```
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder

from sklearn.impute import SimpleImputer

from sklearn.compose import make_column_transformer, ColumnTransformer
from sklearn.pipeline import Pipeline, make_pipeline

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClass


from sklearn.model_selection import cross_val_score, StratifiedKFold, train_test_split, GridSe

# Setting untuk membuat angka mudah dibaca di display
pd.options.display.float_format = '{:20.2f}'.format

# Menampilkan semua kolom pada output
pd.set_option('display.max_columns', None)
```

```
train_df = pd.read_csv('../data/train.csv')
test_df = pd.read_csv('../data/test.csv')
```

```
train_df.head(100) # Saya murni penasaran, umumnya cukup 5 aja.
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0	A/5 21171	7.25	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	1	0	PC 17599	71.28	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0	STON/O2. 3101282	7.92	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	1	0	113803	53.10	C123
4	5	0	3	Allen, Mr. William Henry	male	35.00	0	0	373450	8.05	NaN
...	...	...	...	...	...	...	...	...	...	...	...
95	96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8.05	NaN
96	97	0	1	Goldschmidt, Mr. George B	male	71.00	0	0	PC 17754	34.65	A5
97	98	1	1	Greenfield, Mr. William Bertram	male	23.00	0	1	PC 17759	63.36	D10 D12
98	99	1	2	Doling, Mrs. John T (Ada Julia Bone)	female	34.00	0	1	231919	23.00	NaN
99	100	0	2	Kantor, Mr. Sinai	male	34.00	1	0	244367	26.00	NaN

100 rows × 12 columns

```
test_df.head(100)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.50	0	0	330911	7.83	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.00	1	0	363272	7.00	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.00	0	0	240276	9.69	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.00	0	0	315154	8.66	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.00	1	1	3101298	12.29	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
95	987	3	Tenglin, Mr. Gunnar Isidor	male	25.00	0	0	350033	7.80	NaN	S
96	988	1	Cavendish, Mrs. Tyrell William (Julia Florence...	female	76.00	1	0	19877	78.85	C46	S
97	989	3	Makinen, Mr. Kalle Edvard	male	29.00	0	0	STON/O 2. 3101268	7.92	NaN	S
98	990	3	Braf, Miss. Elin Ester Maria	female	20.00	0	0	347471	7.85	NaN	S
99	991	3	Nancarrow, Mr. William Henry	male	33.00	0	0	A./5. 3338	8.05	NaN	S

100 rows × 11 columns

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
train_df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.00	891.00	891.00	714.00	891.00	891.00	891.00
mean	446.00	0.38	2.31	29.70	0.52	0.38	32.20
std	257.35	0.49	0.84	14.53	1.10	0.81	49.69
min	1.00	0.00	1.00	0.42	0.00	0.00	0.00
25%	223.50	0.00	2.00	20.12	0.00	0.00	7.91
50%	446.00	0.00	3.00	28.00	0.00	0.00	14.45
75%	668.50	1.00	3.00	38.00	1.00	0.00	31.00
max	891.00	1.00	3.00	80.00	8.00	6.00	512.33

```
train_df.describe(include=['O'])
```

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Braund, Mr. Owen Harris	male	347082	G6	S
freq	1	577	7	4	644

```
train_df.groupby(["Pclass"], as_index=False)["Survived"].mean()
```

	Pclass	Survived
0	1	0.63
1	2	0.47
2	3	0.24

```
train_df.groupby(["Sex"], as_index=False)["Survived"].mean()
```

	Sex	Survived
0	female	0.74
1	male	0.19

```
train_df.groupby(["SibSp"], as_index=False)["Survived"].mean()
```

	SibSp	Survived
0	0	0.35
1	1	0.54
2	2	0.46
3	3	0.25
4	4	0.17
5	5	0.00
6	8	0.00

```
train_df.groupby(["Parch"], as_index=False)["Survived"].mean()
```

	Parch	Survived
0	0	0.34
1	1	0.55
2	2	0.50
3	3	0.60
4	4	0.00
5	5	0.20
6	6	0.00

```
train_df['Family_Size'] = train_df['SibSp'] + train_df['Parch'] + 1
test_df['Family_Size'] = test_df['SibSp'] + test_df['Parch'] + 1
```

```
train_df.head(100)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0	A/5 21171	7.25	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	1	0	PC 17599	71.28	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0	STON/O2. 3101282	7.92	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	1	0	113803	53.10	C123
4	5	0	3	Allen, Mr. William Henry	male	35.00	0	0	373450	8.05	NaN
...	...	...	...	...	...	...	...	...	...	...	...
95	96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8.05	NaN
96	97	0	1	Goldschmidt, Mr. George B	male	71.00	0	0	PC 17754	34.65	A5
97	98	1	1	Greenfield, Mr. William Bertram	male	23.00	0	1	PC 17759	63.36	D10 D12
98	99	1	2	Doling, Mrs. John T (Ada Julia Bone)	female	34.00	0	1	231919	23.00	NaN
99	100	0	2	Kantor, Mr. Sinai	male	34.00	1	0	244367	26.00	NaN

100 rows × 13 columns

```
train_df.groupby(["Family_Size"], as_index=False)["Survived"].mean()
```

	Family_Size	Survived
0	1	0.30
1	2	0.55
2	3	0.58
3	4	0.72
4	5	0.20
5	6	0.14
6	7	0.33
7	8	0.00
8	11	0.00

