Portofolio

FINAL PROJECT
DATA SCIENCE
Batch 16

DigitalSkola
Uncover The World Of Digital Skills With Us

Bootcamp Class

Mushroom Classification

Daftar Pembahasan

- Jenis model (Machine Learning) apa yang memiliki performa terbaik pada set data ini?
- Fitur mana yang digunakan untuk menunjukkan perbandingan dan pengklasifikasian jenis jamur yang bisa dimakan maupun beracun?
- Jamur manakah yang aman dimakan atau jamur beracun yang mematikan?

Tujuan Pembahasan

TUJUAN YANG AKAN KITA KETAHUI PADA SESI INI

- Mengetahui model Machine Learning yang memiliki tingkat akurasi terbaik dalam pengolahan dataset mushroom.
- Mengetahui feature-feature yang penting dan berpengaruh signifikan terhadap klasifikasi jamur edible dan poisonous

Tahapan Analisa

TAHAPAN ANALISA YANG AKAN KITA GUNAKAN PADA SESI INI

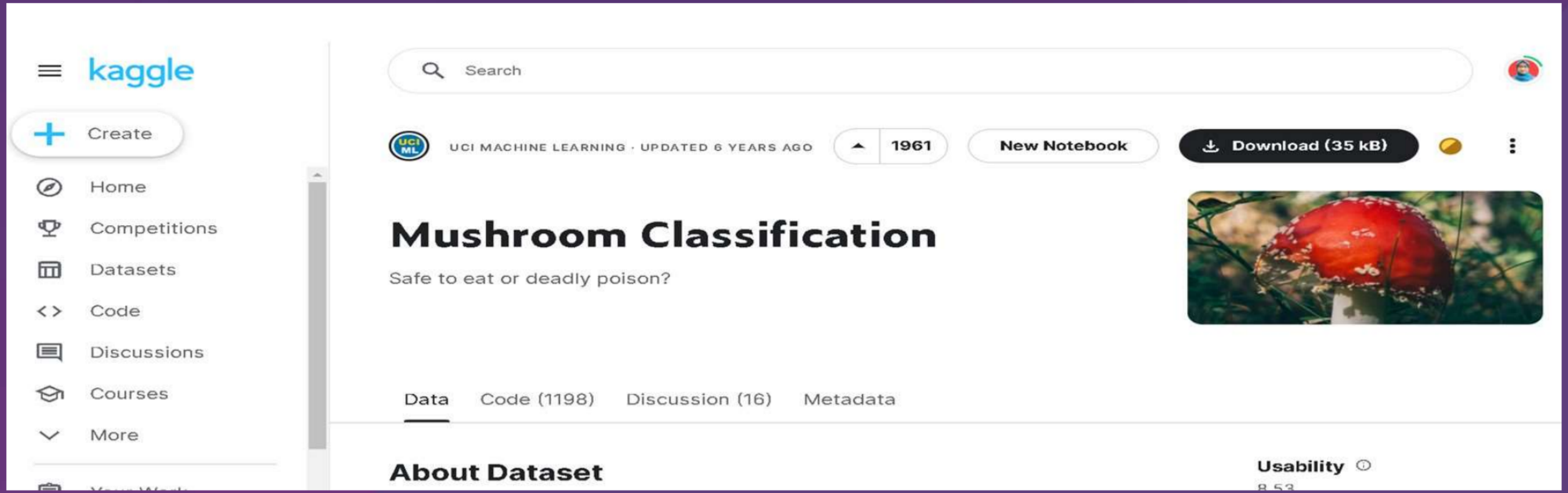
- Exploratory DataAnalysis
- Data Visualization
- Data PreProcessing
- Conclusion (Kesimpulan)

TENTANG DATASET MUSHROOM (JAMUR)

