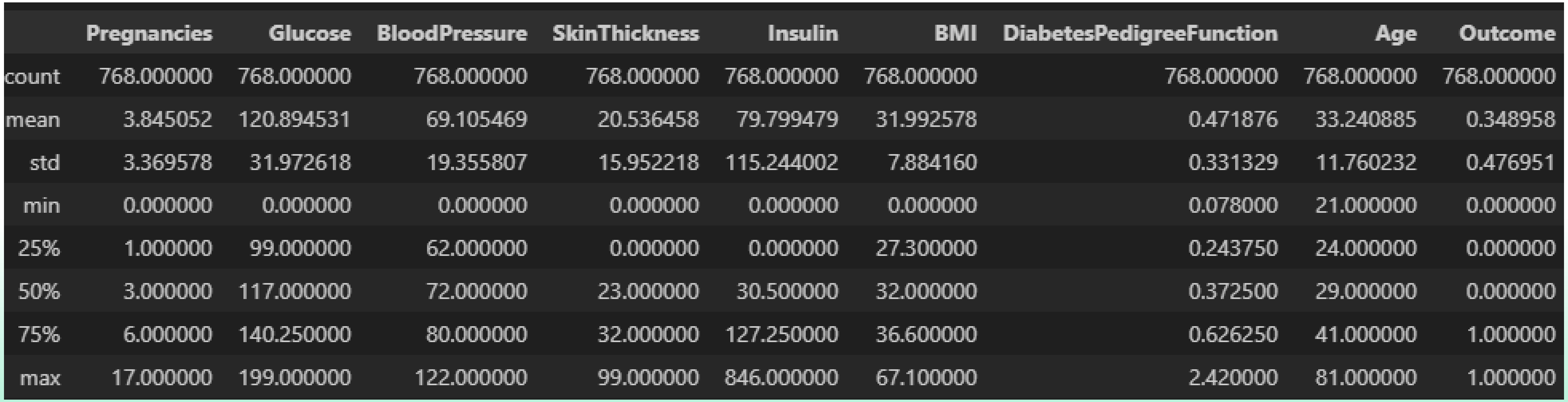
# ML SUPERVISED LEARNING

Ethan Lam

# EDA

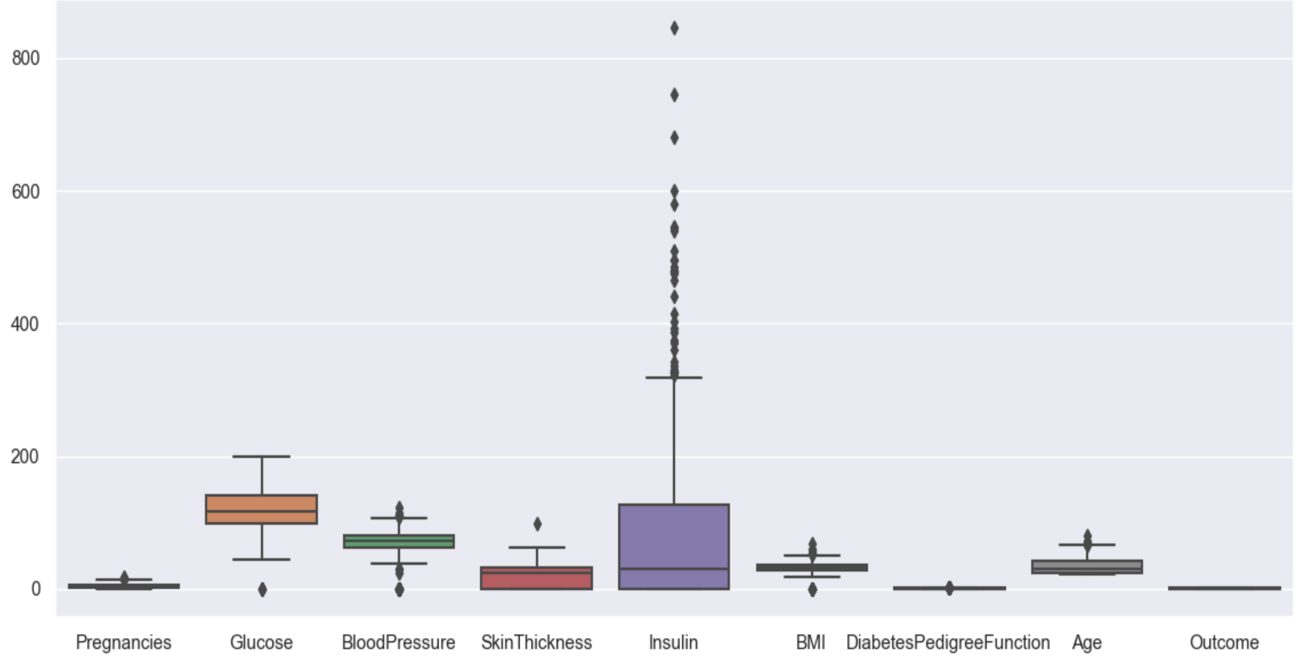
- First check of the data



| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| count | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 |
| mean | 3.845052 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | 0.471876 | 33.240885 | 0.348958 |
| std | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | 0.331329 | 11.760232 | 0.476951 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078000 | 21.000000 | 0.000000 |