```
family_map = {1: 'Alone', 2: 'Small', 3: 'Small', 4: 'Small', 5: 'Medium', 6: 'Medium', 7: 'La
train_df['Family_Size_Grouped'] = train_df['Family_Size'].map(family_map)
test_df['Family_Size_Grouped'] = test_df['Family_Size'].map(family_map) #Kalau di video tutori
```

```
train_df.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Eml
0	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0	A/5 21171	7.25	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	1	0	PC 17599	71.28	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0	STON/O2. 3101282	7.92	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	1	0	113803	53.10	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.00	0	0	373450	8.05	NaN	

```
train_df.groupby(["Family_Size_Grouped"], as_index=False)["Survived"].mean()
```

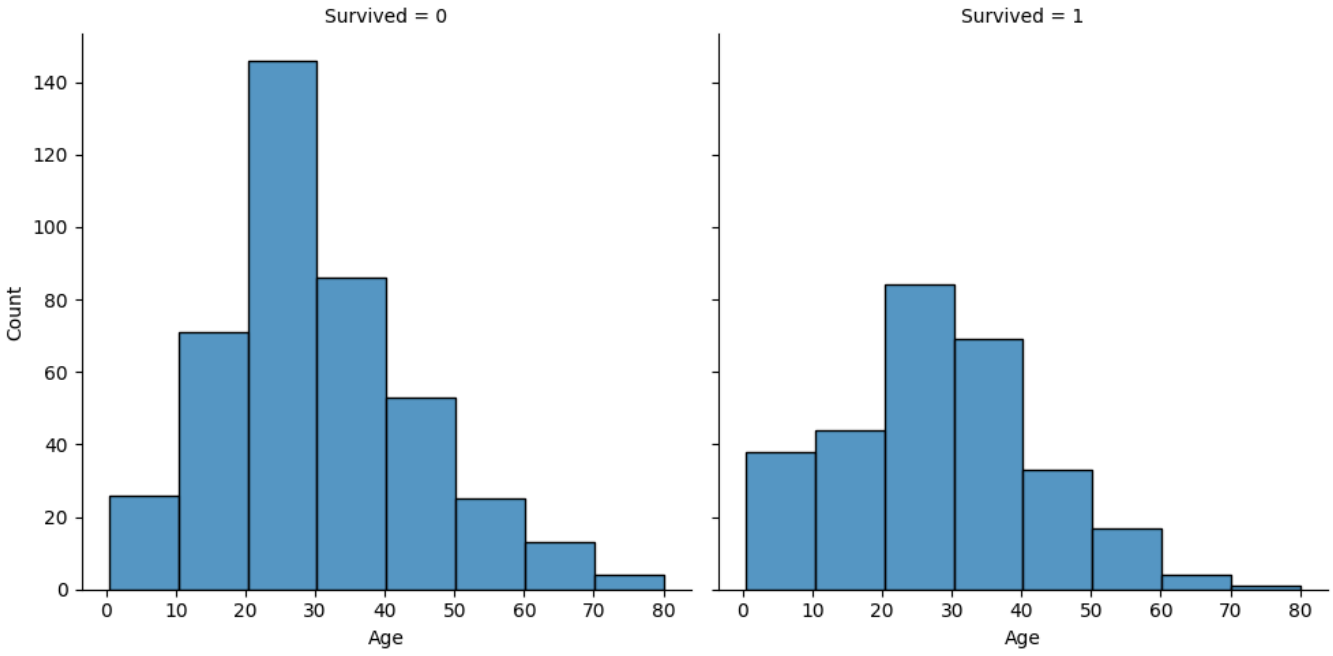
	Family_Size_Grouped	Survived
0	Alone	0.30
1	Large	0.16
2	Medium	0.16
3	Small	0.58

```
train_df.groupby(["Embarked"], as_index=False)["Survived"].mean()
```

	Embarked	Survived
0	C	0.55
1	Q	0.39
2	S	0.34

```
sns.displot(train_df, x='Age', col='Survived', binwidth=10, height=5) # Melihat lebih banyak t
```

```
<seaborn.axisgrid.FacetGrid at 0x1fdc04e6f60>
```




```
train_df['Age_Cut'] = pd.qcut(train_df['Age'], 8)
test_df['Age_Cut'] = pd.qcut(test_df['Age'], 8)
```

```
train_df.groupby(["Age_Cut"], as_index=False, observed = False)["Survived"].mean()
```

