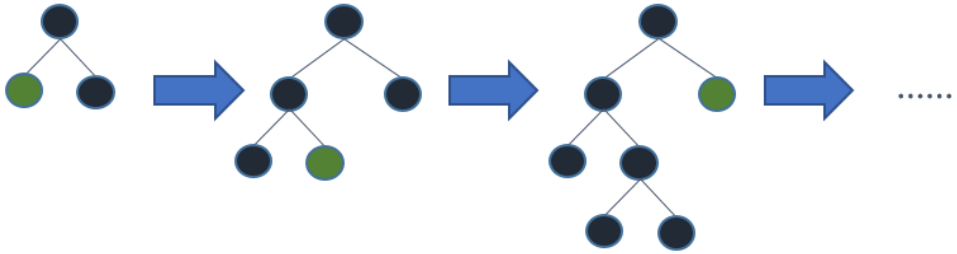
- A Ridge regression formulated as Bayesian estimator.
- **Self-adaptive capability** avoiding overfitting
- 2 important hyperparameters: **α** & **λ**




| MAE | R2 | RMSE |
|---|---|---|
| 3846.62 | 0.9889 | 14859.94 |

# LightGBM

- A **gradient boosting** framework that uses **tree-based learning** algorithm
- High-Speed data processing

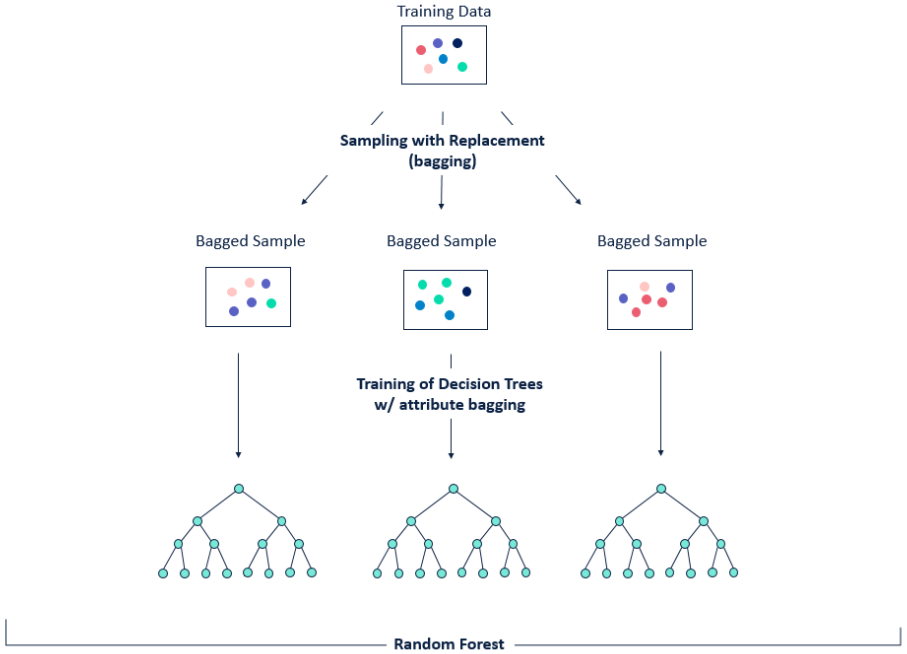


Leaf-wise tree growth

| MAE | R2 | RMSE |
|---|---|---|
| 7860.36 | -29268.17 | 181696.44 |

# Random Forest

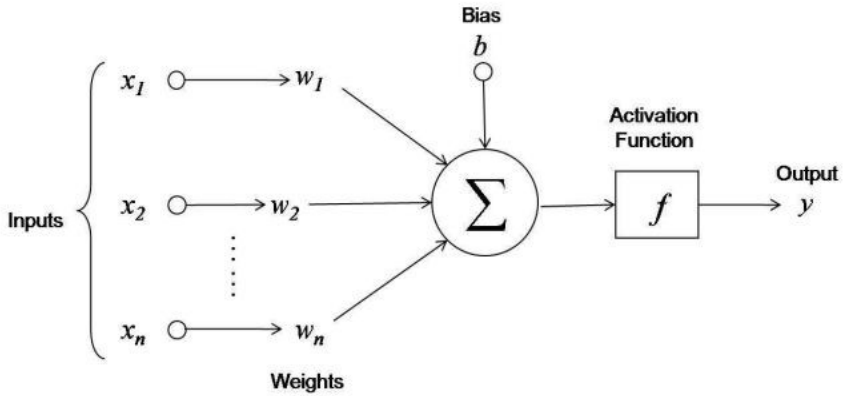
- An ensemble learning method for classification, regression and other tasks.
- In this case, it's possible to use random forests model to predict waiting time of each coil.




| Mean Absolute Error | R2_Score | Root Mean Square Error |
|---|---|---|
| 2106.16 | 0.9997 | 2110.09 |

# Neural Network

**Neural networks** are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. We can use this method to predict waiting time because of its complexity and accuracy.




| MAE | R2_Score | RMSE |
|---|---|---|
| 1185.73 | -4.99 | 140738.01 |

# Model Performance Comparison

| Model Name | MAE | R2_Score | RMSE |
|---|---|---|---|
| Linear Regression | 3772.11 | 0.98869 | 14963.68 |
| KNN | 1830.62 | 0.6235 | 167685.23 |
| Bayesian Model | 3846.62 | 0.9889 | 14859.94 |
| LightGBM | 7860.36 | -29268.17 | 181696.44 |
| **Random Forest** | **2106.16** | **0.9997** | **2110.09** |
| Neural Network | 1185.73 | -4.99 | 140738.01 |

# Part 5: Summary & Conclusion

## EDA

- Wait time for machine 78 is the longest and machine 22 and 44 is the shortest (0 Hour).
- The wait time of on-schedule orders is almost half of wait time of delay orders
- A1, AK(Alloy) coils are more likely go to the long wait machine

## Predictive Model

- Random Forest model has the best performance and can be used in wait time prediction

## Recommendation

- Buy more machine 78 if budget permits
- Some machine 22 and 44 may be unused in production since the wait time is 0
- Adjusting the delivery date according to the predictive wait time can reduce order delay

# THANK YOU

Aurubis
Metals for Progress

# Appendix - timeline