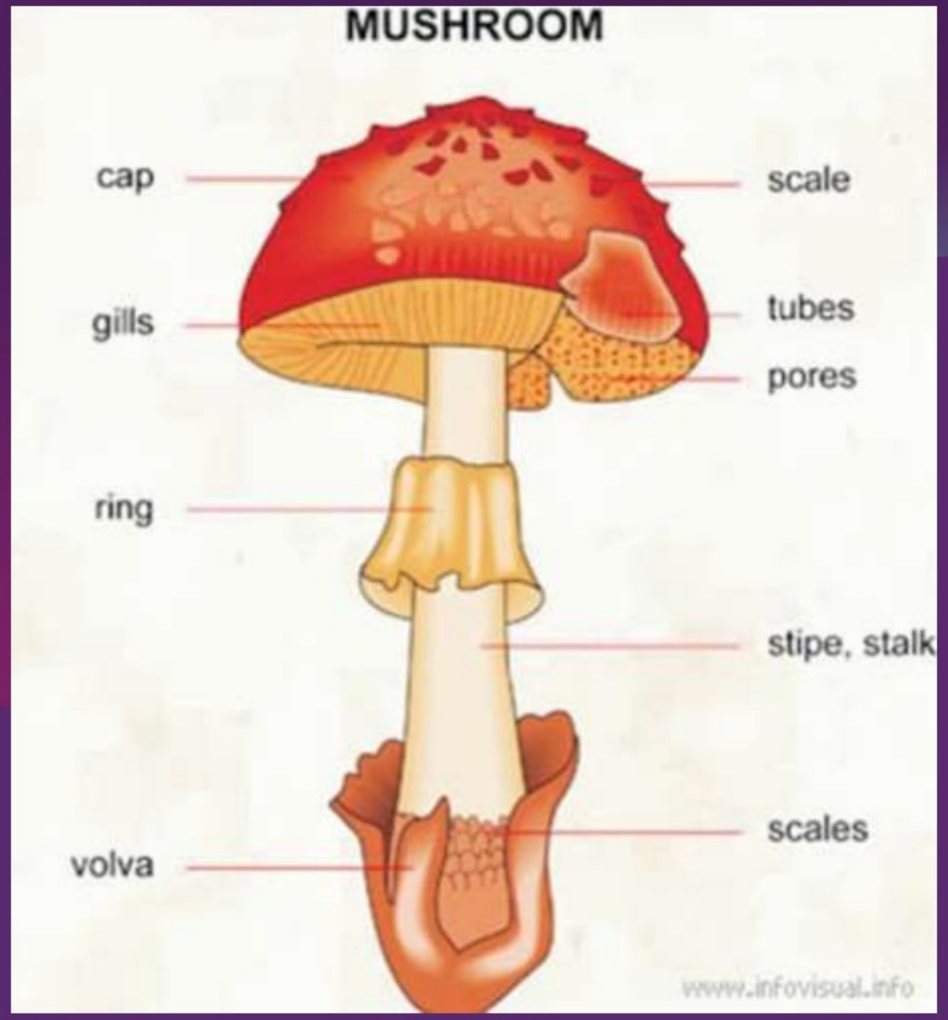
Dataset ini mencakup deskripsi sampel hipotetis yang sesuai dengan 23 spesies jamur insang di Agaricus dan Jamur Keluarga Lepiota yang diambil dari The Audubon Society Field Guide to North American Mushrooms (1981). Setiap spesies diidentifikasi sebagai pasti dapat dimakan, pasti beracun, atau tidak diketahui dapat dimakan dan tidak direkomendasikan untuk dimakan. Objektif bisnis dalam dataset ini ialah sebagai informasi untuk mengantisipasi terjadinya kracunan apabila di konsumsi.

Sumber Data

https://www.kaggle.com/datasets/uciml/mushroom-classification



Informasi Data



1. classes: edible=e, poisonous=p	13. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
2. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s	14. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
3. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s	15. stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
4. cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y	16. stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
5. bruises: bruises=t,no=f	17. veil-type: partial=p,universal=u
6. odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m,none=n, pungent=p, spicy=s	18. veil-color: brown=n,orange=o,white=w,yellow=y
7. gill-attachment: attached=a,descending=d,free=f,notched=n	19. ring-number: none=n,one=o,two=t
8. gill-spacing: close=c,crowded=w,distant=d	20. ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
9. gill-size: broad=b,narrow=n	21. spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
10. gill-color: black=k, brown=n, buff=b , chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y	22. population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
11. stalk-shape: enlarging=e,tapering=t	23. habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d
12. stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?	

Exploratory Data Analysis

Library Yang Digunakan

Library

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier , RandomForestRegressor
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV

import warnings
warnings.filterwarnings('ignore')
```

Deskripsi Data

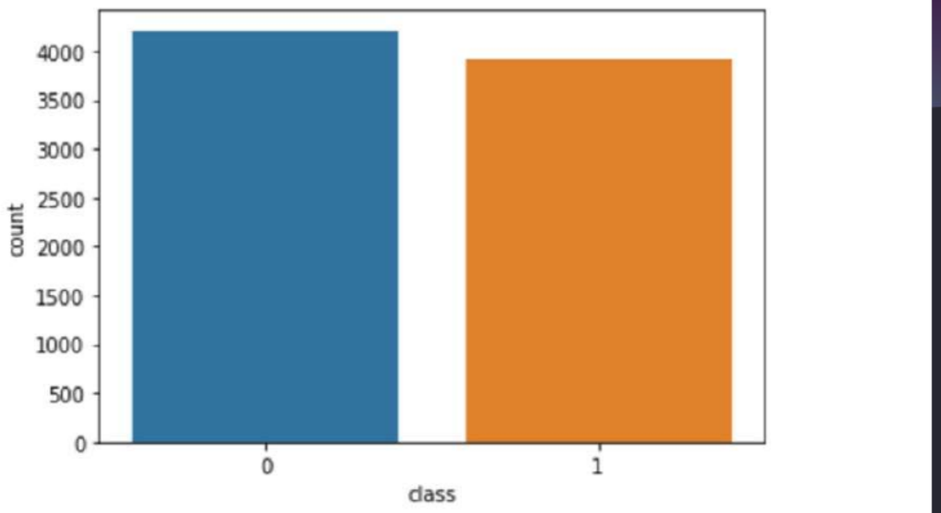
- Terdapat 23 columns
- Semua data dengan type Object
- Termasuk dalam (Categorical data)
- Tidak ada Nulldata
- Tidak ada dataduplicated
- Terdapat data Missing (?) di kolom ‘stalk-root’ (Data Missing (?) diubah dengan data modus) pada kolom tersebut.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   class                     8124 non-null   object
 1   cap-shape                 8124 non-null   object
 2   cap-surface               8124 non-null   object
 3   cap-color                 8124 non-null   object
 4   bruises                   8124 non-null   object
 5   odor                      8124 non-null   object
 6   gill-attachment           8124 non-null   object
 7   gill-spacing              8124 non-null   object
 8   gill-size                 8124 non-null   object
 9   gill-color                8124 non-null   object
 10  stalk-shape               8124 non-null   object
 11  stalk-root                8124 non-null   object
 12  stalk-surface-above-ring  8124 non-null   object
 13  stalk-surface-below-ring  8124 non-null   object
 14  stalk-color-above-ring    8124 non-null   object
 15  stalk-color-below-ring    8124 non-null   object
 16  veil-type                 8124 non-null   object
 17  veil-color                8124 non-null   object
 18  ring-number               8124 non-null   object
 19  ring-type                 8124 non-null   object
 20  spore-print-color         8124 non-null   object
 21  population                8124 non-null   object
 22  habitat                   8124 non-null   object
dtypes: object(23)
```

