Link Dataset :

https://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition

Link Google Colab Classification :

https://colab.research.google.com/drive/1C9EUsCUCYvRJ2L-QS8cQ2caXVcDu_lWw?usp=sharing

A. Latar Belakang Masalah

Tahun 2016, WHO (*World Health Organization*) mengatakan bahwa obesitas dan kelebihan berat badan sebagai akumulasi lemak yang berlebihan di area tubuh tertentu dapat membahayakan kesehatan. Setiap tahunnya, jumlah orang yang menderita obesitas selalu naik, hal ini diperkuat dengan data dengan pada tahun 2014 sudah lebih dari 1900 juta orang dewasa dengan rentang usia 18 tahun atau lebih. Banyak factor yang memicu terjadinya kenaikan berat badan yang signifikan, seperti meningkatkan asupan makanan padat energi dengan lemak yang tinggi, penurunan aktivitas fisik, kemudahan moda transportasi terkini, meningkatnya jumlah urbanisasi, keturunan, gangguin psikologis dan kebiasaan makan, dan lain-lain.

Beberapa penulis sudah melakukan penelitian terkait analisis dan prediksi obesitas menggunakan teknik *data mining* dan pemanfaatan bidang keilmuan kecerdasan buatan. Namun, beberapa penelitian sebelumnya tidak menambahkan faktor seperti latar belakang keluarga. Pada penelitian ini, akan dilakukan percobaan dengan menambahkan beberapa faktor dominan dan relevan sebagai penentu diagnosa obesitas pada seseorang menggunakan beberapa algoritma klasifikasi.

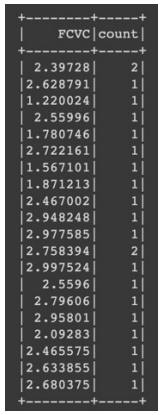
B. Tujuan

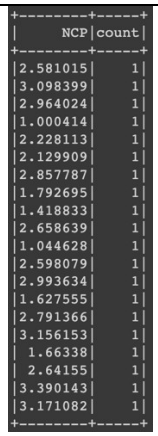
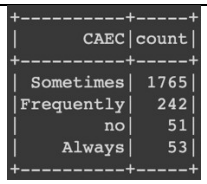
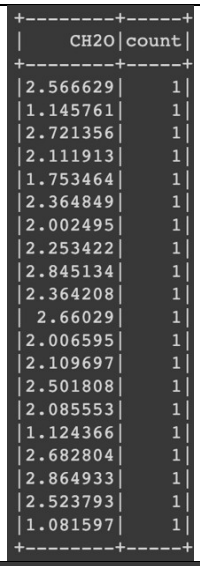
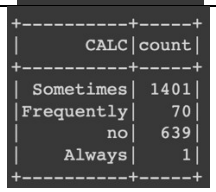
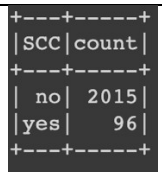
Penelitian ini bertujuan untuk mendapatkan prediksi dan klasifikasi apakah seseorang terkena obesitas atau tidak menggunakan beberapa algoritma *machine learning* yaitu Logistic Regression, Decision Tree, dan Random Forest. Informasi keluaran yang diharapkan adalah prediksi kategori kelompok obesitas berdasarkan data inputan yang diberikan. Harapannya, dengan diketahuinya kategori jenis obesitas seseorang yang diinputkan pada model prediksi, maka dapat membantu masyarakat maupun tenaga medis untuk mengambil keputusan tindakan selanjutnya, baik itu pencegahan maupun pengobatan.

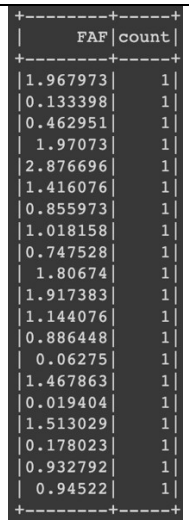
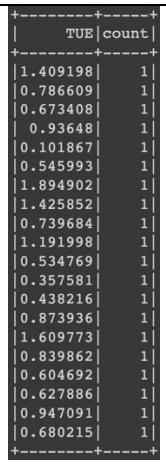
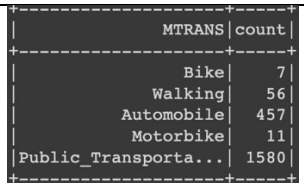
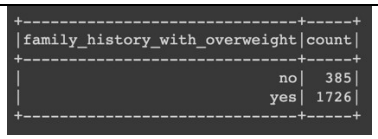
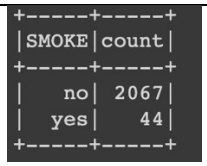
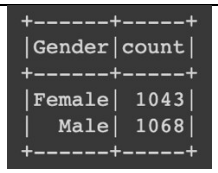
C. Deskripsi Data

Data yang digunakan untuk penelitian ini bersumber dari website UCI Machine Learning Repository pada link https://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition+ . Uci Machine Learning Repository adalah kumpulan basis data, teori domain, dan generator data yang digunakan oleh komunitas pembelajaran mesin untuk analisis empiris algoritma pembelajaran mesin.

