Nama : Shinta Maimunah Heril'in

Kelas : 3B

NIM : 361955401060

Rancanglah sebuah sistem cerdas (intelligent system) dari data public (mengunduh data di kaggle) atau dengan data yang anda miliki sendiri. Anda dapat menerapkan kasus klasifikasi atau klasterisasi!

a. Deskripsikan domain masalah yang anda selesaikan dan deskripsikan juga dataset yang anda gunakan beserta teknik pengambilan datanya! (max poin 30)

Domain masalah yang akan diselesaikan pada tugas ini yaitu mengkategorikan kasus penyakit kanker payudara (breast cancer). Kategori dibagi menjadi dua, yakni Malignant (ganas) dan benign (jinak). Malignant didasarkan pada pertumbuhan sel yang dapat mengganggu atau merusak jaringan didekatnya, sedangkan benign didasarkan pada pertumbuhan yang tidak mengganggu jaringan lain. Dataset diambil dengan API dari Kaggle. Fitur yang digunakan pada dataset ini yaitu radius_mean, perimeter_mean, area_mean, compactness_mean, dan concave points_mean. Berikut kode program untuk mengambil dataset

```
In [1]: !kaggle datasets list -s "breast cancer"
```

```
ref                                                             title                                                size  lastUp
dated          downloadCount  voteCount  usabilityRating
-------------------------------------------------------------  ---------------------------------------------------  -----  ------
-------------  -------------  ---------  ---------------
uciml/breast-cancer-wisconsin-data                              Breast Cancer Wisconsin (Diagnostic) Data Set        49KB  2016-0
9-25 10:49:04         186355       2512  0.85294116
piotrgrabo/breastcancerproteomes                                Breast Cancer Proteomes                               5MB  2019-1
1-14 05:15:12          11391        333  0.64705884
merishnasuwal/breast-cancer-prediction-dataset                  Breast Cancer Prediction Dataset                      8KB  2018-0
9-26 12:41:51          14228        178  0.8235294
amandam1/breastcancerdataset                                    Real Breast Cancer Data                              11KB  2021-0
8-05 17:58:17           1732         39  1.0
yuqing01/breast-cancer                                          breast cancer                                        49KB  2017-1
0-23 11:21:30           3216         47  0.4117647
paultimothymooney/breast-histopathology-images                  Breast Histopathology Images                          3GB  2017-1
2-19 05:46:40          32806        702  0.75
kmader/mias-mammography                                         MIAS Mammography                                    312MB  2017-1
1-01 10:50:49           7545        159  0.75
roustekbio/breast-cancer-csv                                    Wisconsin Breast Cancer Database                     6KB  2017-1
0-30 18:47:52           2942         27  0.7647059
raghadalharbi/breast-cancer-gene-expression-profiles-metabric   Breast Cancer Gene Expression Profiles (METABRIC)    3MB  2020-0
5-26 20:08:07           1944         51  0.9411765
gilsousa/habermans-survival-data-set                            Haberman's Survival Data Set                        998B  2016-1
1-30 18:04:33          24414        504  0.7058824
sarahvch/breast-cancer-wisconsin-prognostic-data-set            Breast Cancer Wisconsin (Prognostic) Data Set        49KB  2017-0
3-31 22:47:50           2808         44  0.8235294
brunogrisci/breast-cancer-gene-expression-cumida                Breast cancer gene expression - CuMiDa               62MB  2020-0
2-01 10:51:48           1038         36  0.9117647
gunesevitan/breast-cancer-metabric                              Breast Cancer (METABRIC)                             77KB  2020-1
2-21 17:22:56            699         23  0.7941176
```

```
In [2]: !kaggle datasets download iabhishekbhardwaj/breast-cancer-prediction
```

```
breast-cancer-prediction.zip: Skipping, found more recently modified local copy (use --force to force download)
```

b. Lakukan pengembangan model (klasifikasi atau klasterisasi) dan tuliskan syntax programnya menggunakan bahasa pemrograman python (anda bisa menggunakan referensi dari materi pada pertemuan sebelumnya)! (max poin 40)

```
In [1]: import pandas as pd

        import matplotlib.pyplot as plt

        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LogisticRegression
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.tree import DecisionTreeClassifier, export_graphviz
        from sklearn import metrics
        from sklearn import tree
```

```
In [2]: df = pd.read_csv("data.csv",header = 0)
        df.head()
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	te
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 33 columns

In [3]:
```
from sklearn.model_selection import train_test_split

traindf, testdf = train_test_split(df, test_size = 0.7)
```

In [4]:
```
print("Banyak data latih setelah dilakukan Train-Validation Split: ", len(traindf))
print("Banyak data uji setelah dilakukan Train-Validation Split: ", len(testdf))
```