Data Perbandingan Jamur yang beracun (P) dengan yang bisa dimakan (e)

```
sns.countplot(Project['class'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe3e11681d0>
```

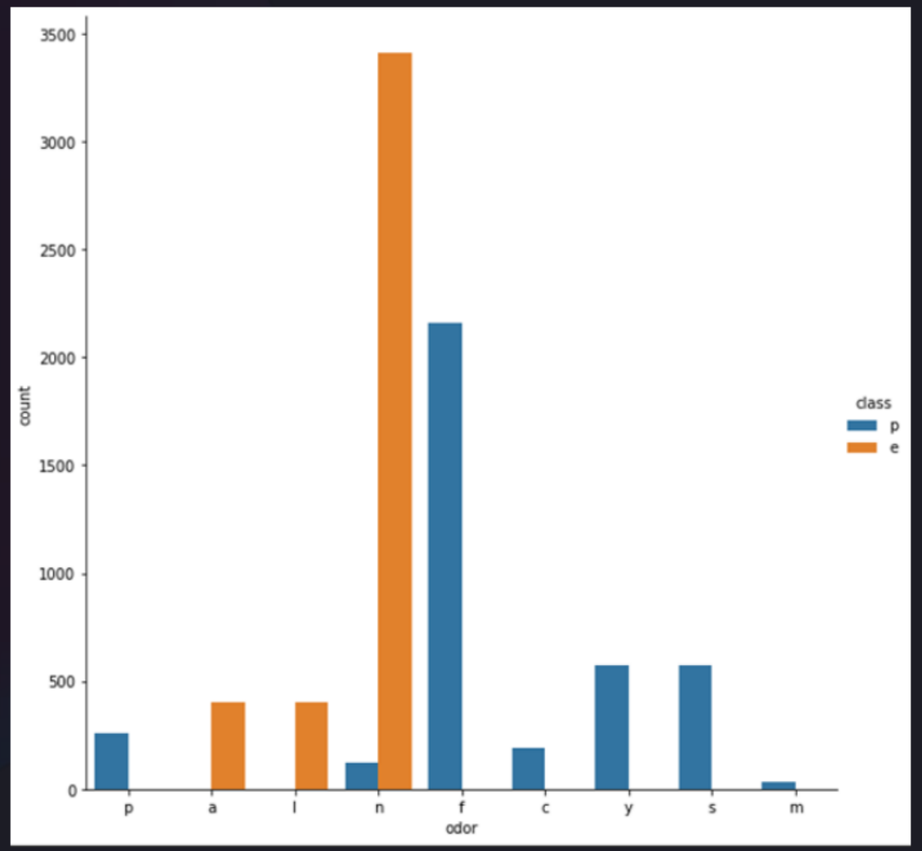


Jamur di bedakan menjadi 2 klasifikasi yaitu jamur yang bisa dimakan (edible (e)) dan jamur yang beracun (poisonous (p)).

Jika ditinjau dari perbandingan grafik dari data jumlah jamur yang tersebar di amerika utara selama tahun 1981, perbandingan nya tidak terlalu signifikan atau dalam arti lain untuk membedakan jamur tersebut bisa dimakan atau beracun membutuhkan beberapa data tambahan yang akan menjadi acuan pemilihan jamur tersebut kategori e atau kategori p.

Dari alasan yang telah disampaikan, kami memproses data ciri-ciri jamur untuk mengetahui feature-feature jamur yang paling berpengaruh untuk menentukan jamur yang bisa dimakan dan jamur yang beracun.

