Nama : Aaron Winston Gho

NIM : 2702210522

1. Disini kelompok saya ngescaling dulu karena kemaren di preprocessing kita belum scalling
   - Untuk train data kita akan fit dan transform untuk Age disini kita pakai robust meningat ada outlier dan kita ga mau itu terpengaruhi sedangkan weight kita pakai standard scaler karena ga ada outlier jadi kita pakai standard.

```
age_scaler = RobustScaler()
weight_scaler = StandardScaler()
```

```
x_train["Age"] = age_scaler.fit_transform(x_train[["Age"]])
x_train["Weight"] = weight_scaler.fit_transform(x_train[["Weight"]])
```

```
x_train.head()
```

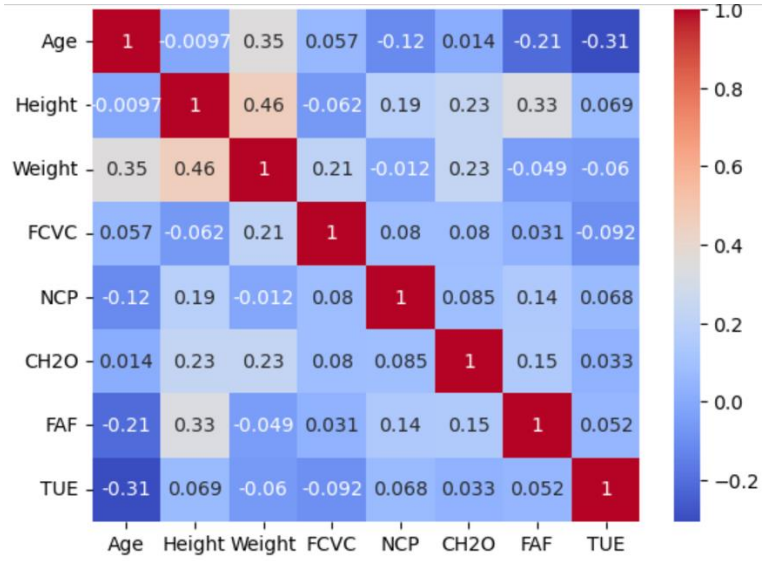
   - Untuk Test data kita hanya transform agar ga overfit nanti datanya kalau di fit juga.

```
caling the test dataset
```

```
x_test["Age"] = age_scaler.transform(x_test[["Age"]])
x_test["Weight"] = weight_scaler.transform(x_test[["Weight"]])
```

```
x_test.head()
```

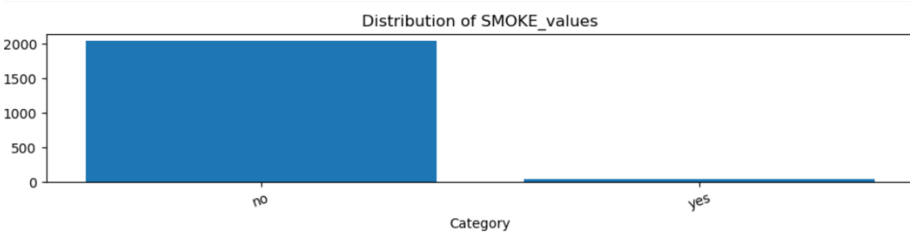
2. Untuk model yang akan digunakan oleh kelompok kami adalah Neural Network. Hal ini dikarenakan kelebihan dari neural network sendiri seperti :
   - Bagus dalam menangani variable atau kolom yang hubungannya non linier atau ga jelas, melalui spearmen di dataset kita, kita menemui ada beberapa variable yang hubungannya ga jelas

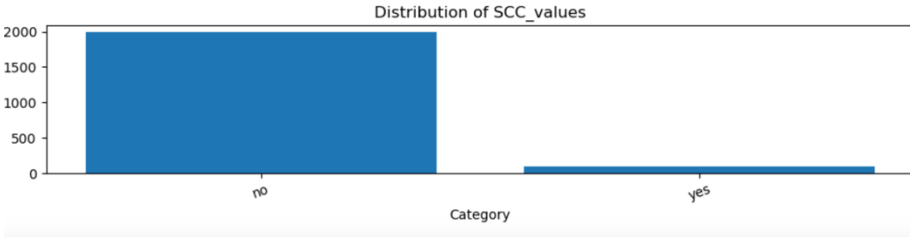


Disini ada height dan NCP lalu Weight dengan TUE, CH2O dengan FCVC dan masih banyak lagi. Oleh karena itu agar memudahkan model nantinya kita akan menggunakan algoritam neural network yang akan membagi layer untuk model nantinya dilatih.