	Age_Cut	Survived
0	(0.419, 16.0]	0.55
1	(16.0, 20.125]	0.34
2	(20.125, 24.0]	0.37
3	(24.0, 28.0]	0.35
4	(28.0, 32.312]	0.42
5	(32.312, 38.0]	0.45
6	(38.0, 47.0]	0.33
7	(47.0, 80.0]	0.42

```
train_df.loc[train_df['Age'] <= 16, 'Age'] = 0
train_df.loc[(train_df['Age'] > 16) & (train_df['Age'] <= 20.125), 'Age'] = 1
train_df.loc[(train_df['Age'] > 20.125) & (train_df['Age'] <= 24), 'Age'] = 2
train_df.loc[(train_df['Age'] > 24) & (train_df['Age'] <= 28), 'Age'] = 3
train_df.loc[(train_df['Age'] > 28) & (train_df['Age'] <= 32.312), 'Age'] = 4
train_df.loc[(train_df['Age'] > 32.312) & (train_df['Age'] <= 38), 'Age'] = 5
train_df.loc[(train_df['Age'] > 38) & (train_df['Age'] <= 47), 'Age'] = 6
train_df.loc[(train_df['Age'] > 47) & (train_df['Age'] <= 80), 'Age'] = 7
train_df.loc[train_df['Age'] > 80, 'Age'] = 8

test_df.loc[test_df['Age'] <= 16, 'Age'] = 0
test_df.loc[(test_df['Age'] > 16) & (test_df['Age'] <= 20.125), 'Age'] = 1
test_df.loc[(test_df['Age'] > 20.125) & (test_df['Age'] <= 24), 'Age'] = 2
test_df.loc[(test_df['Age'] > 24) & (test_df['Age'] <= 28), 'Age'] = 3
test_df.loc[(test_df['Age'] > 28) & (test_df['Age'] <= 32.312), 'Age'] = 4
test_df.loc[(test_df['Age'] > 32.312) & (test_df['Age'] <= 38), 'Age'] = 5
test_df.loc[(test_df['Age'] > 38) & (test_df['Age'] <= 47), 'Age'] = 6
test_df.loc[(test_df['Age'] > 47) & (test_df['Age'] <= 80), 'Age'] = 7
test_df.loc[test_df['Age'] > 80, 'Age'] = 8
```

```
train_df.head()
```

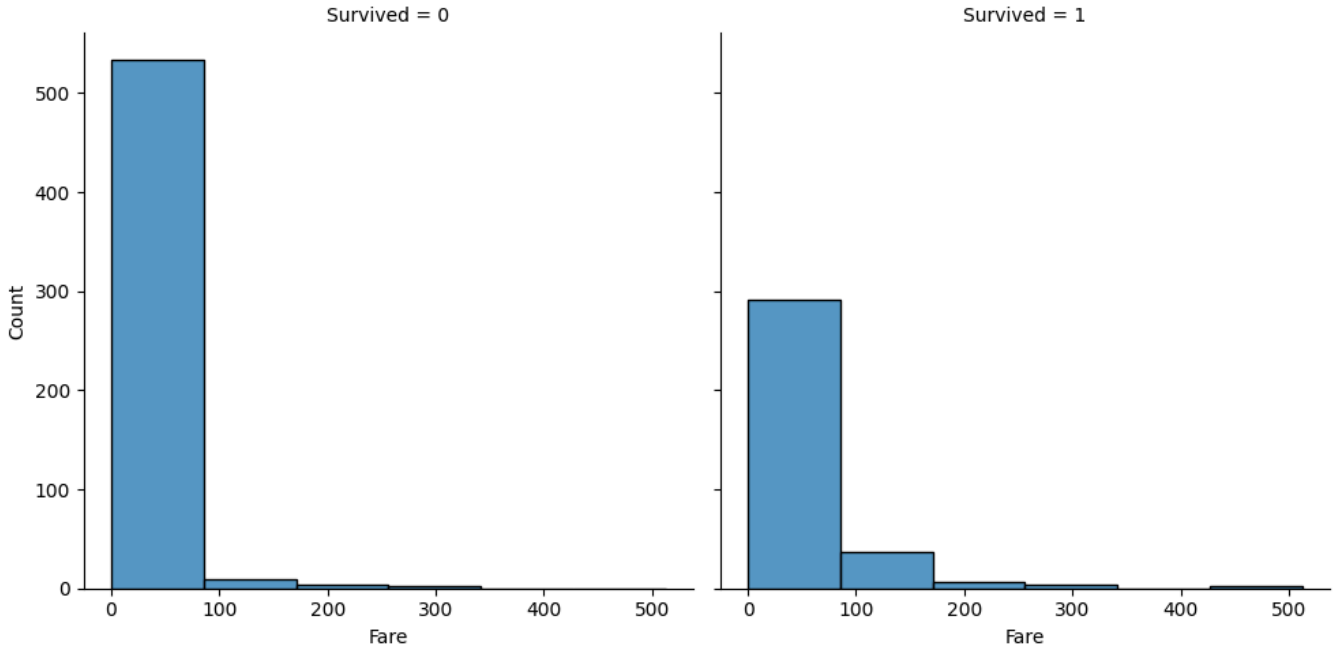
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	1	0	3	Braund, Mr. Owen Harris	male	2.00	1	0	A/5 21171	7.25	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	5.00	1	0	PC 17599	71.28	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	3.00	0	0	STON/O2. 3101282	7.92	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	5.00	1	0	113803	53.10	C123	
4	5	0	3	Allen, Mr. William Henry	male	5.00	0	0	373450	8.05	NaN	

```
train_df.groupby(["Age"], as_index=False)["Survived"].mean()
```

	Age	Survived
0	0.00	0.55
1	1.00	0.34
2	2.00	0.37
3	3.00	0.35
4	4.00	0.42
5	5.00	0.45
6	6.00	0.33
7	7.00	0.42

```
sns.displot(train_df, x='Fare', col='Survived', binwidth=80, height=5)
```

```
<seaborn.axisgrid.FacetGrid at 0x1fdc0cb6c60>
```




```
# Cara Modern (Tanpa inplace=True)
test_df['Fare'] = test_df['Fare'].fillna(test_df['Fare'].mean())
```

```
train_df['Fare_Cut'] = pd.qcut(train_df['Fare'], 6)
test_df['Fare_Cut'] = pd.qcut(test_df['Fare'], 6)
```

```
train_df.groupby(["Fare_Cut"], as_index=False, observed = False)["Survived"].mean()
```