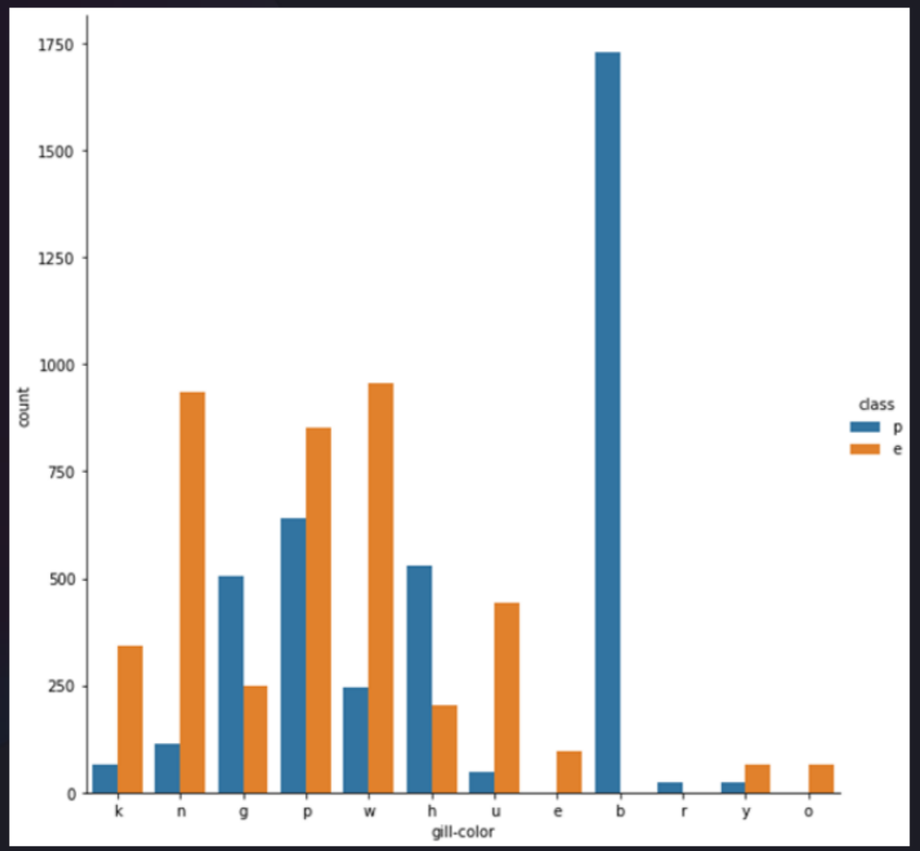
ODOR (BAU)



Ket:
almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s

Kesimpulan:

- Jamur yang memiliki bau Foul (f), Creosote (c), Fishy (y), Spicy (s), Musty (m), dan Pungent (p) semuanya merupakan jamur beracun
- Semua jamur dengan bau almond (a) dan anise (l) merupakan jamur yang dapat dimakan
- Sementara jamur yang tidak berbau (none=n) ada yang beracun dan ada yang bisa dimakan

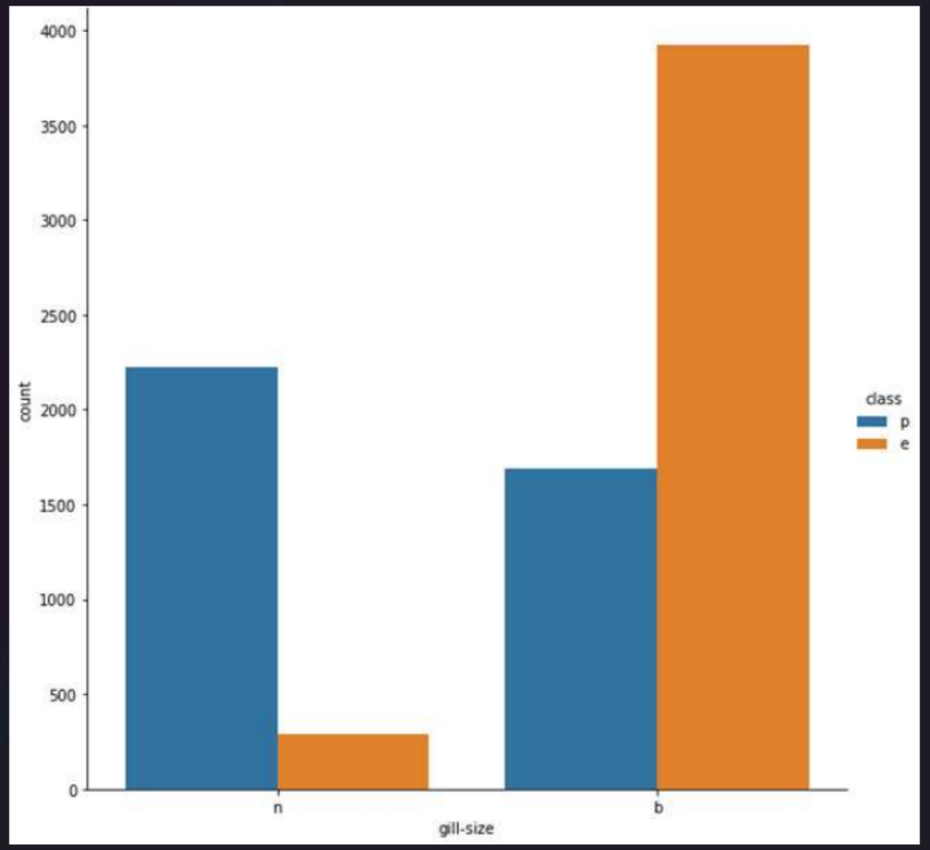


Gill-Color

Ket: gill-color:
black=k,brown=n,buff=b,chocolate=h,gray=g,
green=r,orange=o,pink=p,purple=u,red=e,white=w,yellow=y

Kesimpulan:

•Semua jamur yang memiliki warna gill B (Buff) adalah jamur beracun sementara jamur yang memiliki warna gill E (Red) dan O (Orange) adalah jamur yang dapat dimakan