- Inbalance data

  Dari data di atas masih ada beberapa variable yang inbalance seperti dua ini dibawah, hal ini yang menjadi salah satu penguat juga kita memilih neural network karena lebih mudah untuk mengatasi masalah seperti inbalance data. Jadi nantinya kita bisa pakai oversampling atau class weighting dari parameter NN.





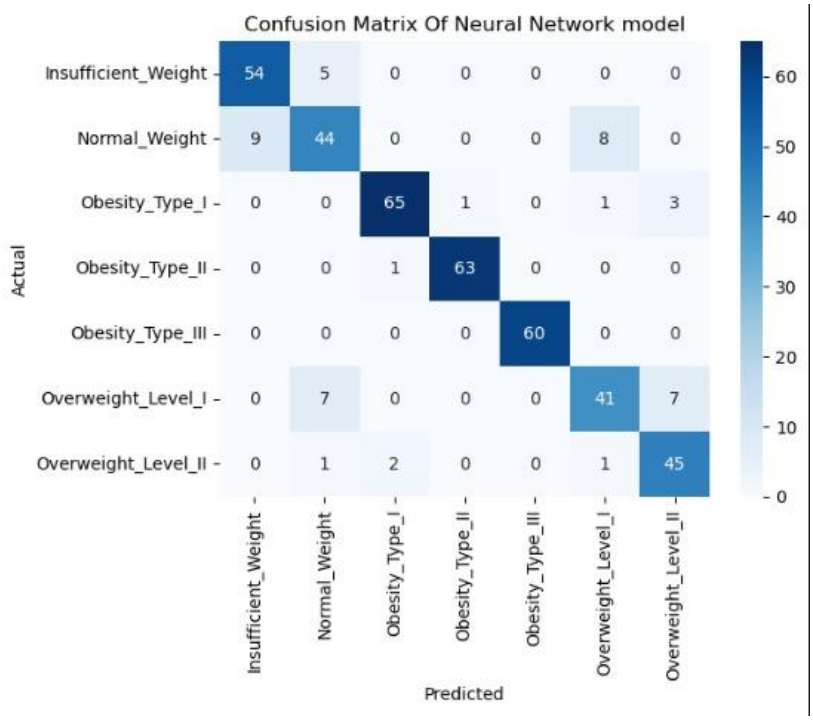
- Selanjutnya kemampuan neural network yang gampang dilatihnya untuk menangani data kita yang kompleks dan cukup ribet.
- Last but not least adalah karena regulasi yang memudahkan kita untuk melatih data. Regulasi NN seperti dropout, L2 regularization, dan early stopping dapat kita pakai untuk menghindari overfitting. Ini very useful bgt ketika bekerja

dengan data yang sangat bervariasi, seperti dataset yang berhubungan dengan obesitas.

3. Hasil yang didapatkan dari data kita melalui menggunakan model Neural Network sebagai berikut:

```
Precision:      0.8844904887792439
Recall:         0.8876191476661031
F1_score:       0.8851974457222843
Accuracy:       0.8899521531100478
```

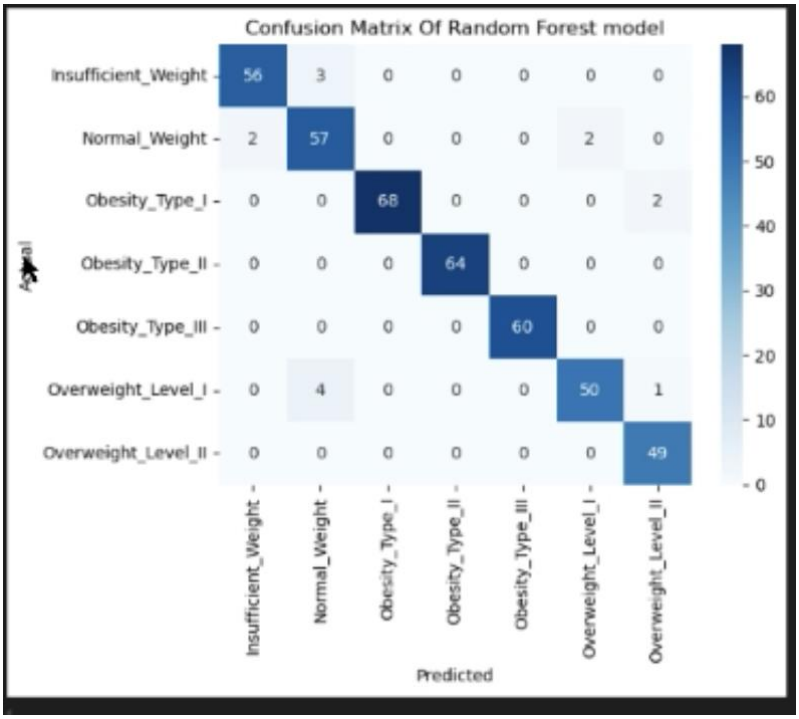
Dapat kita lihat dalam gambar kalau kita berhasil menghasilkan hasil yang cukup bagus yaitu 88.7% an ke atas. Di satu sisi iya ini menghasilkan persentase yang lebih kecil dari based model yang dipakai. Walaupun begitu satu sisi yang bagus disini recall lebih besar dari precision. Kenapa ini bagus ? karena Dari data ini kita mau memfokuskan pada recall yang lebih besar dikarenakan lebih bagus yang ga kena obesitas dinyatakan obesitas daripada yang obesitas tapi ga dinyatakan demikian. Jadi focus tujuan nya di recall yang lebih tinggi dari precision dan disini kita berhasil meraih itu.