	Fare_Cut	Survived
0	(-0.001, 7.775]	0.21
1	(7.775, 8.662]	0.19
2	(8.662, 14.454]	0.37
3	(14.454, 26.0]	0.44
4	(26.0, 52.369]	0.42
5	(52.369, 512.329]	0.70

```
train_df.loc[train_df['Fare'] <= 7.775, 'Fare'] = 0
train_df.loc[(train_df['Fare'] > 7.775) & (train_df['Fare'] <= 8.662), 'Fare'] = 1
train_df.loc[(train_df['Fare'] > 8.662) & (train_df['Fare'] <= 14.454), 'Fare'] = 2
train_df.loc[(train_df['Fare'] > 14.454) & (train_df['Fare'] <= 26.0), 'Fare'] = 3
train_df.loc[(train_df['Fare'] > 26.0) & (train_df['Fare'] <= 52.369), 'Fare'] = 4
train_df.loc[(train_df['Fare'] > 52.369) & (train_df['Fare'] <= 512.329), 'Fare'] = 5
train_df.loc[train_df['Fare'] > 512.329, 'Fare']

test_df.loc[test_df['Fare'] <= 7.775, 'Fare'] = 0
test_df.loc[(test_df['Fare'] > 7.775) & (test_df['Fare'] <= 8.662), 'Fare'] = 1
test_df.loc[(test_df['Fare'] > 8.662) & (test_df['Fare'] <= 14.454), 'Fare'] = 2
test_df.loc[(test_df['Fare'] > 14.454) & (test_df['Fare'] <= 26.0), 'Fare'] = 3
test_df.loc[(test_df['Fare'] > 26.0) & (test_df['Fare'] <= 52.369), 'Fare'] = 4
```

```
test_df.loc[(test_df['Fare'] > 52.369) & (test_df['Fare'] <= 512.329), 'Fare'] = 5
test_df.loc[test_df['Fare'] > 512.329, 'Fare']
```

```
343    512.33
Name: Fare, dtype: float64
```

```
train_df.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	2.00	1	0	A/5 21171	0.00	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	5.00	1	0	PC 17599	5.00	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	3.00	0	0	STON/O2. 3101282	1.00	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	5.00	1	0	113803	5.00	C123	
4	5	0	3	Allen, Mr. William Henry	male	5.00	0	0	373450	1.00	NaN	

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   PassengerId          891 non-null    int64
 1   Survived             891 non-null    int64
 2   Pclass               891 non-null    int64
 3   Name                 891 non-null    object
 4   Sex                  891 non-null    object
 5   Age                  714 non-null    float64
 6   SibSp                891 non-null    int64
 7   Parch                891 non-null    int64
 8   Ticket               891 non-null    object
 9   Fare                 891 non-null    float64
 10  Cabin                204 non-null    object
 11  Embarked             889 non-null    object
 12  Family_Size          891 non-null    int64
 13  Family_Size_Grouped  891 non-null    object
 14  Age_Cut              714 non-null    category
 15  Fare_Cut             891 non-null    category
dtypes: category(2), float64(2), int64(6), object(6)
memory usage: 100.0+ KB
```

```
test_df.head(5)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Family
0	892	3	Kelly, Mr. James	male	5.00	0	0	330911	1.00	NaN	Q	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	6.00	1	0	363272	0.00	NaN	S	
2	894	2	Myles, Mr. Thomas Francis	male	7.00	0	0	240276	2.00	NaN	Q	
3	895	3	Wirz, Mr. Albert	male	3.00	0	0	315154	2.00	NaN	S	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	2.00	1	1	3101298	2.00	NaN	S	

```
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype   
---  ------               --------------  -----   
 0   PassengerId          418 non-null    int64   
 1   Pclass               418 non-null    int64   
 2   Name                 418 non-null    object  
 3   Sex                  418 non-null    object  
 4   Age                  332 non-null    float64 
 5   SibSp                418 non-null    int64   
 6   Parch                418 non-null    int64   
 7   Ticket               418 non-null    object  
 8   Fare                 418 non-null    float64 
 9   Cabin                91 non-null     object  
 10  Embarked             418 non-null    object  
 11  Family_Size          418 non-null    int64   
 12  Family_Size_Grouped  418 non-null    object  
 13  Age_Cut              332 non-null    category
 14  Fare_Cut             418 non-null    category
dtypes: category(2), float64(2), int64(5), object(6)
memory usage: 44.1+ KB
```

```
train_df['Name']
```

```
0                                Braund, Mr. Owen Harris
1      Cumings, Mrs. John Bradley (Florence Briggs Th...
2                                 Heikkinen, Miss. Laina
3           Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                               Allen, Mr. William Henry
                             ...                        
886                                Montvila, Rev. Juozas
887                         Graham, Miss. Margaret Edith
888             Johnston, Miss. Catherine Helen "Carrie"
889                                Behr, Mr. Karl Howell
890                                  Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

```
train_df['Name'].str.split(pat= ",", expand=True) # Seperti yang dapat dilihat, bahwa ini maks
```