| 25% | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.243750 | 24.000000 | 0.000000 |
| 50% | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 32.000000 | 0.372500 | 29.000000 | 0.000000 |
| 75% | 6.000000 | 140.250000 | 80.000000 | 32.000000 | 127.250000 | 36.600000 | 0.626250 | 41.000000 | 1.000000 |
| max | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420000 | 81.000000 | 1.000000 |

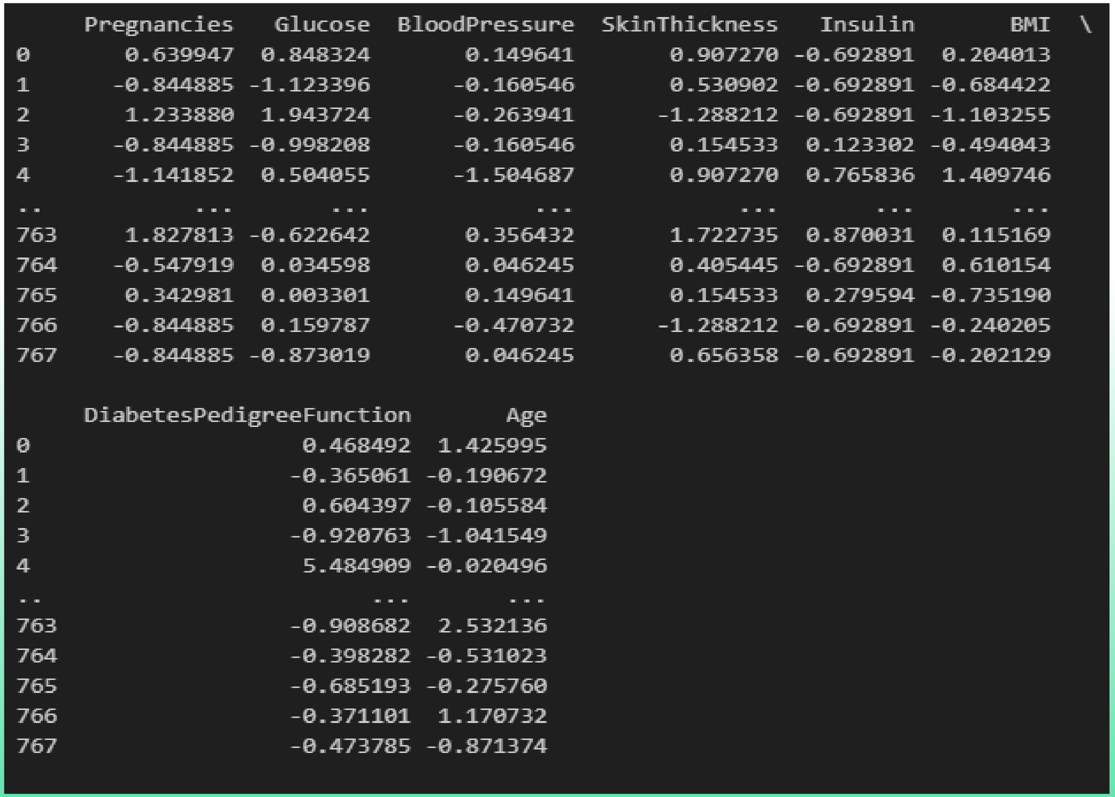
# EDA – HANDLING MISSING VALUES

- Update missing values by median
- No duplicated values found
- Outliers has been founded in Insulin feature



