Docare

Project 2

Our Team

BANYU

Product Manager and Business Analyst

MELLISA

Data Scientist

ARIFA

Full Stack Developer

PRISSY

Data Modelling Specialist

TAUFIQ

Data Engineer

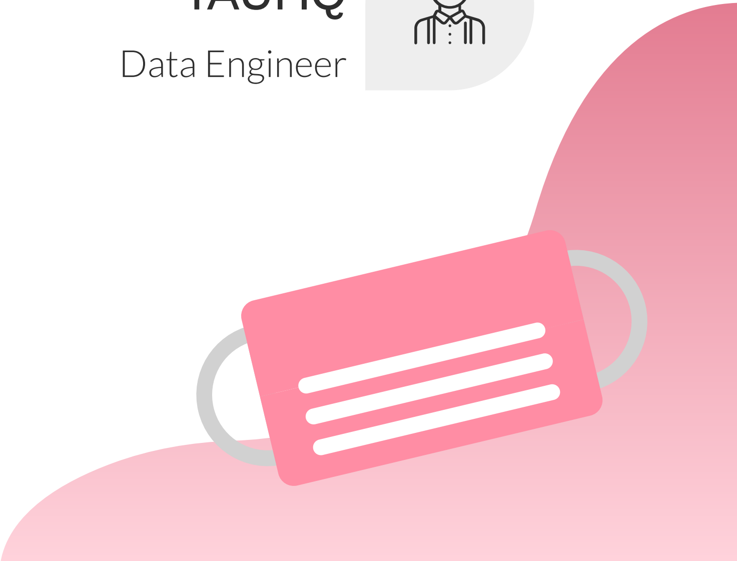


Table of Contents

01
Business Understanding

02
Tujuan

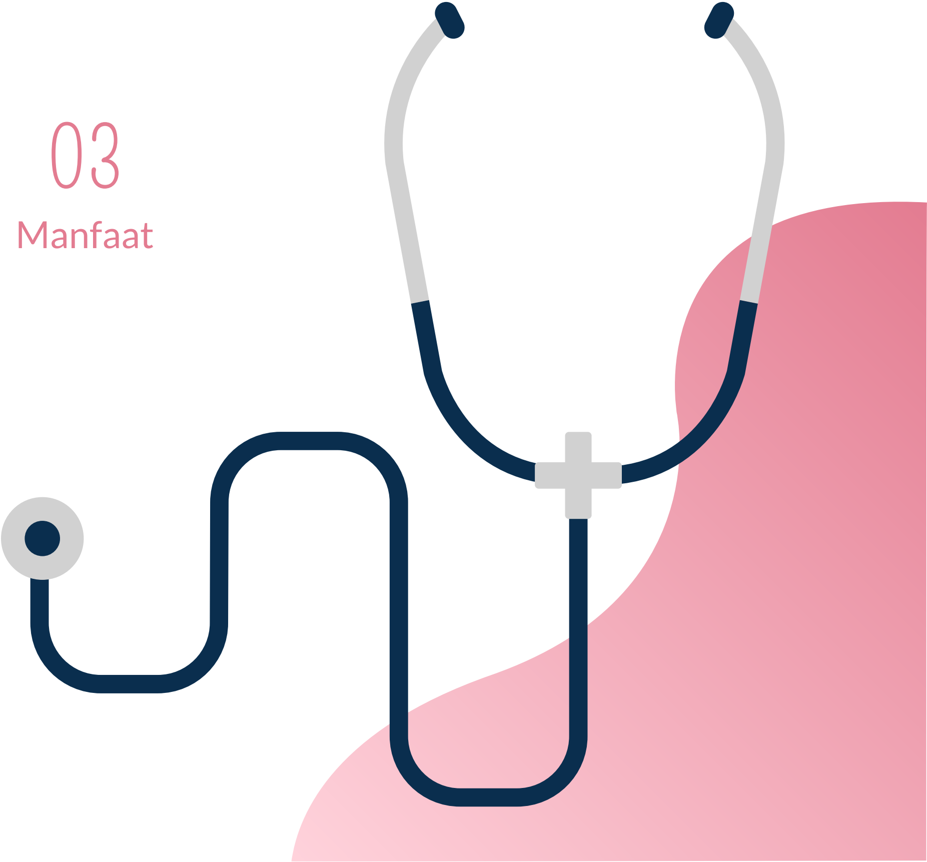
03
Manfaat

04
Data Understanding

05
Data Preparation

06
Modelling

07
Evaluation

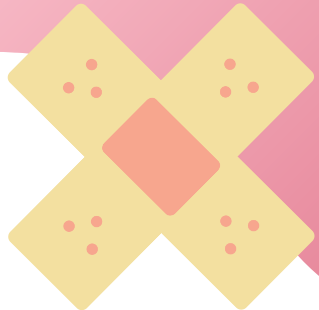


01

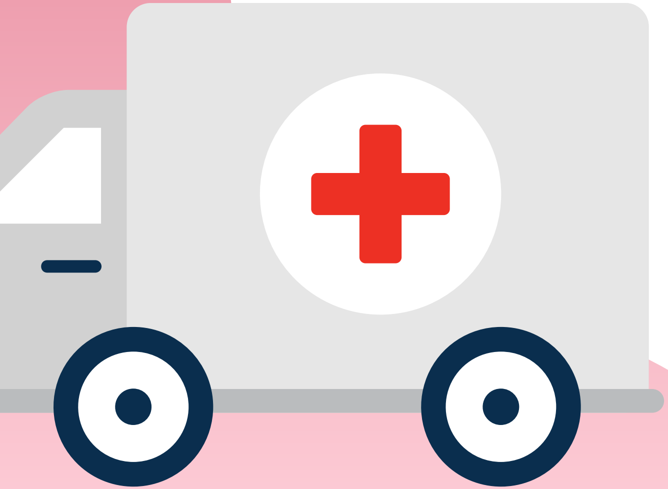
Business Understanding

Docare merupakan sebuah perusahaan yang bergerak dalam bidang kesehatan, dimana saat ini bidang kesehatan merupakan garda terdepan