	0	1
0	Braund	Mr. Owen Harris
1	Cumings	Mrs. John Bradley (Florence Briggs Thayer)
2	Heikkinen	Miss. Laina
3	Futrelle	Mrs. Jacques Heath (Lily May Peel)
4	Allen	Mr. William Henry
...	...	...
886	Montvila	Rev. Juozas
887	Graham	Miss. Margaret Edith
888	Johnston	Miss. Catherine Helen "Carrie"
889	Behr	Mr. Karl Howell
890	Dooley	Mr. Patrick

891 rows × 2 columns

```
train_df['Name'].str.split(pat= ",", expand=True)[1] #Enggak peduli dengan last name mereka, p
```

```
0                                Mr. Owen Harris
1      Mrs. John Bradley (Florence Briggs Thayer)
2                                  Miss. Laina
3            Mrs. Jacques Heath (Lily May Peel)
4                              Mr. William Henry
                        ...
886                              Rev. Juozas
887                        Miss. Margaret Edith
888              Miss. Catherine Helen "Carrie"
889                              Mr. Karl Howell
890                                  Mr. Patrick
Name: 1, Length: 891, dtype: object
```

```
train_df['Name'].str.split(pat= ",", expand=True)[1].str.split(pat= ".", expand=True) # Nah l
```

	0	1	2
0	Mr	Owen Harris	None
1	Mrs	John Bradley (Florence Briggs Thayer)	None
2	Miss	Laina	None
3	Mrs	Jacques Heath (Lily May Peel)	None
4	Mr	William Henry	None
...	...	...	...
886	Rev	Juozas	None
887	Miss	Margaret Edith	None
888	Miss	Catherine Helen "Carrie"	None
889	Mr	Karl Howell	None
890	Mr	Patrick	None

891 rows × 3 columns

```
train_df['Name'].str.split(pat= ",", expand=True)[1].str.split(pat= ".", expand=True)[0]
```

```
0         Mr
1        Mrs
2       Miss
3        Mrs
4         Mr
       ...
886       Rev
887     Miss
888     Miss
889       Mr
890       Mr
Name: 0, Length: 891, dtype: object
```

```
train_df['Title'] = train_df['Name'].str.split(pat= ",", expand=True)[1].str.split(pat= ".", e
test_df['Title'] = test_df['Name'].str.split(pat= ",", expand=True)[1].str.split(pat= ".", exp
```

```
train_df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	2.00	1	0	A/5 21171	0.00	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	5.00	1	0	PC 17599	5.00	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	3.00	0	0	STON/O2. 3101282	1.00	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	5.00	1	0	113803	5.00	C123	
4	5	0	3	Allen, Mr. William Henry	male	5.00	0	0	373450	1.00	NaN	

```
train_df.groupby(["Title"], as_index=False,)["Survived"].mean()
```

	Title	Survived
0	Capt	0.00
1	Col	0.50
2	Don	0.00
3	Dr	0.43
4	Jonkheer	0.00
5	Lady	1.00
6	Major	0.50
7	Master	0.57
8	Miss	0.70
9	Mlle	1.00
10	Mme	1.00
11	Mr	0.16
12	Mrs	0.79
13	Ms	1.00
14	Rev	0.00
15	Sir	1.00
16	the Countess	1.00

```
train_df['Title'].value_counts() # Kode ini baru ditambahkan setelah penjelasan video tutorial
```

```
Title
Mr              517
Miss            182
Mrs             125
Master           40
Dr                7
Rev               6
Col               2
Mlle              2
Major             2
Ms                1
Mme               1
Don               1
Lady              1
Sir               1
Capt              1
the Countess      1
Jonkheer          1
Name: count, dtype: int64
```

```
train_df['Title'] = train_df['Title'].replace({
    'Capt': 'Military',
    'Col': 'Military',
    'Major': 'Military',
    'Jonkheer': 'Noble',
    'the Countess': 'Noble',
    'Don': 'Noble',
    'Lady': 'Noble',
    'Sir': 'Noble',
    'Mlle': 'Noble',
    'Ms': 'Noble',
    'Mme': 'Noble'
})

test_df['Title'] = test_df['Title'].replace({
    'Capt': 'Military',
    'Col': 'Military',
    'Major': 'Military',
    'Jonkheer': 'Noble',
    'the Countess': 'Noble',
    'Don': 'Noble',
    'Lady': 'Noble',
    'Sir': 'Noble',
    'Mlle': 'Noble',
    'Ms': 'Noble',
    'Mme': 'Noble'
})
```