```
Banyak data latih setelah dilakukan Train-Validation Split:  170
Banyak data uji setelah dilakukan Train-Validation Split:  399
```

In [5]:
```
def classification_model(model, data, predictors, outcome):
  #Fit the model:
  model.fit(data[predictors],data[outcome])

  #Make predictions on training set:
  predictions = model.predict(data[predictors])

  #Print accuracy
  accuracy = metrics.accuracy_score(predictions,data[outcome])
  print("Accuracy : %s" % "{0:.3%}".format(accuracy))

  #Fit the model again so that it can be refered outside the function:
  model.fit(data[predictors],data[outcome])
```

In [6]:
```
# Logistic Regression

predictor_var = ['radius_mean','perimeter_mean','area_mean','compactness_mean','concave points_mean']
outcome_var='diagnosis'
model=LogisticRegression()
classification_model(model,traindf,predictor_var,outcome_var)
```

```
Accuracy : 90.000%
```

In [7]:
```
# Decision Tree

from sklearn.tree import DecisionTreeClassifier
predictor_var = ['radius_mean','perimeter_mean','area_mean','compactness_mean','concave points_mean']
model = DecisionTreeClassifier()
classification_model(model,traindf,predictor_var,outcome_var)
```

```
Accuracy : 100.000%
```

In [8]:
```
# Naive Bayes

from sklearn import naive_bayes
predictor_var = ['radius_mean','perimeter_mean','area_mean','compactness_mean','concave points_mean']
model = naive_bayes.BernoulliNB()
classification_model(model,traindf,predictor_var,outcome_var)
```

```
Accuracy : 66.471%
```

In [9]:
```
# kNN

from sklearn.neighbors import KNeighborsClassifier
predictor_var = ['radius_mean','perimeter_mean','area_mean','compactness_mean','concave points_mean']
model = KNeighborsClassifier()
classification_model(model,traindf,predictor_var,outcome_var)
```

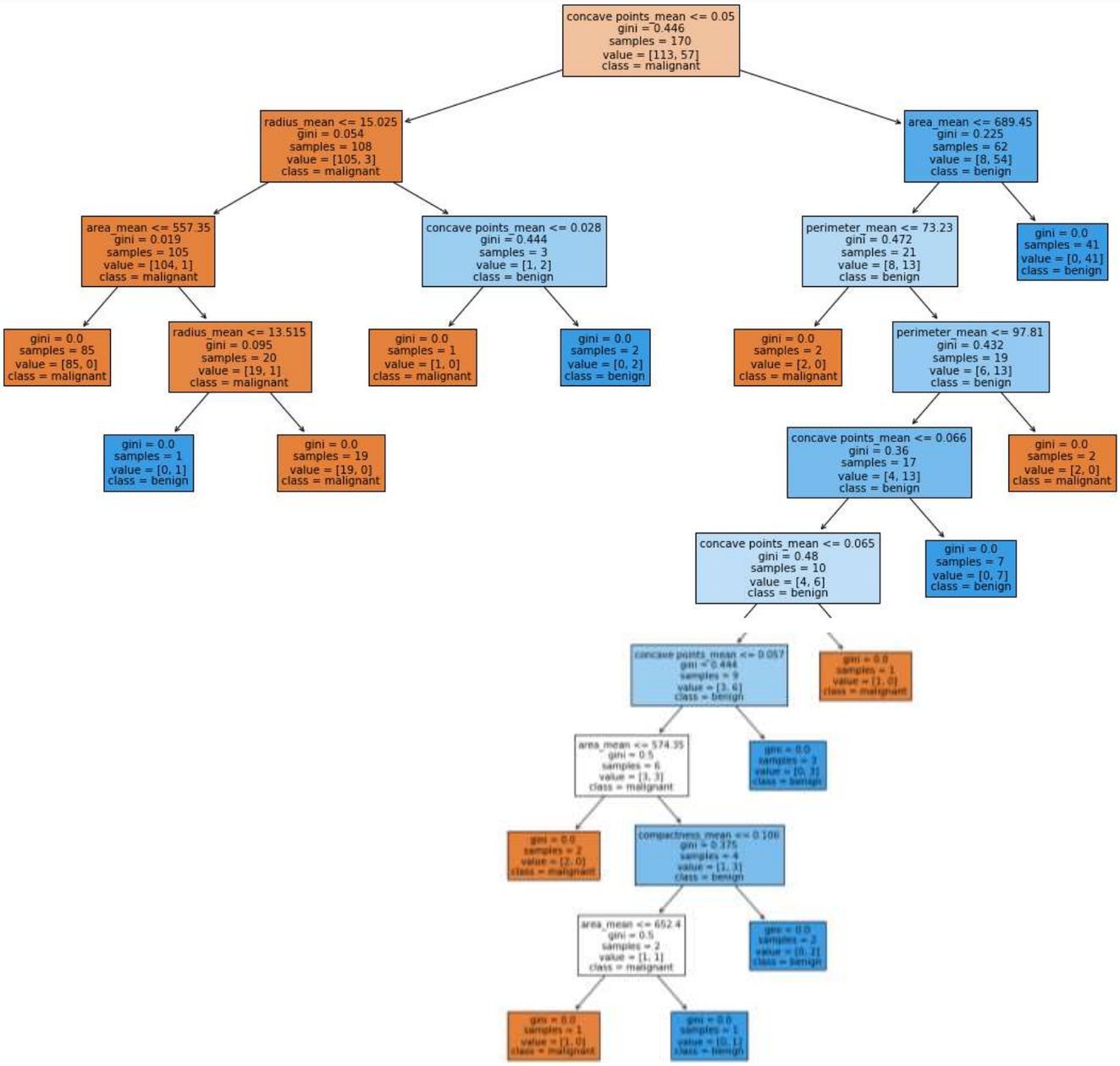
```
Accuracy : 92.941%
```

c. Deskripsikan tentang metode atau pendekatan yang anda gunakan dan bagaimana hasil performansi sistemnya (bisa menggunakan nilai akurasi, presisi, dan metode pengujian sistem yang sesuai)! (max poin 30)

Metode yang digunakan untuk percobaan menggunakan metode Logistic Regression, Decision Tree, Naive Bayes, dan kNN. Dari keempat metode tersebut didapatkan hasil akurasi 100% untuk metode Decision Tree. Namun metode Decision Tree bukanlah metode yang paling baik untuk digunakan pada semua kasus klasifikasi, karena setiap dataset yang digunakan memiliki nilai data yang berbeda sehingga nilai akurasi yang dihasilkan juga berbeda.

In [10]:
```
from sklearn.tree import DecisionTreeClassifier
predictor_var = ['radius_mean','perimeter_mean','area_mean','compactness_mean','concave points_mean']
model = DecisionTreeClassifier()
classification_model(model,traindf,predictor_var,outcome_var)
cn=['malignant', 'benign',]
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (20,20))
tree.plot_tree(model,
               feature_names = predictor_var,
               class_names=cn,
               filled = True);
```