Penelitian ini menggunakan data dengan judul "**Estimation of obesity levels based on eating habits and physical condition**" yang berisi informasi terkait perkiraan tingkat obesitas pada orang-orang dari negara-negara Meksiko, Peru dan Kolombia, dengan usia antara 14 dan 61 tahun. Data dikumpulkan menggunakan platform web dengan survei di mana anonym pengguna menjawab setiap pertanyaan, kemudian informasi diproses memperoleh 17 atribut dan 2111 baris data. Berikut adalah fitur-fitur yang digunakan pada dataset beserta dengan penjelasannya:

Atribut	Penjelasan	Value	Tipe Data
Frequent Consumption of High Caloric Food (FAVC)	Sering atau tidak dalam mengkonsumsi makanan berkalori tinggi		String
Frequency of consumption of vegetables (FCVC)	Sering atau tidak dalam mengkonsumsi sayuran		Double

Number of main meals (NCP)	Jumlah makanan utama		String
Consumption of food between meals (CAEC)	Konsumsi makanan di antara waktu makan		String
Consumption of water daily (CH2O)	Konsumsi air setiap hari		Double
Consumption of alcohol (CALC)	Konsumsi alkohol		String
Calories consumption monitoring (SCC)	Pemantauan konsumsi kalori		String

Physical activity frequency (FAF)	Frekuensi aktivitas fisik		Double
Time using technology devices (TUE)	Waktu penggunaan perangkat teknologi		Double
Transportation used (MTRANS)	Transportasi yang digunakan		String
Family History with Overweight	Riwayat obesitas dari keturunan		String
SMOKE	Aktifitas merokok harian		String
Jenis Kelamin	Gender dari responden		String
Umur	Umur dari responden	Jumlah data terlalu banyak	Double
Tinggi	Tinggi dari responden	Jumlah data terlalu banyak	Double

Berat	Berat dari responden	Jumlah data terlalu banyak	Double

Untuk label digunakan pada variabel kelas NObesity dibuat dengan 7 kategori, yaitu nilai-nilai:

a. InsufficientWeight (Berat Badan Kurang)
b. *NormalWeight (*Berat Badan Normal)
c. *Overweight Level I (*Kegemukan Tingkat I)
d. *Overweight Level II (*Kegemukan Tingkat II)
e. *Obesity Type I (*Obesitas Tipe I)
f. *Obesity Type II (*Obesitas Tipe II)
g. *Obesity Type III (*Obesitas Tipe III)

Data tersebut berisi data numerik dan data kontinyu, sehingga dapat digunakan untuk analisis berbasis pada algoritma klasifikasi, prediksi, segmentasi dan asosiasi. Berikut adalah struktur dataset yang digunakan:

```
df.show(4,False)
```

```
+------+----+------+------+-----------------------------+----+----+---+----------+-----+----+---+---+---+----------+---------------------+-----------------+
|Gender|Age |Height|Weight|family_history_with_overweight|FAVC|FCVC|NCP|CAEC      |SMOKE|CH2O|SCC|FAF|TUE|CALC      |MTRANS               |NObeyesdad       |
+------+----+------+------+-----------------------------+----+----+---+----------+-----+----+---+---+---+----------+---------------------+-----------------+
|Female|21.0|1.62  |64.0  |yes                          |no  |2.0 |3.0|Sometimes|no   |2.0 |no |0.0|1.0|no        |Public_Transportation|Normal_Weight    |
|Female|21.0|1.52  |56.0  |yes                          |no  |3.0 |3.0|Sometimes|yes  |3.0 |yes|3.0|0.0|Sometimes |Public_Transportation|Normal_Weight    |
|Male  |23.0|1.8   |77.0  |yes                          |no  |2.0 |3.0|Sometimes|no   |2.0 |no |2.0|1.0|Frequently|Public_Transportation|Normal_Weight    |
|Male  |27.0|1.8   |87.0  |no                           |no  |3.0 |3.0|Sometimes|no   |2.0 |no |2.0|0.0|Frequently|Walking              |Overweight_Level_I|
+------+----+------+------+-----------------------------+----+----+---+----------+-----+----+---+---+---+----------+---------------------+-----------------+
only showing top 4 rows
```