dalam menanggulangi penyakit. Berfokus pada menganalisis data penyakit, Docare berinovasi dan meningkatkan pelayanan guna memenuhi kebutuhan di bidang kesehatan ini.

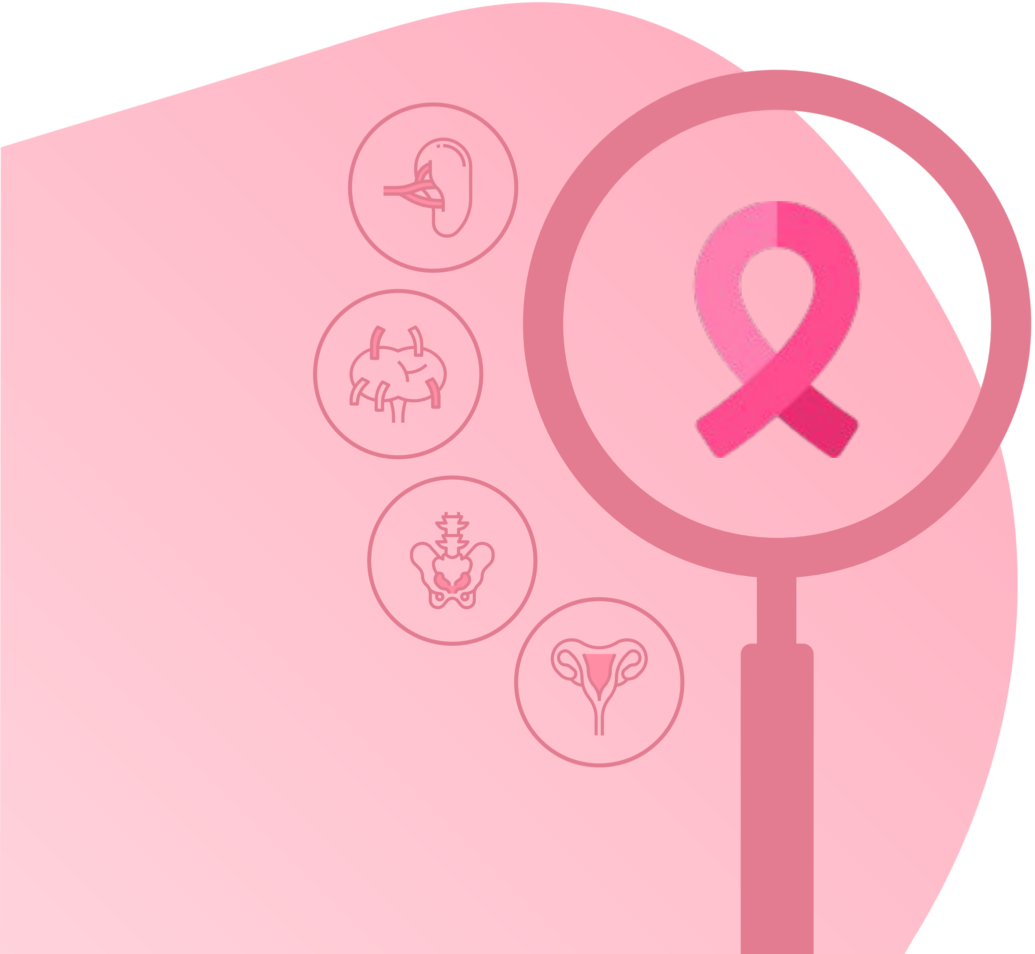
Tim Data Science kami akan membantu Anda untuk mengklasifikasikan penyakit dalam mengetahui tingkatan penyakit yang diderita seseorang.



02 Tujuan



Mengetahui tingkatan penyakit yang diderita seorang pasien kanker payudara dengan menggunakan klasifikasi. Hasil klasifikasi ini kemudian akan dianalisa tingkatan kanker tersebut. Kemudian akan dibentuk pengklasifikasian apakah kanker tersebut termasuk kanker payudara jinak atau ganas.