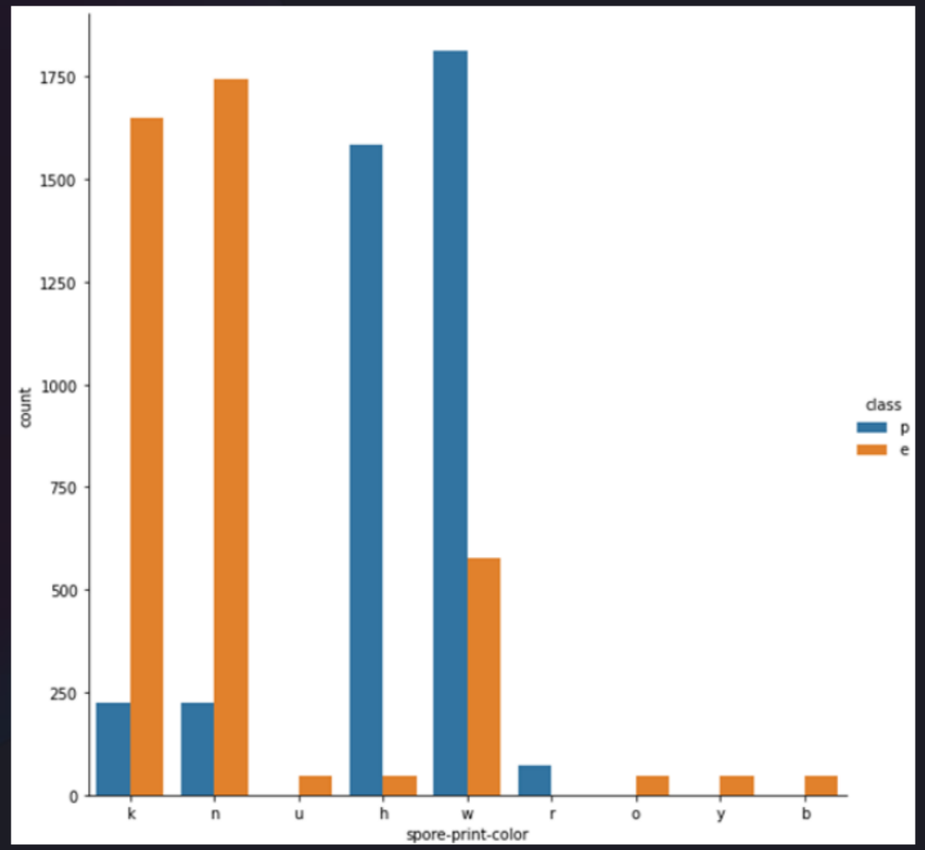
Gill-size

Kode broad =b, narrow =n

Kesimpulan:

•Sebagian besar jamur yang memiliki gill size B (Broad) merupakan jamur edible

Spore Print Color



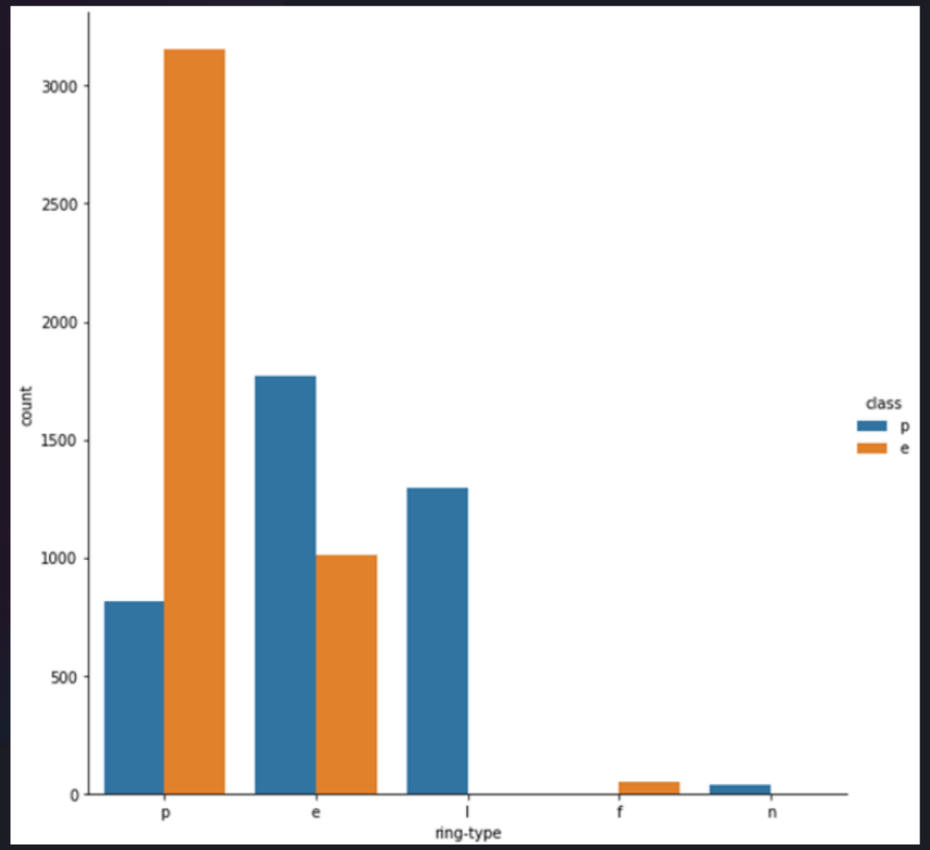
Ket:
black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y