```
train_df.groupby(["Title"], as_index=False)["Survived"].agg(['count', 'mean'])
```

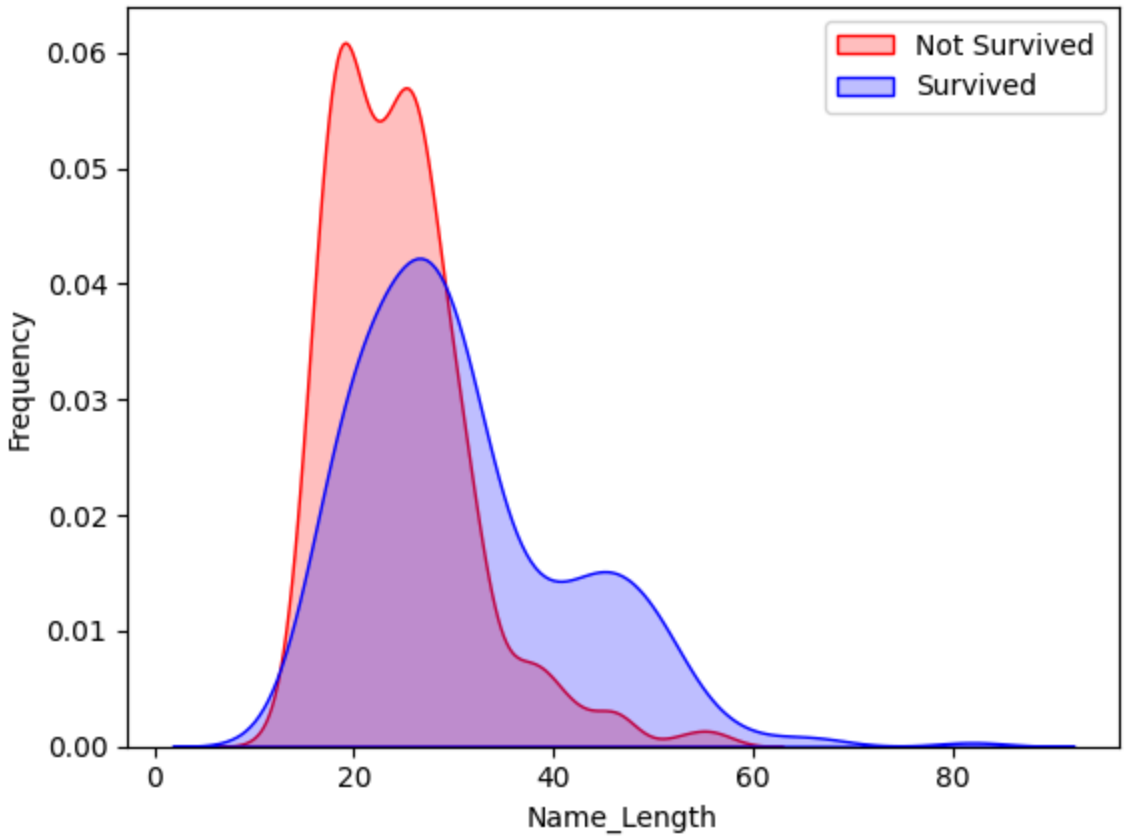
	Title	count	mean
0	Dr	7	0.43
1	Master	40	0.57
2	Military	5	0.40
3	Miss	182	0.70
4	Mr	517	0.16
5	Mrs	125	0.79
6	Noble	9	0.78
7	Rev	6	0.00

```
train_df['Name_Length'] = train_df['Name'].apply(lambda x: len(x))
test_df['Name_Length'] = test_df['Name'].apply(lambda x: len(x))
```

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   PassengerId          891 non-null    int64
 1   Survived             891 non-null    int64
 2   Pclass               891 non-null    int64
 3   Name                 891 non-null    object
 4   Sex                  891 non-null    object
 5   Age                  714 non-null    float64
 6   SibSp                891 non-null    int64
 7   Parch                891 non-null    int64
 8   Ticket               891 non-null    object
 9   Fare                 891 non-null    float64
 10  Cabin                204 non-null    object
 11  Embarked             889 non-null    object
 12  Family_Size          891 non-null    int64
 13  Family_Size_Grouped  891 non-null    object
 14  Age_Cut              714 non-null    category
 15  Fare_Cut             891 non-null    category
 16  Title                891 non-null    object
 17  Name_Length          891 non-null    int64
dtypes: category(2), float64(2), int64(7), object(7)
memory usage: 113.9+ KB
```

```
g = sns.kdeplot(train_df['Name_Length'][(train_df['Survived']==0) & (train_df['Name_Length'].r
g = sns.kdeplot(train_df['Name_Length'][(train_df['Survived']==1) & (train_df['Name_Length'].r
g.set_xlabel('Name_Length')
g.set_ylabel('Frequency')
g = g.legend(['Not Survived', 'Survived'])
```




```
train_df.groupby(["Name_Length"], as_index=False,)["Survived"].mean()
```

	Name_Length	Survived
0	12	0.50
1	13	0.50
2	14	0.33
3	15	0.13
4	16	0.23
5	17	0.21
6	18	0.20
7	19	0.23
8	20	0.28
9	21	0.33
10	22	0.32
11	23	0.28
12	24	0.37
13	25	0.33
14	26	0.22
15	27	0.36
16	28	0.37
17	29	0.50
18	30	0.43
19	31	0.40
20	32	0.57
21	33	0.55
22	34	0.43
23	35	1.00
24	36	0.33
25	37	0.70
26	38	0.44
27	39	0.44
28	40	0.43
29	41	1.00

	Name_Length	Survived
30	42	0.20
31	43	0.80
32	44	1.00
33	45	0.78
34	46	0.57
35	47	0.73
36	48	1.00
37	49	1.00
38	50	1.00
39	51	1.00
40	52	0.75
41	53	1.00
42	54	0.00
43	55	0.50
44	56	0.67
45	57	0.50
46	61	1.00
47	65	1.00
48	67	1.00
49	82	1.00

```
train_df['Name_LengthGB'] = pd.qcut(train_df['Name_Length'], 3)
test_df['Name_LengthGB'] = pd.qcut(test_df['Name_Length'], 3)
```

```
train_df.groupby(['Name_LengthGB'], as_index=False,  observed=False)['Survived'].mean()
```