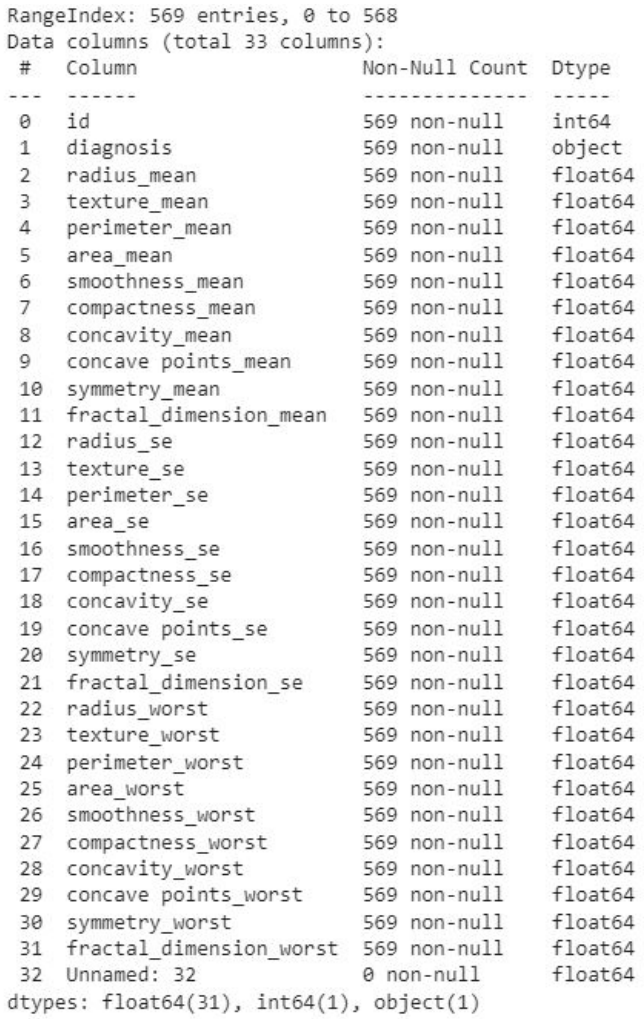
03 Manfaat

Mengetahui penyakit kanker payudara yang diderita seseorang berdasarkan klasifikasi kanker tersebut malignant (ganas) atau benign (jinak).

04

Data Understanding

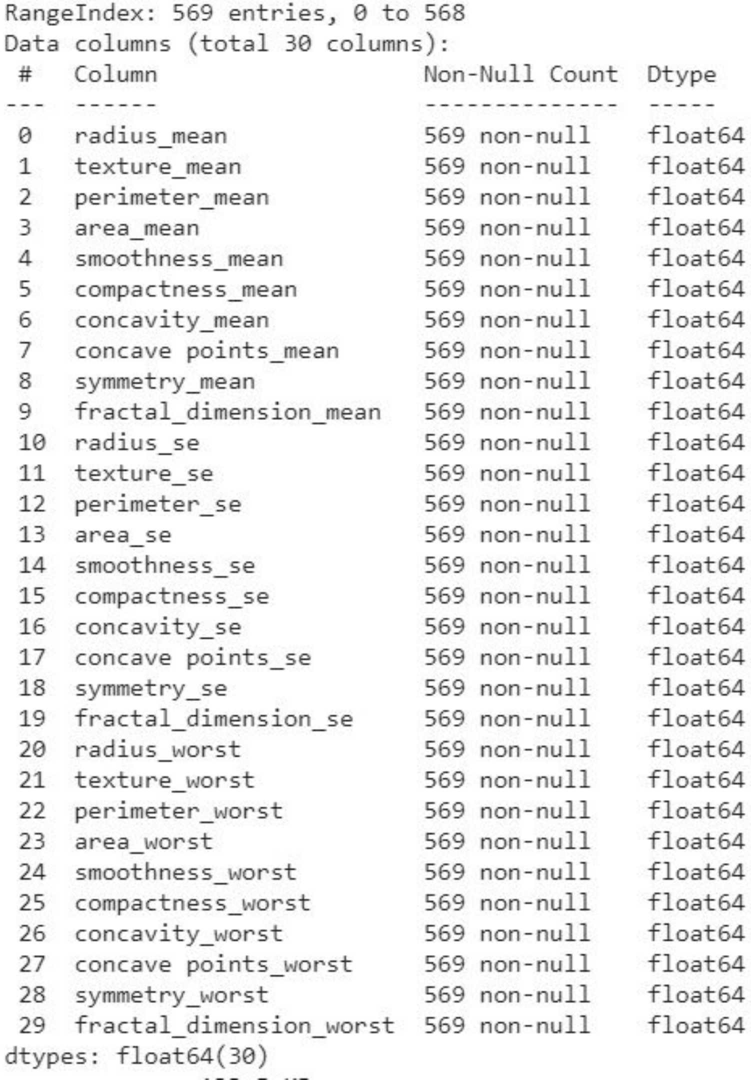
Menggunakan Breast Cancer Wisconsin (Diagnostic) Data Set untuk dilakukan pengkasifikasian kanker.



```
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
```

05 Data Preparation

Data yang kami gunakan mencakup semua kolom, kecuali untuk kolom id, diagnosis dan kolom ke-33 tidak kami masukan karena tidak memiliki nama dan datanya juga tidak tersedia.