Kesimpulan:

•Semua jamur yang memiliki warna spora U (Purple), O (Orange), Y (Yellow), dan B (Buff), dan Sebagian besar jamur berwarna K (Black) dan N (Brown) adalah jamur yang edible

•Sementarasebagianbesar jamur yang memilikiwarna spora H (Chocolate) dan W (White) adalahjamur beracun

Ring Type



Ket:
cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z

Kesimpulan:

- Sebagian besar jamur dengan ring type P (pendant) dan semua jamur dengan ring type F (flaring) adalah jamur yang edible
- Semuajamur dengan ring type L (Large) dan N (None) adalahjamur beracun

Manipulasi data yang mising pada kolom ‘stalk-root’

```
Project['stalk-root']
```

```
0         e
1         c
2         c
3         e
4         e
        ..
8119      ?
8120      ?
8121      ?
8122      ?
8123      ?
Name: stalk-root, Length: 8124, dtype: object
```

```
Project['stalk-root'].value_counts(normalize=True)
```

```
b    0.464796
?    0.305268
e    0.137863
c    0.068439
r    0.023634
Name: stalk-root, dtype: float64
```

```
#Mengganti data missing(?) dengan data modus
Project['stalk-root'].replace(['?'], 'b')
```

```
0         e
1         c
2         c
3         e
4         e
        ..
8119      b
8120      b
8121      b
8122      b
8123      b
Name: stalk-root, Length: 8124, dtype: object
```

Data PreProcessing

Label Encoder

M

```
f  Pro ec..app y La e Enc er().fit_transform)
f he d(10)
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number
0	1	5	2	4	1	6	1	0	1	4	...	2	7	7	0	2	1
1	0	5	2	9	1	0	1	0	0	4	...	2	7	7	0	2	1
2	0	0	2	8	1	3	1	0	0	5	...	2	7	7	0	2	1
3	1	5	3	8	1	6	1	0	1	5	...	2	7	7	0	2	1
4	0	5	2	3	0	5	1	1	0	4	...	2	7	7	0	2	1
5	0	5	3	9	1	0	1	0	0	5	...	2	7	7	0	2	1
6	0	0	2	8	1	0	1	0	0	2	...	2	7	7	0	2	1
7	0	0	3	8	1	3	1	0	0	5	...	2	7	7	0	2	1
8	1	5	3	8	1	6	1	0	1	7	...	2	7	7	0	2	1
9	0	0	2	9	1	0	1	0	0	2	...	2	7	7	0	2	1

Mappings Label Encoder

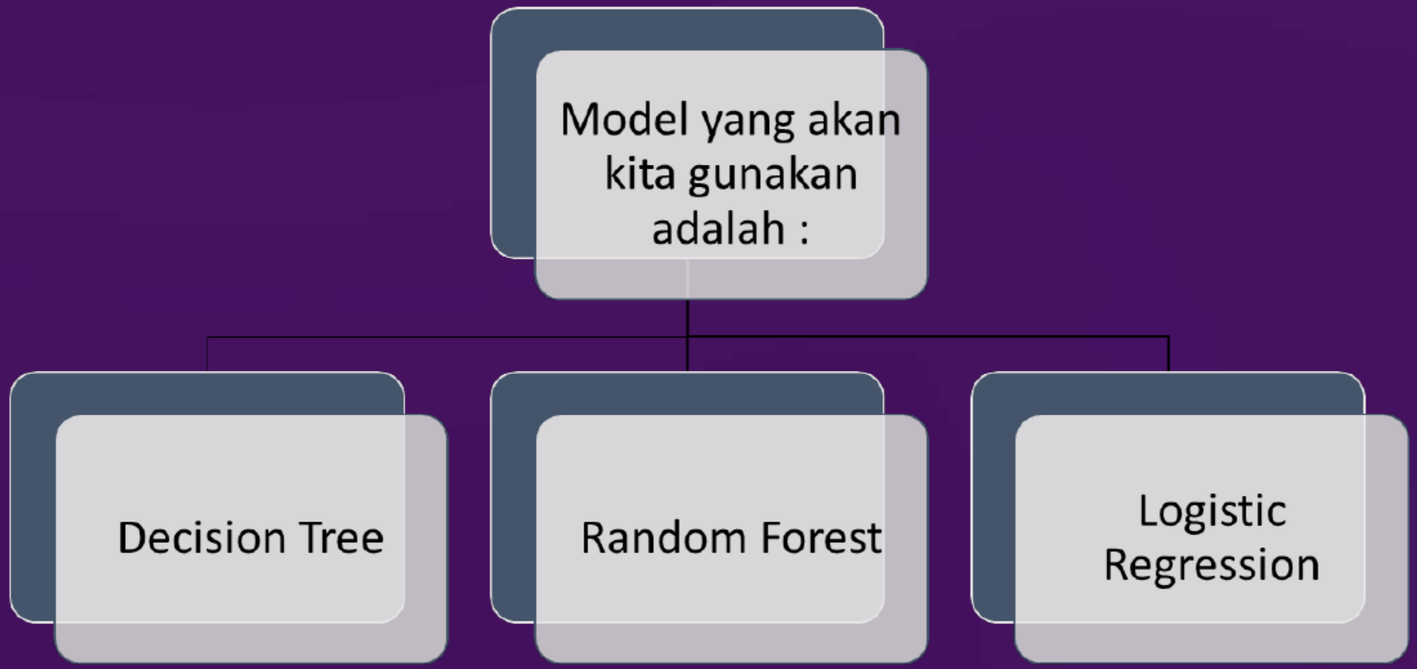
```
mappings = list()