Nah untuk confussion matrixnya kita bisa lihat very demure, masih ada beberapa yang salah dan tentunya lebih banyak dari based. Namun perlu diinget ini recallnya > dari precision dan angka segini ga terlalu overfit juga. Bedanya drastic ama based ☺.

Jadi dapat diajukan kalau model NN ini dapat digunakan sebagai ganti random forest jika ingin dipakai.

4. Selain model yang kita sarakan, kita juga meneliti model yang orang orang pakai dan kita lihat kalau orang biasanya memakai Randomforest atau XGboost di data ini. Disini kita pilih satu doang untuk dicompare jadi kita pilih randomforest karena emang lebih cepat dan efisien dari XGboost. Nah disini kita lihat kalau hasil yang dihasilkan dibawah ini:



Disini kita bisa lihat kalau model sangat successful banget untuk mengidentifikasi siapa yang obesitas dan tidak walaupun masih ada 1-5 yang ga bener tapi ini menandakan model kita udah bagus banget dengan model yang biasanya orang pakai.

Dengan classification report sebagai berikut:

Precision, Recall, F1 score, Accuracy

```
test_precision = precision_score(y_test, rf_prediction, average="macro")
test_recall = recall_score(y_test, rf_prediction, average="macro")
test_f1 = f1_score(y_test, rf_prediction, average="macro")
test_accuracy = accuracy_score(y_test, rf_prediction)

print(f"Precision:\t{test_precision} \nRecall:\t\t{test_recall} \nF1_score:\t{test_f1}\nAccurac
```

```
Precision:      0.9657126278893521
Recall:         0.966299750342937
F1_score:       0.9656640960852966
Accuracy:       0.9665071770334929
```

Ini yang ditakutin dari kita cuman satu yaitu overfit aja sih. Karena 96% itu sangat besar banget. Akan tetapi jeng jeng plot twistnya ini dataset sintesis jadi emang udah expected sebagus ini not overfit. Selain itu, walaupun ada data yang inbalance di beberapa kolom, y nya distribute uniform jadi mesin bisa belajar banyak banget dari situ. Sayangnya untuk model Randomforest kita dapat melihat kalau nilai recall > dari nilai precision jadi kemungkinan untuk model mengidentifikasi seseorang yang obesitas jadi ga obesitas lebih besar dan ini bahaya banget. oleh karena itu saya menyarankan kita bisa pakai menggunakan neural network atau pilihan lainnya adalah grid search cari recall yang paling tinggi.

5. Existing Challenges dan Possible Works yang bisa dilakukan di masa depan

Dari Segi Tantangan dan apa yang harus dilakukan Ada beberapa di dataset kita:

No	Challenges	Possible Work
1	**Ketidakseimbangan data** Variabel-variabel atau kolom yang distribusinya sangat ga wajar atau unbalance. Ini merupakan salah satu tantangan terbesar juga di data kita. Walaupun kita tahu Y nya distribute uniform ya.	**Oversampling** Kedepannya dapat diterapkan Teknik oversampling agar mendapatkan hasil yang ga overfit banget (bayangin pak run pertama dapatnya 1 :) , for the first time in forever)
2	**Overfitting** Meskipun sudah diatasi dengan model yang NN, model sering memberikan hasil yang overfitting terkadang.	**Hyperparameter Tuning** Melakukan gridsearch untuk menemukan kombinasi hyperparameter yang membuat model lebih optimal
3	**Fitur yang tidak sepenuhnya usable untuk data dan memicu bias pada satu fitur** Dari data ini tentunya ada kolom yang memberikan kontribusi lebih kecil dibandingkan kolom lain sehingga mengacu pada bias ujungnya.	**Eksplorasi Data yang Lebih Mendalam** Memanfaatkan teknik feature engineering dan data augmentation untuk meningkatkan kualitas data dan hasil model dan mencari tau kenapa sih dengan y yang distribute uniform tetapi kolom lain masih ada yang inbalance bisa dapatin hasil bagus.