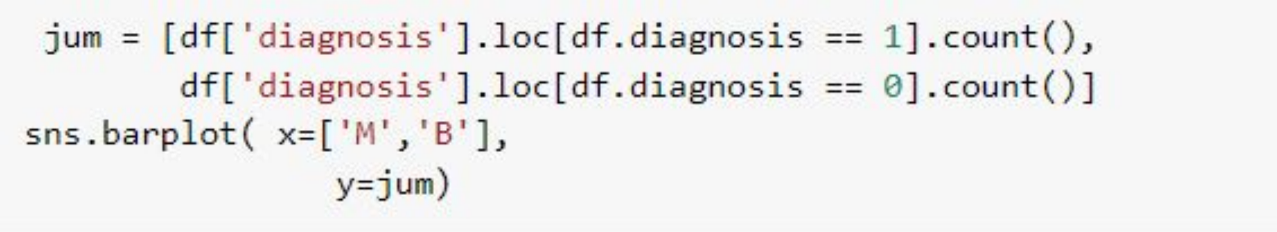


```
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   radius_mean              569 non-null    float64
 1   texture_mean             569 non-null    float64
 2   perimeter_mean           569 non-null    float64
 3   area_mean                569 non-null    float64
 4   smoothness_mean          569 non-null    float64
 5   compactness_mean         569 non-null    float64
 6   concavity_mean           569 non-null    float64
 7   concave points_mean      569 non-null    float64
 8   symmetry_mean            569 non-null    float64
 9   fractal_dimension_mean   569 non-null    float64
 10  radius_se                569 non-null    float64
 11  texture_se               569 non-null    float64
 12  perimeter_se             569 non-null    float64
 13  area_se                  569 non-null    float64
 14  smoothness_se            569 non-null    float64
 15  compactness_se           569 non-null    float64
 16  concavity_se             569 non-null    float64
 17  concave points_se        569 non-null    float64
 18  symmetry_se              569 non-null    float64
 19  fractal_dimension_se     569 non-null    float64
 20  radius_worst             569 non-null    float64
 21  texture_worst            569 non-null    float64
 22  perimeter_worst          569 non-null    float64
 23  area_worst               569 non-null    float64
 24  smoothness_worst         569 non-null    float64
 25  compactness_worst        569 non-null    float64
 26  concavity_worst          569 non-null    float64
 27  concave points_worst     569 non-null    float64
 28  symmetry_worst           569 non-null    float64
 29  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30)
```

06 Modelling

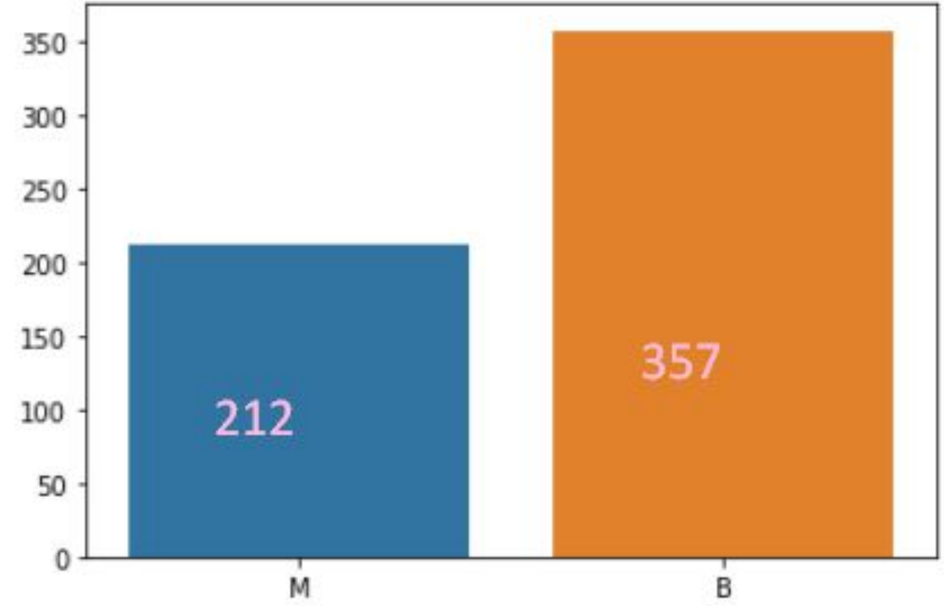
“Algoritma yang kami gunakan adalah KNN (K-Nearest Neighbour)”

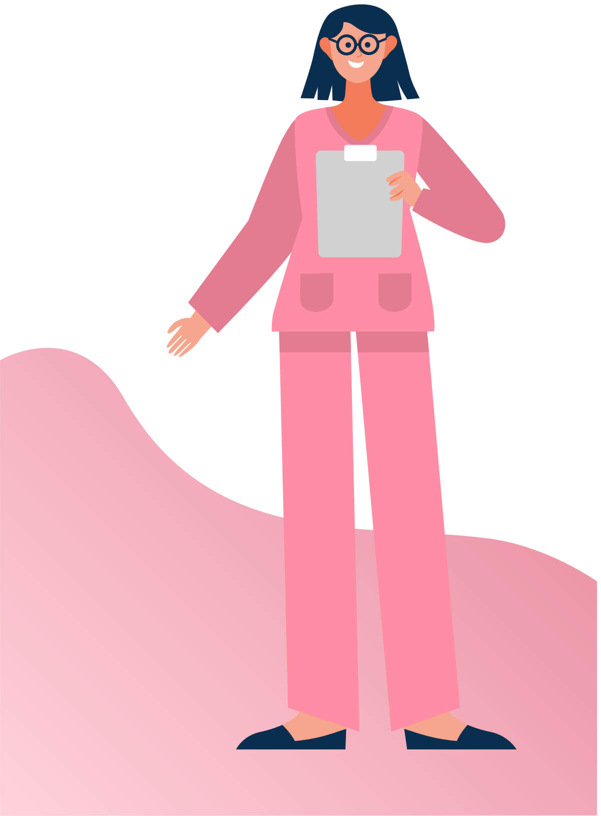
“Karena data tersebut bersifat multi-variabel”