	Name_LengthGB	Survived
0	(11.999, 22.0]	0.25
1	(22.0, 28.0]	0.32
2	(28.0, 82.0]	0.59

```
train_df.loc[train_df['Name_Length'] <= 22, 'Name_Size'] = 0
train_df.loc[(train_df['Name_Length'] > 22) & (train_df['Name_Length'] <= 28), 'Name_Size'] =
train_df.loc[(train_df['Name_Length'] > 28) & (train_df['Name_Length'] <= 82), 'Name_Size'] =
train_df.loc[train_df['Name_Length'] > 82, 'Name_Size']
```

```
test_df.loc[test_df['Name_Length'] <= 22, 'Name_Size'] = 0
test_df.loc[(test_df['Name_Length'] > 22) & (test_df['Name_Length'] <= 28), 'Name_Size'] = 1
test_df.loc[(test_df['Name_Length'] > 28) & (test_df['Name_Length'] <= 82), 'Name_Size'] = 2
test_df.loc[test_df['Name_Length'] > 82, 'Name_Size']
```

```
Series([], Name: Name_Size, dtype: float64)
```

```
train_df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	2.00	1	0	A/5 21171	0.00	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	5.00	1	0	PC 17599	5.00	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	3.00	0	0	STON/O2. 3101282	1.00	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	5.00	1	0	113803	5.00	C123	
4	5	0	3	Allen, Mr. William Henry	male	5.00	0	0	373450	1.00	NaN	

```
train_df['Ticket']
```

```
0            A/5 21171
1             PC 17599
2     STON/O2. 3101282
3               113803
4               373450
            ...
886             211536
887             112053
888         W./C. 6607
889             111369
890             370376
Name: Ticket, Length: 891, dtype: object
```

```
train_df['TicketNumber'] = train_df['Ticket'].apply(lambda x: pd.Series({'Ticket': x.split()[
test_df['TicketNumber'] = test_df['Ticket'].apply(lambda x: pd.Series({'Ticket': x.split()[-1
```

```
train_df['TicketNumber']
```

```
0        21171
1        17599
2      3101282
3       113803
4       373450
        ...
886     211536
887     112053
888       6607
889     111369
890     370376
Name: TicketNumber, Length: 891, dtype: object
```

```
train_df.groupby(['TicketNumber'], as_index=False,  observed=False)['Survived'].agg(['count',
```

	TicketNumber	count	mean
196	2343	7	0.00
464	347082	7	0.00
94	1601	7	0.71
168	2144	6	0.00
468	347088	6	0.00
...	...	...	...
674	8475	1	0.00
675	851	1	0.00
676	9234	1	1.00
63	11769	1	1.00
647	5727	1	0.00

679 rows × 3 columns

```
train_df.groupby('TicketNumber')['TicketNumber'].transform('count')
```

```
0      1
1      1
2      1
3      2
4      1
      ..
886    1
887    1
888    2
889    1
890    1
Name: TicketNumber, Length: 891, dtype: int64
```

```
train_df['TicketNumberCounts'] = train_df.groupby('TicketNumber')['TicketNumber'].transform('
test_df['TicketNumberCounts'] = test_df.groupby('TicketNumber')['TicketNumber'].transform('cou
```

```
train_df.groupby(['TicketNumberCounts'], as_index=False,  observed=False)['Survived'].agg(['co
```

	TicketNumberCounts	count	mean
0	1	544	0.30
1	2	188	0.57
2	3	66	0.71
3	4	44	0.50
6	7	21	0.24
5	6	18	0.00
4	5	10	0.00

```
train_df['Ticket']
```

```
0            A/5 21171
1             PC 17599
2     STON/O2. 3101282
3               113803
4               373450
            ...
886             211536
887             112053
888         W./C. 6607
889             111369
890             370376
Name: Ticket, Length: 891, dtype: object
```

```
train_df['Ticket'].str.split(pat=" ", expand=True)
```

	0	1	2
0	A/5	21171	None
1	PC	17599	None
2	STON/O2.	3101282	None
3	113803	None	None
4	373450	None	None
...	...	...	...
886	211536	None	None
887	112053	None	None
888	W./C.	6607	None
889	111369	None	None
890	370376	None	None

891 rows × 3 columns

```
train_df['TicketLocation'] = np.where(train_df['Ticket'].str.split(pat=" ", expand=True)[1].nc
test_df['TicketLocation'] = np.where(test_df['Ticket'].str.split(pat=" ", expand=True)[1].notr
```

```
train_df['TicketLocation'].value_counts()
```

```
TicketLocation
Blank         665
PC             60
C.A.           27
STON/O         12
A/5            10
W./C.           9
CA.             8
SOTON/O.Q.      8
A/5.            7
SOTON/OQ        7
STON/O2.        6
CA              6
C               5
S.O.C.          5
SC/PARIS        5
F.C.C.          5
SC/Paris        4
A/4.            3
PP              3
A/4             3
S.O./P.P.       3
SC/AH           3
A./5.           2
P/PP            2
A.5.            2
WE/P            2
SOTON/O2        2
S.C./PARIS      2
S.C./A.4.       1
Fa              1
S.O.P.          1
SO/C            1
S.P.            1
A4.             1
W.E.P.          1
A/S             1
SC              1
SW/PP           1
SCO/W           1
W/C             1
S.W./PP         1
F.C.            1
C.A./SOTON      1
Name: count, dtype: int64
```