```
Accuracy : 100.000%
```


concave points_mean <= 0.05
gini = 0.446
samples = 170
value = [113, 57]
class = malignant
radius_mean <= 15.025
gini = 0.054
samples = 108
value = [105, 3]
class = malignant
area_mean <= 689.45
gini = 0.225
samples = 62
value = [8, 54]
class = benign
area_mean <= 557.35
gini = 0.019
samples = 105
value = [104, 1]
class = malignant
concave points_mean <= 0.028
gini = 0.444
samples = 3
value = [1, 2]
class = benign
perimeter_mean <= 73.23
gini = 0.472
samples = 21
value = [8, 13]
class = benign
gini = 0.0
samples = 41
value = [0, 41]
class = benign
gini = 0.0
samples = 85
value = [85, 0]
class = malignant
radius_mean <= 13.515
gini = 0.095
samples = 20
value = [19, 1]
class = malignant
gini = 0.0
samples = 1
value = [1, 0]
class = malignant
gini = 0.0
samples = 2
value = [0, 2]
class = benign
gini = 0.0
samples = 2
value = [2, 0]
class = malignant
perimeter_mean <= 97.81
gini = 0.432
samples = 19
value = [6, 13]
class = benign
gini = 0.0
samples = 1
value = [0, 1]
class = benign
gini = 0.0
samples = 19
value = [19, 0]
class = malignant
concave points_mean <= 0.066
gini = 0.36
samples = 17
value = [4, 13]
class = benign
gini = 0.0
samples = 2
value = [2, 0]
class = malignant
concave points_mean <= 0.065
gini = 0.48
samples = 10
value = [4, 6]
class = benign
gini = 0.0
samples = 7
value = [0, 7]
class = benign
concave points_mean <= 0.057
gini = 0.444
samples = 9
value = [3, 6]
class = benign
gini = 0.0
samples = 1
value = [1, 0]
class = malignant
area_mean <= 574.35
gini = 0.5
samples = 6
value = [3, 3]
class = malignant
gini = 0.0
samples = 3
value = [0, 3]
class = benign
gini = 0.0
samples = 2
value = [2, 0]
class = malignant
compactness_mean <= 0.106
gini = 0.375
samples = 4
value = [1, 3]
class = benign
area_mean <= 652.4
gini = 0.5
samples = 2
value = [1, 1]
class = malignant
gini = 0.0
samples = 2
value = [0, 2]
class = benign
gini = 0.0
samples = 1
value = [1, 0]
class = malignant
gini = 0.0
samples = 1
value = [0, 1]
class = benign