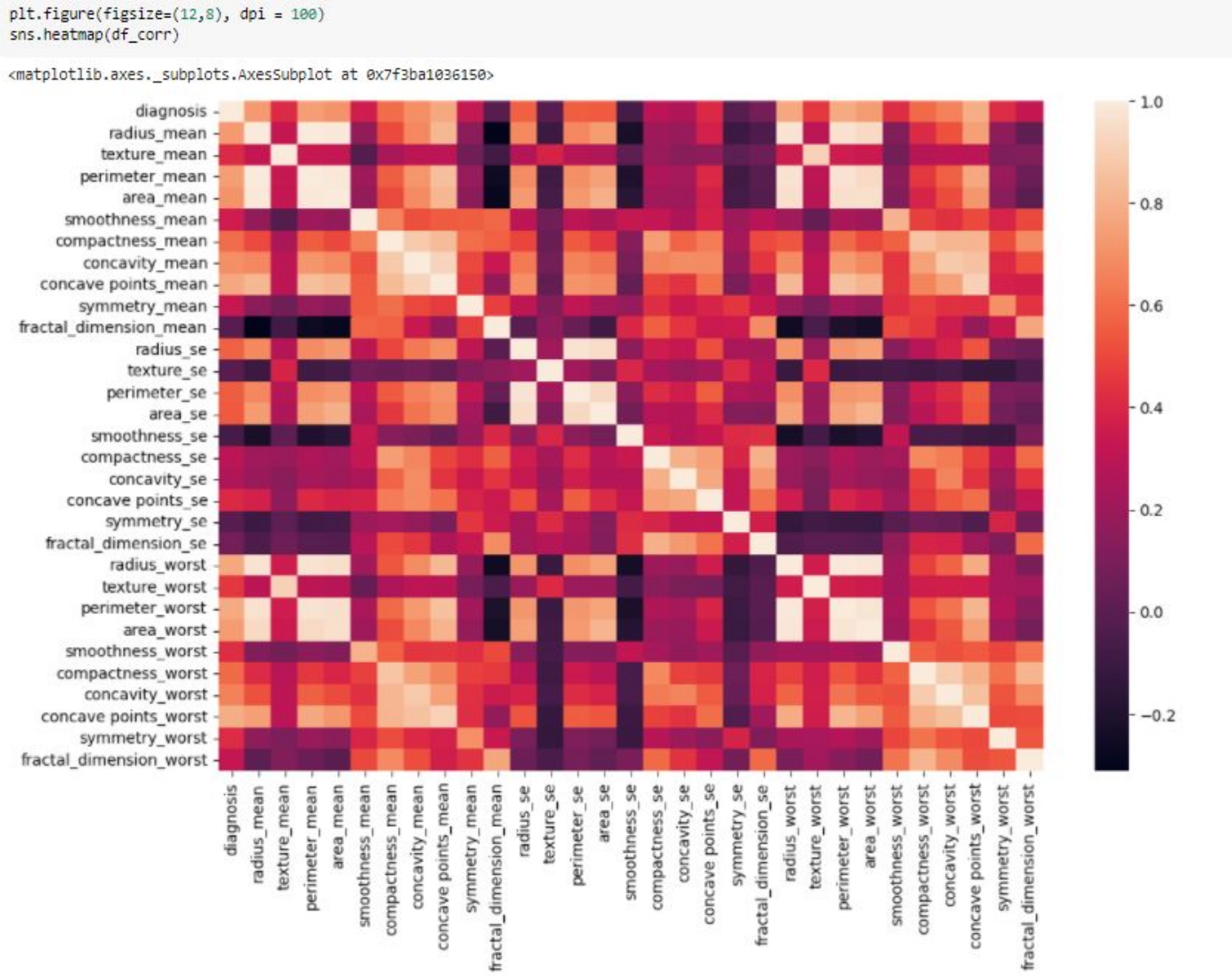


```
 jum = [df['diagnosis'].loc[df.diagnosis == 1].count(),
        df['diagnosis'].loc[df.diagnosis == 0].count()]
sns.barplot( x=['M','B'],
                y=jum)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3b9f29b190>
```