# SCALING

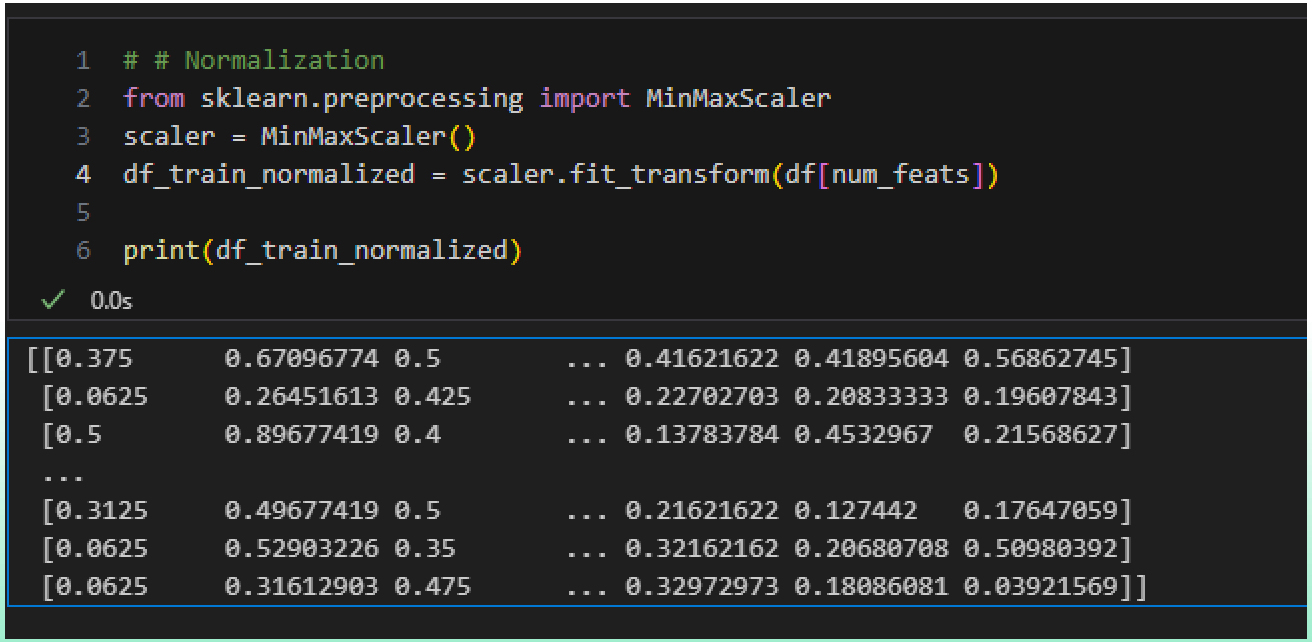
- Scaling is necessary because the features in dataset have different scales
- The output of the scaling



```
    Pregnancies   Glucose  BloodPressure  SkinThickness   Insulin       BMI  \
0      0.639947  0.848324       0.149641       0.907270 -0.692891  0.204013
1     -0.844885 -1.123396      -0.160546       0.530902 -0.692891 -0.684422
2      1.233880  1.943724      -0.263941      -1.288212 -0.692891 -1.103255
3     -0.844885 -0.998208      -0.160546       0.154533  0.123302 -0.494043
4     -1.141852  0.504055      -1.504687       0.907270  0.765836  1.409746
..          ...       ...            ...            ...       ...       ...
763    1.827813 -0.622642       0.356432       1.722735  0.870031  0.115169
764   -0.547919  0.034598       0.046245       0.405445 -0.692891  0.610154
765    0.342981  0.003301       0.149641       0.154533  0.279594 -0.735190
766   -0.844885  0.159787      -0.470732      -1.288212 -0.692891 -0.240205
767   -0.844885 -0.873019       0.046245       0.656358 -0.692891 -0.202129

    DiabetesPedigreeFunction       Age
0                   0.468492  1.425995
1                  -0.365061 -0.190672
2                   0.604397 -0.105584
3                  -0.920763 -1.041549
4                   5.484909 -0.020496
..                       ...       ...
763                -0.908682  2.532136
764                -0.398282 -0.531023
765                -0.685193 -0.275760
766                -0.371101  1.170732
767                -0.473785 -0.871374
```

# NORMALIZATION



```python
# # Normalization
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df_train_normalized = scaler.fit_transform(df[num_feats])

print(df_train_normalized)
```

```
[[0.375      0.67096774 0.5        ... 0.41621622 0.41895604 0.56862745]
 [0.0625     0.26451613 0.425      ... 0.22702703 0.20833333 0.19607843]
 [0.5        0.89677419 0.4        ... 0.13783784 0.4532967  0.21568627]
 ...
 [0.3125     0.49677419 0.5        ... 0.21621622 0.127442   0.17647059]
 [0.0625     0.52903226 0.35       ... 0.32162162 0.20680708 0.50980392]
 [0.0625     0.31612903 0.475      ... 0.32972973 0.18086081 0.03921569]]
```

# TRAINING MODEL

- There are three models used

  Linear regression

  XGBRegressor

  RandomForestRegressor

- The model evaluation metrics

  Mean squared error

  F1 score

  AUC-score

→ Model Random Forest has the best fit as it has the higher F1Score and AUC score

# CONCLUSION

- Linear regression is the best fit model in this scenario
- The most difficult part is to do feature selection