PL = LabelEncoder()
for columns in range (len(Project.columns)):
  Project[Project.columns[columns]] = PL.fit_transform(Project[Project.columns[columns]])
  mappings_dict = {index :label for index, label in enumerate(PL.classes_)}
  mappings.append (mappings_dict)
```

```
mappings

[{0: 'e', 1: 'p'},
 {0: 'b', 1: 'c', 2: 'f', 3: 'k', 4: 's', 5: 'x'},
 {0: 'f', 1: 'g', 2: 's', 3: 'y'},
 {0: 'b',
  1: 'c',
  2: 'e',
  3: 'g',
  4: 'n',
  5: 'p',
  6: 'r',
  7: 'u',
  8: 'w',
  9: 'y'},
 {0: 'f', 1: 't'},
 {0: 'a', 1: 'c', 2: 'f', 3: 'l', 4: 'm', 5: 'n', 6: 'p', 7: 's', 8: 'y'},
 {0: 'a', 1: 'f'},
 {0: 'c', 1: 'w'},
 {0: 'b', 1: 'n'},
 {0: 'b',
  1: 'e',
  2: 'g',
  3: 'h',
  4: 'k',
  5: 'n',
  6: 'o',
  7: 'p',
  8: 'r',
  9: 'u',
  10: 'w',
  11: 'y'},
 {0: 'e', 1: 't'},
 {0: '?', 1: 'b', 2: 'c', 3: 'e', 4: 'r'},
 {0: 'f', 1: 'k', 2: 's', 3: 'y'},
 {0: 'f', 1: 'k', 2: 's', 3: 'y'},
 {0: 'b', 1: 'c', 2: 'e', 3: 'g', 4: 'n', 5: 'o', 6: 'p', 7: 'w', 8: 'y'},
 {0: 'b', 1: 'c', 2: 'e', 3: 'g', 4: 'n', 5: 'o', 6: 'p', 7: 'w', 8: 'y'},
 {0: 'p'},
 {0: 'n', 1: 'o', 2: 'w', 3: 'y'},
 {0: 'n', 1: 'o', 2: 't'},
 {0: 'e', 1: 'f', 2: 'l', 3: 'n', 4: 'p'},
 {0: 'b', 1: 'h', 2: 'k', 3: 'n', 4: 'o', 5: 'r', 6: 'u', 7: 'w', 8: 'y'},
 {0: 'a', 1: 'c', 2: 'n', 3: 's', 4: 'v', 5: 'y'},
 {0: 'd', 1: 'g', 2: 'l', 3: 'm', 4: 'p', 5: 'u', 6: 'w'}]
```

Target Feature

```
x = df.drop(['class'], axis=1)
y = df['class']

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.3 , random_state=0)
```

Model Machine Learning



Modelling

```
dt = DecisionTreeClassifier ( max_depth=3, random_state=0)
dt.fit (x_train, y_train)

rm = RandomForestClassifier (max_depth=3, random_state=0)
rm.fit (x_train, y_train)

lr = LogisticRegression()
lr.fit (x_train, y_train)
```

Hasil Uji Model Machine Learning

Decision Tree

	precision	recall	f1-score	support
0	0.98	0.95	0.96	1272
1	0.95	0.97	0.96	1166
accuracy			0.96	2438
macro avg	0.96	0.96	0.96	2438
weighted avg	0.96	0.96	0.96	2438

	precision	recall	f1-score	support

Random Forest

	precision	recall	f1-score	support
0	0.96	1.00	0.98	1272
1	1.00	0.96	0.98	1166
accuracy			0.98	2438
macro avg	0.98	0.98	0.98	2438
weighted avg	0.98	0.98	0.98	2438

Logistic Regression

	precision	recall	f1-score	support
0	0.94	0.96	0.95	1272
1	0.96	0.94	0.95	1166
accuracy			0.95	2438
macro avg	0.95	0.95	0.95	2438
weighted avg	0.95	0.95	0.95	2438

Perbandingan Akurasi Model

```
pd.DataFrame({'Models' : ['DT', 'RM', 'LR'],
              'ACC': [accuracy_score(y_test, y_pred1),
                      accuracy_score(y_test, y_pred2),
                      accuracy_score(y_test, y_pred3)]})
```

	Models	ACC
0	DT	0.961444
1	RM	0.977851
2	LR	0.949139

Hyper Parameter Tuning