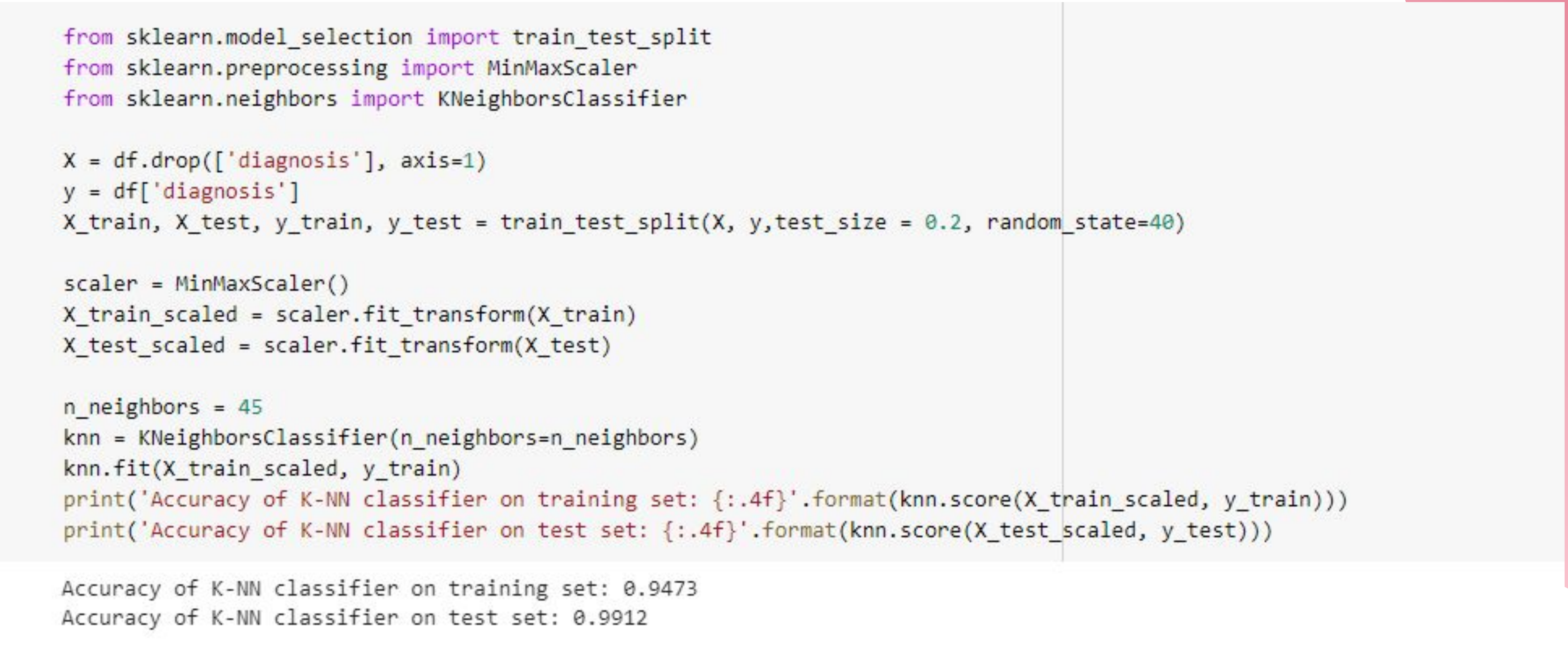






```
plt.figure(figsize=(12,8), dpi = 100)
sns.heatmap(df_corr)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3ba1036150>
```



```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.neighbors import KNeighborsClassifier

X = df.drop(['diagnosis'], axis=1)
y = df['diagnosis']
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.2, random_state=40)

scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.fit_transform(X_test)

n_neighbors = 45
knn = KNeighborsClassifier(n_neighbors=n_neighbors)
knn.fit(X_train_scaled, y_train)
print('Accuracy of K-NN classifier on training set: {:.4f}'.format(knn.score(X_train_scaled, y_train)))
print('Accuracy of K-NN classifier on test set: {:.4f}'.format(knn.score(X_test_scaled, y_test)))
```

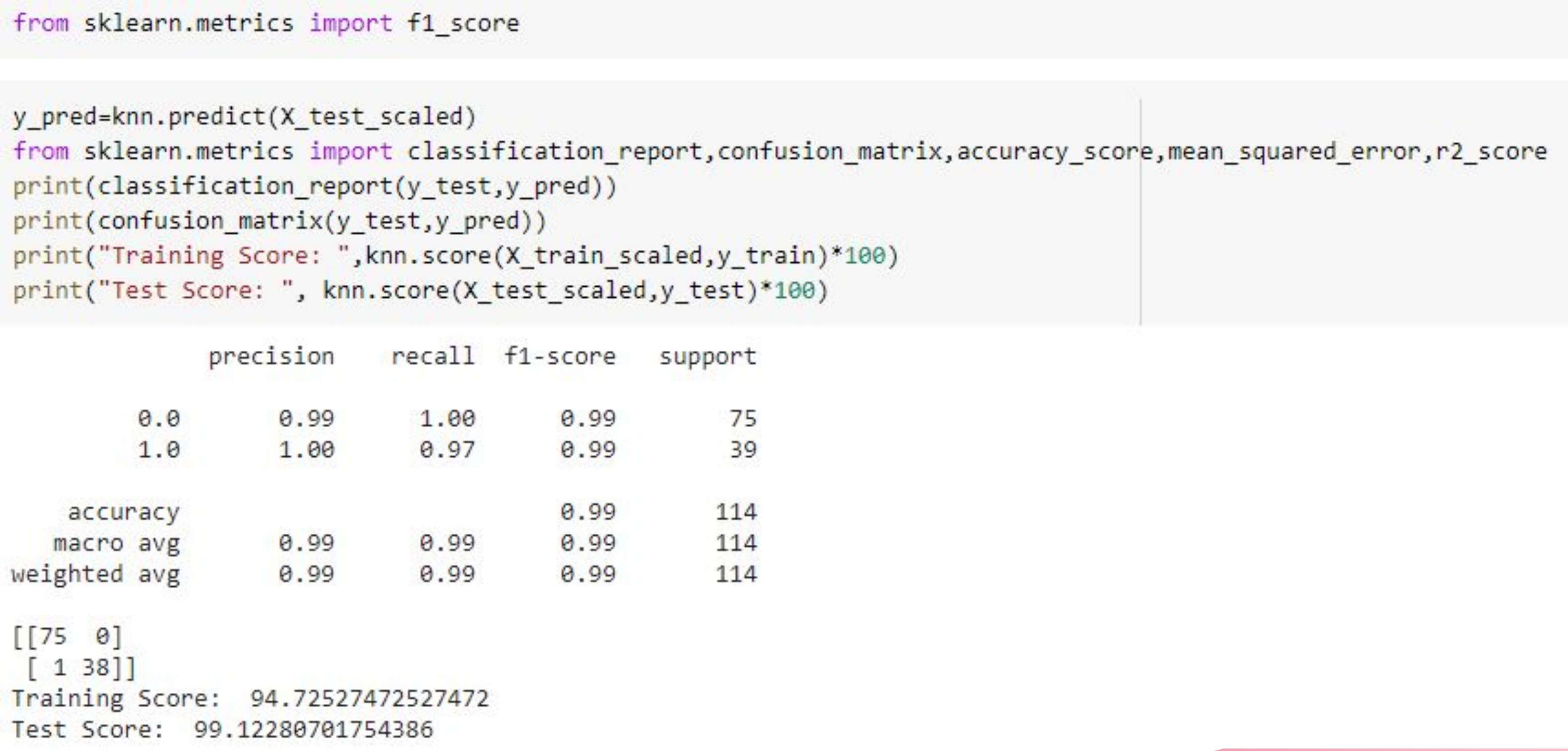
```
Accuracy of K-NN classifier on training set: 0.9473
Accuracy of K-NN classifier on test set: 0.9912
```

