INTRODUCTION TO DATA SCIENCE

Group 14 -- Project phase 2

First Variable Selection

Name	Student ID
Pham, Quoc Huy	2299356
Hussain, Zakiuddin	2338350
Lee, Daeul Haven	2308018
Preetham	2288949
Jayanth	2288552
Srikavya	2311351


```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.feature_selection import *
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LassoCV
from sklearn.metrics import accuracy_score
```

Q2. Selecting Perform Variable Selection. Choose one of the following techniques and explain why you selected that one.

- Lasso: https://en.wikipedia.org/wiki/Lasso_(statistics))
- KNN
- Correlation

1. Read the data

```
#Read the data
data = pd.read_csv('Group_14_Clean_Data.csv')
data.head()
```

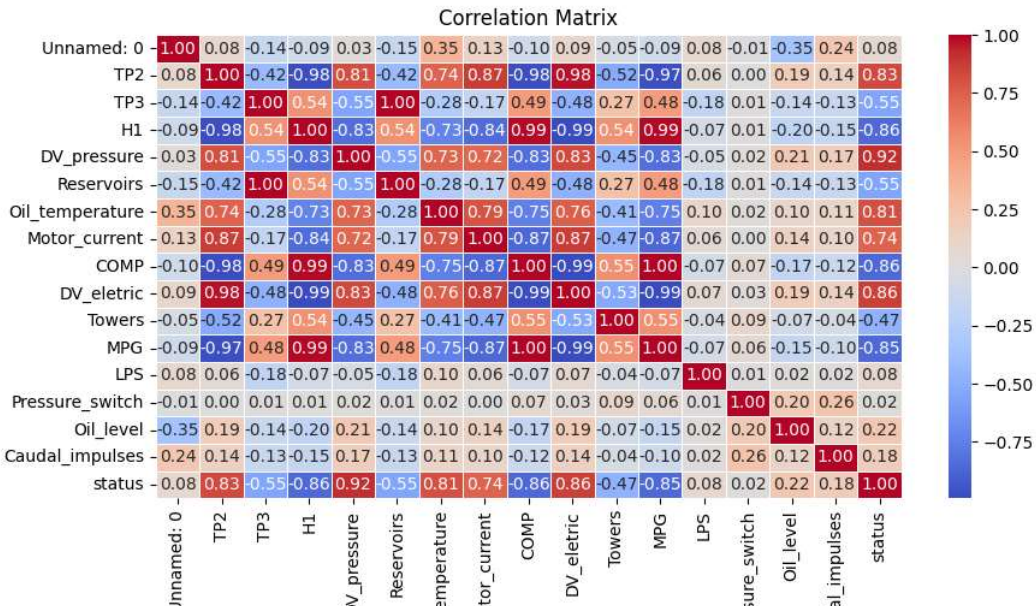
	Unnamed: 0	timestamp	TP2	TP3	H1	DV_pressure	Reservoirs	Oil_temperature	Motor_current	COMP	DV_eletric	Towers	MPG	LPS	Pressure_switch
0	562564	2020-04-18 00:00:01	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	1.0	0.0	1.0	1.0	0.0	1.0
1	562565	2020-04-18 00:00:13	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	1.0	0.0	1.0	1.0	0.0	1.0
2	562566	2020-04-18 00:00:24	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	1.0	0.0	1.0	1.0	0.0	1.0
3	562567	2020-04-18	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	0.0	0.0	0.0	0.0	0.0	0.0

2. Visualize correlation between variables

```
#Checking the correlation matrix
corr = data.corr()
```

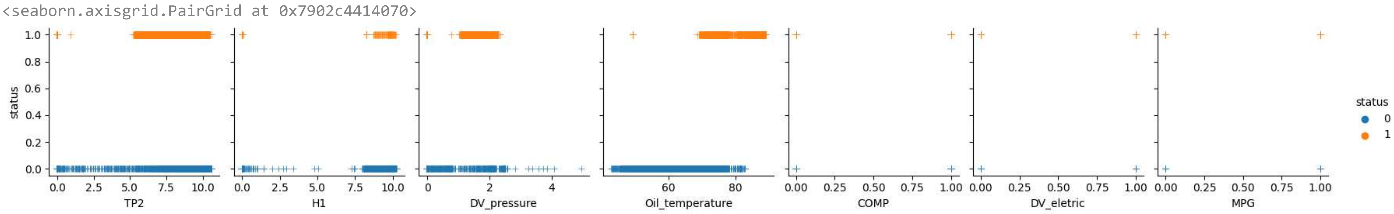
```
<ipython-input-5-0ba1501a6a21>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will defaul
  corr = data.corr()
```

```
# Plot the correlation matrix using Seaborn's heatmap
plt.figure(figsize=(10, 5))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```




```
sns.pairplot(data,
             x_vars=["TP2","H1","DV_pressure","Oil_temperature", 'COMP', 'DV_eletric', 'MPG'],
             y_vars=["status"], hue="status", markers = "+")
```

<seaborn.axisgrid.PairGrid at 0x7902c4414070>



3. Choose selection method

Since the correlation plot shows strong relationships between predictor variables and target variables, our group chose the Lasso method for feature selection. The Lasso method is known for its ability to select relevant features and reduce the impact of irrelevant ones, making it a suitable choice for our analysis. Additionally, it helps prevent overfitting by introducing a penalty term that encourages sparsity in the model. Compared to using KNN and correlation coefficients, the Lasso method provides a more robust and interpretable solution. It not only considers the relationships between predictors and the target variable but also takes into account the multicollinearity among predictors, which can lead to biased coefficient estimates. Moreover, the Lasso method allows for automatic feature selection, eliminating the need for manual selection based on expert knowledge or trial and error. Overall, using the Lasso method will enhance the accuracy and interpretability of our analysis.

```
x = data.iloc[:, 2:-1]
y = data.iloc[:, -1]
x.head()
```

	TP2	TP3	H1	DV_pressure	Reservoirs	Oil_temperature	Motor_current	COMP	DV_eletric	Towers	MPG	LPS	Pressure_switch	Oil_level	Caudal_im
0	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	1.0	0.0	1.0	1.0	0.0	1.0	1.0	
1	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	1.0	0.0	1.0	1.0	0.0	1.0	1.0	
2	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	1.0	0.0	1.0	1.0	0.0	1.0	1.0	
3	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	-0.018	8.248	8.238	-0.024	8.248	49.45	0.04	1.0	0.0	1.0	1.0	0.0	1.0	1.0	

```
# Create a LassoCV model
lasso_cv = LassoCV(cv=10)

# Use SelectFromModel to perform feature selection based on feature importance
sfm = SelectFromModel(lasso_cv, threshold='median')


#Separate train test set
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
lasso_cv.fit(X_train, y_train)
```

```
▾ LassoCV
LassoCV(cv=10)
```

```
print(f"List of features before selection:\n {', '.join(X_train.columns)}")
```

```
List of features before selection:
 TP2, TP3, H1, DV_pressure, Reservoirs, Oil_temperature, Motor_current, COMP, DV_eletric, Towers, MPG, LPS, Pressure_switch, Oil_level, Caudal_impulses
```

```
selected_features = X_train.columns[lasso_cv.coef_ != 0]
print(f"List of selected features:\n {', '.join(selected_features)}")
```

```
List of selected features:
 TP2, H1, DV_pressure, Reservoirs, Oil_temperature, Motor_current, Oil_level
```