D. Praproses

Pada penggunaan dataset ini, dilakukan beberapa tahapan praproses dengan tujuan untuk mempersiapkan data yang bersih dan siap guna sehingga bisa menghasilkan nilai evaluasi terbaik, dengan penjelasan sebagai berikut:

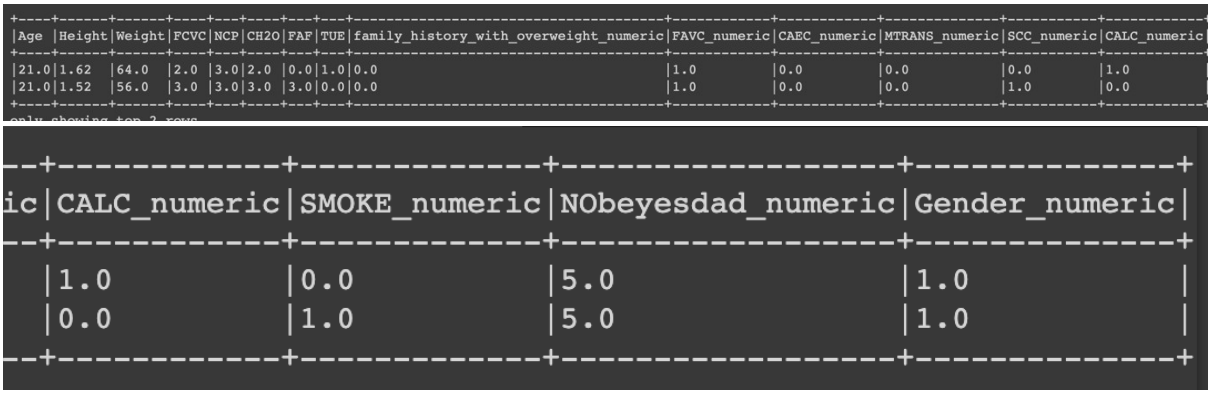
1. Melakukan pengecekan nilai NULL untuk keseluruhan dataset dengan tujuan untuk memastikan bahwa dataset tidak memiliki nilai kosong. Pada proses ini, tidak didapatkan nilai NULL sehingga tidak terjadi perubahan jumlah data baik dari sebelum maupun sesudah melakukakan *dropping data.*

```
print("Jumlah data sebelum drop null:", df.count())
df = df.na.drop("any")
print("Jumlah data setelah drop null : ",df.count())
```

```
Jumlah data sebelum drop null: 2111
Jumlah data setelah drop null :  2111
```

2. Melakukan konversi data kategorikal menjadi data numerical untuk 9 fitur menggunakan library one hot encoder, yaitu untuk fitur:
   - Family_history_with_overweight
   - FAVC;
   - CAEC;
   - MTRANS;
   - SCC;
   - CALC;
   - SMOKE;
   - NObeyesdad;
   - Gender.

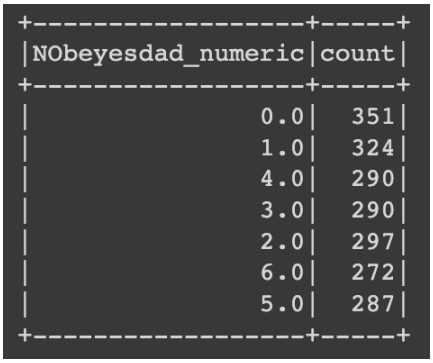
   Proses ini dilakukan agar dataset dapat diproses oleh algoritma yang nantinya akan digunakan. *Output* yang didapatkan setelah proses konversi ini adalah nilai *numerical* yang mewakili interprerasi *value* pada fitur tersebut. Berikut adalah tampilan struktur dataset yang digunakan setelah dilakukan implementasi *library one hot encoder.*



Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE	family_history_with_overweight_numeric	FAVC_numeric	CAEC_numeric	MTRANS_numeric	SCC_numeric	CALC_numeric	SMOKE_numeric	NObeyesdad_numeric	Gender_numeric
21.0	1.62	64.0	2.0	3.0	2.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	5.0	1.0
21.0	1.52	56.0	3.0	3.0	3.0	3.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	5.0	1.0

only showing top 2 rows

3. Selanjutnya, dilakukan pengecekan jumlah data untuk setiap label yaitu pada fitur Nobeyesdad_numeric. Didapatkan jumlah untuk setiap label sebagai berikut:



NObeyesdad_numeric	count
0.0	351
1.0	324
4.0	290
3.0	290
2.0	297
6.0	272
5.0	287

   Proses ini bertujuan untuk melakukan pengecekan keseimbangan jumlah data antarlabel. Karena, apabila didapati jumlah yang tidak seimbang, maka akan dilakukan *imbalanced handling.*

4. Setelah dataset sudah dipastikan seimbang dan sudah *berbentuk numerical,* selanjutnya adalah melakukan tahapan *Vector Assembler*. Tujuan dari proses ini adalah pada PySpark, semua kolom kecuali target perlu diubah menjadi vektor, yang disebut fitur. Proses ini dilakukan menggunakan bantuan library VectorAssembler.

```
+--------------------------------------------------------------------+
|features                                                            |
+--------------------------------------------------------------------+
|[1.0,21.0,1.62,64.0,0.0,1.0,2.0,3.0,0.0,0.0,2.0,0.0,0.0,1.0,1.0,0.0]|
|[1.0,21.0,1.52,56.0,0.0,1.0,3.0,3.0,0.0,1.0,3.0,1.0,3.0,0.0,0.0,0.0]|
+--------------------------------------------------------------------+
```

5. Langkah selanjutnya adalah memisahkan kolom features yang sudah berbentuk vektor dan kolom NObeyesdad_numeric untuk selanjutnya akan dijadikan data final yang masuk pada tahapan *Data Splitting*.
6. Kemudian, data final ini akan dilakukan *spli*t menjadi data *train* (latih) dan data *test* (uji) dengan rasion 70:30. Pemilihan jumlah rasio ini diambil setelah melakukan percobaan pada 80:20, dan mendapatkan hasil evaluasi akhir yang tidak sebaik 70:30.

```
[16] #To train our model, we combine "features" and "target" as input/output.
     final_data = output.select('features', 'NObeyesdad_numeric')

[50] #Then, we can split final_data to train and test as follows:
     train, test = final_data.randomSplit([0.7, 0.3])

     train.show(2, False)

     +--------------------------------------------------------------------+------------------+
     |features                                                            |NObeyesdad_numeric|
     +--------------------------------------------------------------------+------------------+
     |(16,[0,1,2,3,4,6,7,8,10],[1.0,21.0,1.52,42.0,1.0,3.0,1.0,1.0,1.0])|6.0               |
     |(16,[0,1,2,3,4,6,7,8,10],[1.0,21.0,1.52,42.0,1.0,3.0,1.0,1.0,1.0])|6.0               |
     +--------------------------------------------------------------------+------------------+
     only showing top 2 rows

[89] test.show(2, False)

     +--------------------------------------------------------------------+------------------+
     |features                                                            |NObeyesdad_numeric|
     +--------------------------------------------------------------------+------------------+
     |(16,[0,1,2,3,4,6,7,8,10],[1.0,21.0,1.52,42.0,1.0,3.0,1.0,1.0,1.0])|6.0               |
     |(16,[0,1,2,3,4,6,7,8,10],[1.0,21.0,1.52,42.0,1.0,3.0,1.0,1.0,1.0])|6.0               |
     +--------------------------------------------------------------------+------------------+
     only showing top 2 rows
```

E. Analisis Pemilihan Algoritma

Pada percobaan ini, digunakan 3 algoritma klasifikasi untuk *supervised learning* menggunakan PySpark, yaitu Logistik Regression (LR), Random Forest (RF), dan Decision Tree (DT). Dasar alasan pemilihan ketiga algoritma ini adalah, untuk Random Forest dapat mengatasi noise dan missing value serta dapat mengatasi data dalam jumlah

yang besar. Hal ini cocok dan sesuai dengan jumlah data yang termasuk kategori besar karena lebih dari 1000 baris. Untuk algoritma Logistik Regression, memiliki kelebihan lgoritma sangat fleksibel, mengambil segala jenis input, dan mendukung beberapa tugas analitik yang berbeda. Sedangkan, Decision Tree digunakan karena algoritma ini memiliki kelebihan yaitu dapat Menemukan kombinasi data yang tidak terduga dan dapat menghilangkan perhitungan yang tidak diperlukan. Dikarenakan dengan metode ini sampel hanya akan diuji berdasarkan kriteria ataupun kelas tertentu.

F. Analisis Penentuan Parameter

Parameter yang digunakan pada algoritma eksperimen yaitu Logistik Regression (LR), Random Forest (RF), dan Decision Tree (DT), adalah sebagai berikut:

1. LR

   Pada algoritma LR digunakan parameter default sesuai dengan dokumentasi LR untuk PySpark. Pada eksperimen ini tidak dilakukan skema lain maupun modifikasi dari parameter LR.