07 Evaluation

01 Akurasi cukup bagus: 0.993/1

02 Akurasi antara ***data training*** dengan ***data testing*** tidak berbeda jauh (tidak overfit maupun underfit).

03 Maka, model yang dibuat sudah cukup baik memprediksikan apakah tumor kanker tsb. ***malignant*** atau ***benign*** dari 30 variabel.



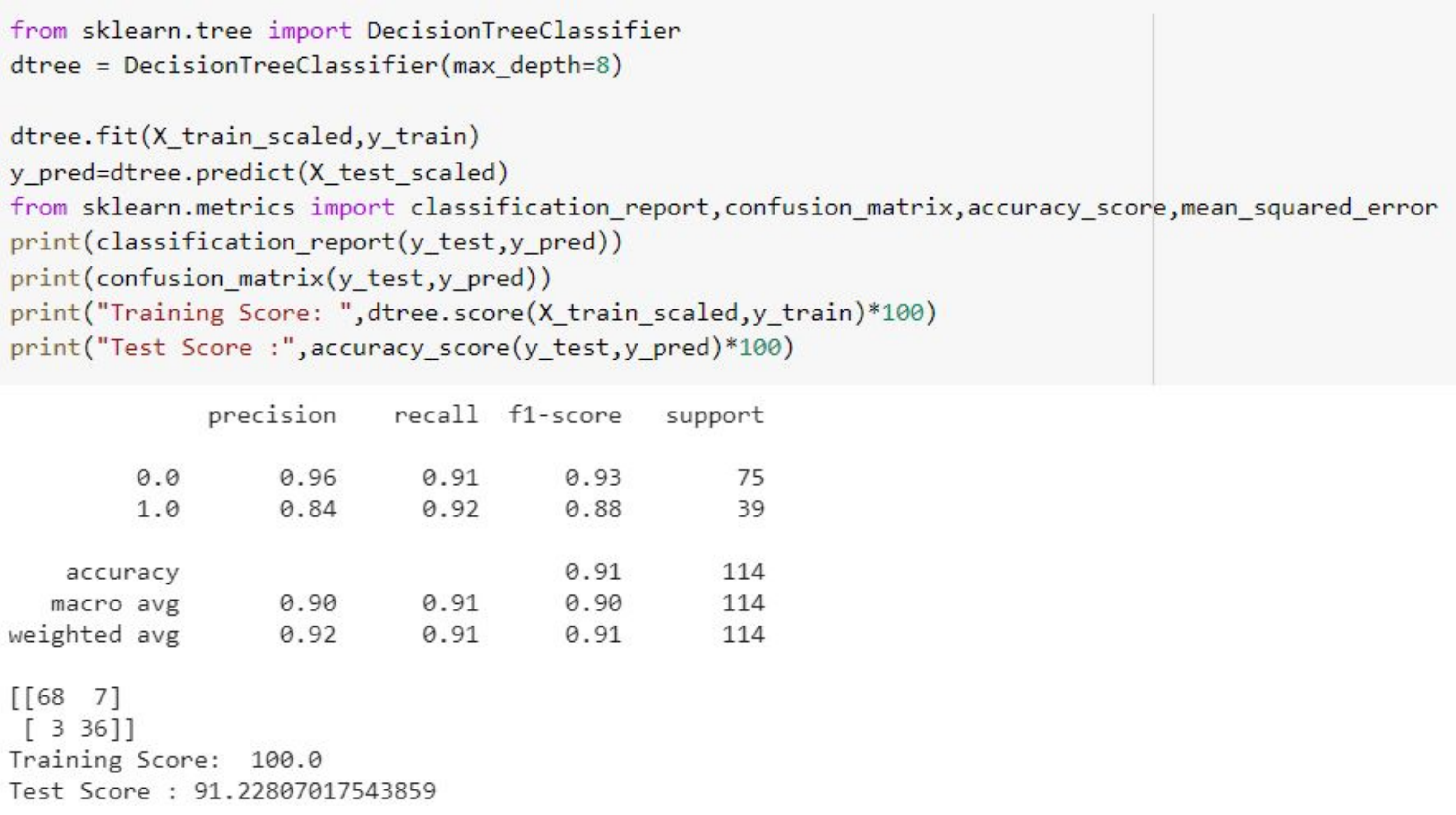
```
from sklearn.metrics import f1_score
```

```
y_pred=knn.predict(X_test_scaled)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,mean_squared_error,r2_score
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print("Training Score: ",knn.score(X_train_scaled,y_train)*100)
print("Test Score: ", knn.score(X_test_scaled,y_test)*100)
```

```
              precision    recall  f1-score   support

         0.0       0.99      1.00      0.99        75
         1.0       1.00      0.97      0.99        39

    accuracy                           0.99       114
   macro avg       0.99      0.99      0.99       114
weighted avg       0.99      0.99      0.99       114

[[75  0]
 [ 1 38]]
Training Score:   94.72527472527472
Test Score:   99.12280701754386
```