```
rf_Grid.best_params_
```

```
{'bootstrap': True,
 'max_depth': 5,
```

```
ACC rm 0.9913863822805579
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1272
           1       1.00      0.98      0.99      1166

    accuracy                           0.99      2438
   macro avg       0.99      0.99      0.99      2438
weighted avg       0.99      0.99      0.99      2438
```

99%

Hasil tingkat akurasi dari Model Random Forest setelah dilakukan Hyper Parameter Tuning adalah 99%

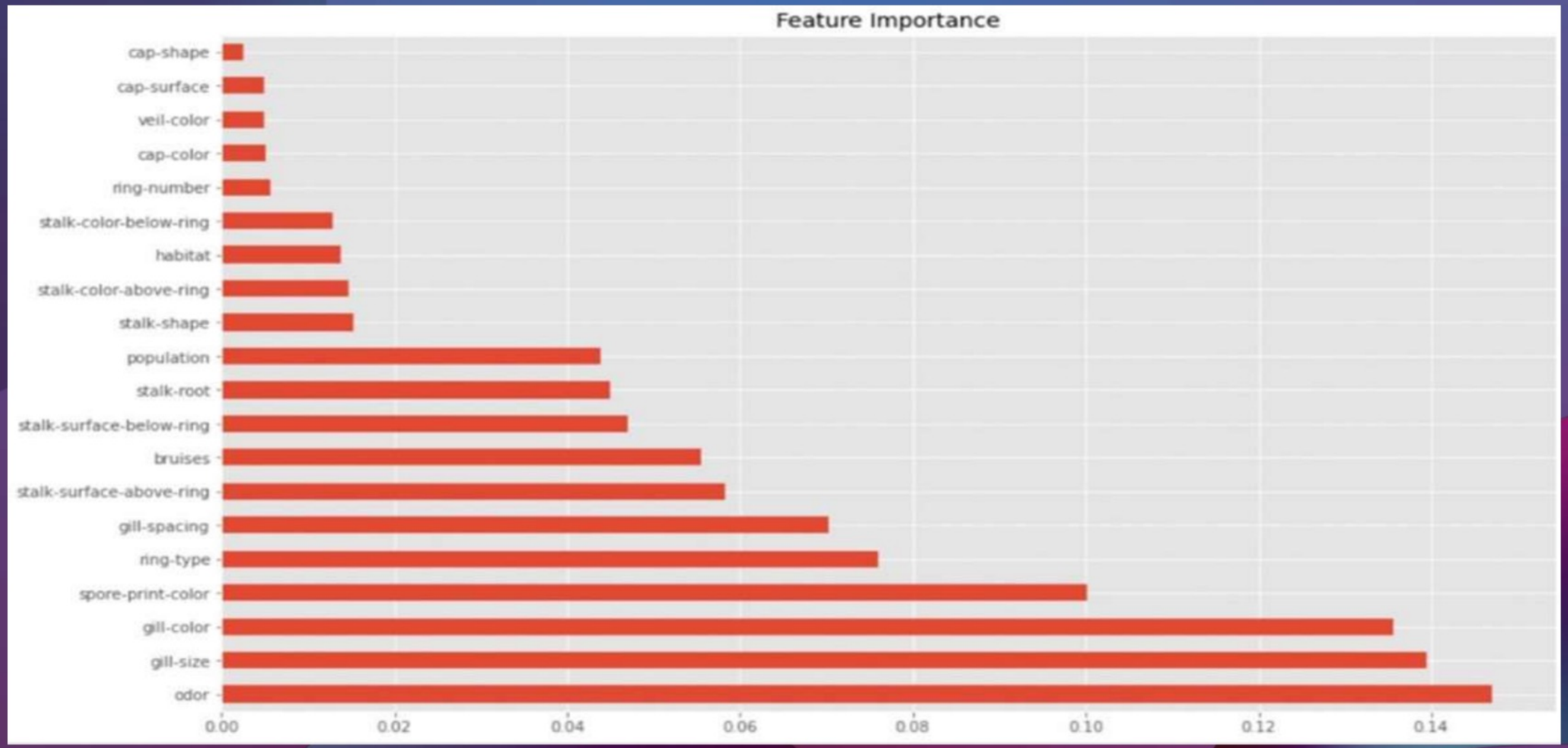
Menentukan Feature Terbaik (Feature Importance Score)

```
1 feature_scores = pd.Series(rf .feature_importances_, index=X_train.columns).sort_values(ascending=False)
2
3 feature_scores
```

```
odor                        0.147087
gill-size                   0.139379
gill-color                  0.135588
spore-print-color           0.100080
ring-type                   0.075994
gill-spacing                0.070265
stalk-surface-above-ring    0.058209
bruises                     0.055499
stalk-surface-below-ring    0.047065
stalk-root                  0.045055
population                  0.043893
stalk-shape                 0.015192
stalk-color-above-ring      0.014812
habitat                     0.013825
stalk-color-below-ring      0.012897
ring-number                 0.005749
cap-color                   0.005063
veil-color                  0.004919
cap-surface                 0.004856
cap-shape                   0.002567
gill-attachment             0.002008
veil-type                   0.000000
dtype: float64
```

Berdasarkan hasil pengujian Features Importance untuk Model Random Forest, 5 features dengan indeks nilai penting tertinggi adalah Odor, , Gill Size, Gill Color, Spore Print Color, dan Ring Type.

Visualisasi Training Semua Feature



Kesimpulan

Model Machine Learning yang memiliki akurasi terbaik untuk dataset ini adalah Random Forest dengan tingkat akurasi 99%

Feature yang penting dan memiliki pengaruh signifikan terhadap pengklasifikasian jamur antara yang edible dan poisonous adalah Odor, Gill Color, Gill Size, Spore Print Color, dan Ring Type.

Terima kasih!