2. RF

   Pada algoritma RF digunakan parameter default sesuai dengan dokumentasi RF untuk PySpark. Pada eksperimen dilakukan skema lain maupun modifikasi dari parameter LR terhadap beberapa parameter berikut:
   - Seed = 2500 (Default: None)
   - maxDepth = 10 (Default: 4)
   - numTrees = 30

3. DT
   - maxDepth = 2 (Default: 5)

G. Hasil Percobaan

Pada studi kasus ini, dilakukan beberapa eksperimen yaitu membandingkan 3 algoritma yang digunakan, yaitu Logistik Regression, Random Forest, dan Decision Tree. Berikut adalah rangkuman hasil evaluasi matrix yang didapatkan:

Data Type	Algoritma	Accuracy	Precision	Recall	F1-Scoree
Train	Logistik Regression	89.22%	89.21%	89.14%	89.22%
Test		89.14%	89.13%	88.98%	89.14%
Train	Random Forest	**99.79%**	**99.79%**	**99.79%**	**99.79%**
Test		**93.51%**	**93.58%**	**93.53%**	**93.51%**
Train	Decision Tree	54.62%	37.38%	42.79%	54.62%
Test		55.52%	37.80%	43.22%	55.20%

Pada hasil tabel di atas, dapat diketahui bahwa Algoritma Random Forest memiliki nilai yang paling tinggi untuk semua matriks evaluasi dan untuk seluruh algoritma yang digunakan. Hal ini dikarenakan, algoritma RF dapat mengatasi permasalahan overfitting yang sering dialami pada algoritma DT. Pembuktian ini juga diperkuat dengan perbandingan hasil dari algoritma RF dan DT. Algorima RF mampu menghasilkan akurasi yang nyaris sempurna yaitu 99.79% untuk data latih dan 93.51% untuk data uji, sedangkan pada algoritma DT hanya didapatkan akurasi 54.62% untuk data latih dan 55.52% untuk data uji. Sedangkan, untuk algoritma LR didapatkan hasil yang cukup baik dibandingkan dengan DT, yaitu 89.22% untuk data latih, dan 89.14% untuk data uji. Hal ini dikarenakan dataset yang digunakan juga dapat diimplementasikan untuk proses prediksi ataupun regression. Selain itu, algoritma ini juga dapat memberikan hasil *insight* bahwa berhasilya terindentifikasi pengaruh antar *variable predictor* atau *variable* independen terhadap *variable* lainnya atau *variable* dependen.

Hal ini berarti, algoritma RF dapat membantu dalam proses pencapaian tujuan akhir dari penelitian ini, yaitu memprediksi dan melakukan klasifikasi terhadap dataset yang di*input*-kan yang kemudian akan mengeluarkan kategori obesitas dari individu tersebut. Karena RF dapat mengatasi permasalahan yang dihadapi oleh kedua algoritma perbandingan yang lain.

H. Ringkasan Model Yang Diperoleh

Pada penelitian ini, untuk nilai Accuracy, Recall, Precision, dan F1-Score terbaik diungguli oleh **Algoritma Random Forest** baik untuk data train maupun data test. Dengan selisih:

- 10.57% untuk data train Algoritma LR
- 4.37% untuk data test Algoritma LR
- 45.17% untuk data train Algoritma DT
- 37.99 untuk data test Algoritma DT

I. Interpretasi Model

Dari penelitian ini, dapat diketahui bahwa algoritma Random Forest dapat dengan tepat dan tergolong akurat dalam melakukan prediksi maupun klasifikasi untuk dataset yang digunakan denan tujuan untuk mengeluarkan output label kategori tingkatan obesitas untuk orang-orang dari negara-negara Meksiko, Peru dan Kolombia. Jumlah data tingkat obesitas yang berhasil diprediksi oleh algoritma RF adalah sebagai berikut:

prediction	count
0.0	92
1.0	103
4.0	96
3.0	86
2.0	95
6.0	95
5.0	96

Didapatkan informasi bahwa, dari ke-7 kategori obesitas, jumlah paling banyak diperoleh oleh kelas 1 yaitu *NormalWeight (*Berat Badan Normal). Artinya, pada negara meksiko, peru, dan kolombia masih didominasi oleh masyarakat dengan berat badan normal. Namun, jumlah ini memiliki selisih yang tidak jauh berbeda dengan ketegori obesitas yang lainnya. Artinya juga, masih ada atau bahkan banyak masyarakat di negara tersebut yang mengalami obesitas maupun kekurangan gizi, walaupun berat badan normal masih mendominasi.