Model KNN yang dibuat berhasil memprediksi dengan benar 38 kasus kanker payudara dari total 39 kasus yang ada

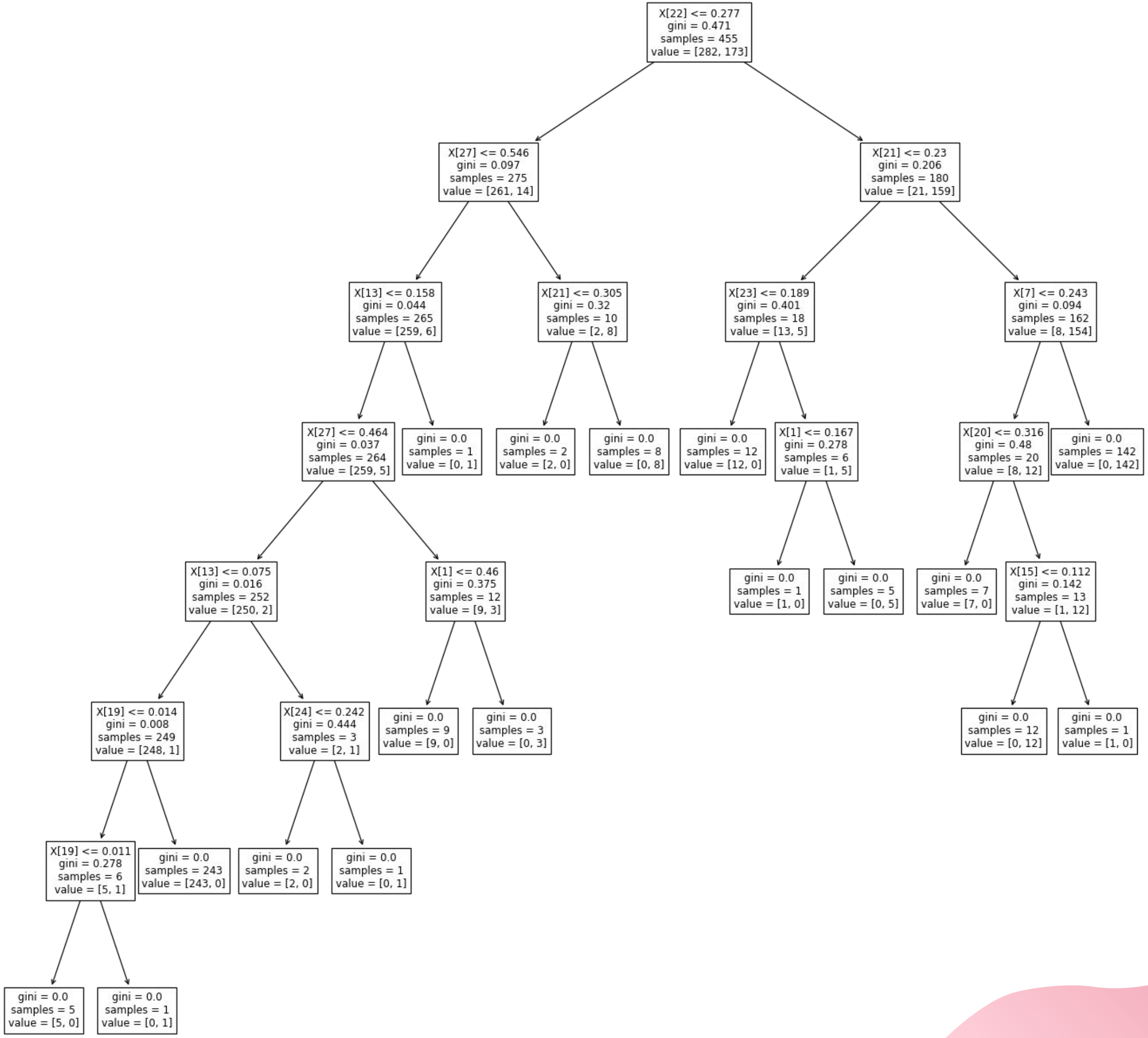
Perbandingan dengan Decision Tree



```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(max_depth=8)

dtree.fit(X_train_scaled,y_train)
y_pred=dtree.predict(X_test_scaled)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,mean_squared_error
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print("Training Score: ",dtree.score(X_train_scaled,y_train)*100)
print("Test Score :",accuracy_score(y_test,y_pred)*100)
```

```
              precision    recall  f1-score   support

         0.0       0.96      0.91      0.93        75
         1.0       0.84      0.92      0.88        39

    accuracy                           0.91       114
   macro avg       0.90      0.91      0.90       114
weighted avg       0.92      0.91      0.91       114

[[68  7]
 [ 3 36]]
Training Score:  100.0
Test Score : 91.22807017543859
```

Visualisasi



References

- https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/code
- https://colab.research.google.com/drive/1hhRm4C1ZGYEbbBoTWpz1SE9UoGjNmzVJ?usp=sharing

Thanks

Please give us critics and suggestions ^^