```
train_df['TicketLocation'] = train_df['TicketLocation'].replace({
    'SOTON/O.Q.':'SOTON/OQ',
    'C.A.':'CA',
    'CA.':'CA',
    'SC/PARIS':'SC/Paris',
    'S.C./PARIS':'SC/Paris',
    'A/4.':'A/4',
    'A/5.':'A/5',
    'A.5.':'A/5',
    'A./5.':'A/5',
```

```
    'W./C.':'W/C',
})

test_df['TicketLocation'] = test_df['TicketLocation'].replace({
    'SOTON/O.Q.':'SOTON/OQ',
    'C.A.':'CA',
    'CA.':'CA',
    'SC/PARIS':'SC/Paris',
    'S.C./PARIS':'SC/Paris',
    'A/4.':'A/4',
    'A/5.':'A/5',
    'A.5.':'A/5',
    'A./5.':'A/5',
    'W./C.':'W/C',
})
```

```
train_df.groupby(['TicketLocation'], as_index=False)['Survived'].agg(['count', 'mean'])
```

	TicketLocation	count	mean
0	A/4	6	0.00
1	A/5	21	0.10
2	A/S	1	0.00
3	A4.	1	0.00
4	Blank	665	0.38
5	C	5	0.40
6	C.A./SOTON	1	0.00
7	CA	41	0.34
8	F.C.	1	0.00
9	F.C.C.	5	0.80
10	Fa	1	0.00
11	P/PP	2	0.50
12	PC	60	0.65
13	PP	3	0.67
14	S.C./A.4.	1	0.00
15	S.O./P.P.	3	0.00
16	S.O.C.	5	0.00
17	S.O.P.	1	0.00
18	S.P.	1	0.00
19	S.W./PP	1	1.00
20	SC	1	1.00
21	SC/AH	3	0.67
22	SC/Paris	11	0.45
23	SCO/W	1	0.00
24	SO/C	1	1.00
25	SOTON/O2	2	0.00
26	SOTON/OQ	15	0.13
27	STON/O	12	0.42
28	STON/O2.	6	0.50
29	SW/PP	1	1.00

	TicketLocation	count	mean
30	W.E.P.	1	0.00
31	W/C	10	0.10
32	WE/P	2	0.50

```
train_df['Cabin']
```

```
0       NaN
1       C85
2       NaN
3      C123
4       NaN
       ...
886     NaN
887     B42
888     NaN
889    C148
890     NaN
Name: Cabin, Length: 891, dtype: object
```

```
train_df['Cabin'] = train_df['Cabin'].fillna('U')
train_df['Cabin'] = pd.Series([i[0] if not pd.isnull(i) else 'x' for i in train_df['Cabin']])

test_df['Cabin'] = test_df['Cabin'].fillna('U')
test_df['Cabin'] = pd.Series([i[0] if not pd.isnull(i) else 'x' for i in test_df['Cabin']])
```

```
train_df.groupby(['Cabin'], as_index=False)['Survived'].agg(['count', 'mean'])
```

	Cabin	count	mean
0	A	15	0.47
1	B	47	0.74
2	C	59	0.59
3	D	33	0.76
4	E	32	0.75
5	F	13	0.62
6	G	4	0.50
7	T	1	0.00
8	U	687	0.30

```
train_df['Cabin_Assigned'] = train_df['Cabin'].apply(lambda x: 0 if x in ['U'] else 1)
test_df['Cabin_Assigned'] = test_df['Cabin'].apply(lambda x: 0 if x in ['U'] else 1)
```

```
train_df.groupby(['Cabin_Assigned'], as_index=False)['Survived'].agg(['count', 'mean'])
```

	Cabin_Assigned	count	mean
0	0	687	0.30
1	1	204	0.67

```
train_df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	2.00	1	0	A/5 21171	0.00	U	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	5.00	1	0	PC 17599	5.00	C	
2	3	1	3	Heikkinen, Miss. Laina	female	3.00	0	0	STON/O2. 3101282	1.00	U	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	5.00	1	0	113803	5.00	C	
4	5	0	3	Allen, Mr. William Henry	male	5.00	0	0	373450	1.00	U	

```
test_df.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Family
0	892	3	Kelly, Mr. James	male	5.00	0	0	330911	1.00	U	Q	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	6.00	1	0	363272	0.00	U	S	
2	894	2	Myles, Mr. Thomas Francis	male	7.00	0	0	240276	2.00	U	Q	
3	895	3	Wirz, Mr. Albert	male	3.00	0	0	315154	2.00	U	S	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	2.00	1	1	3101298	2.00	U	S	

```
train_df.shape
```

```
(891, 24)
```

```
test_df.shape
```

```
(418, 23)
```

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 24 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   PassengerId          891 non-null    int64
 1   Survived             891 non-null    int64
 2   Pclass               891 non-null    int64
 3   Name                 891 non-null    object
 4   Sex                  891 non-null    object
 5   Age                  714 non-null    float64
 6   SibSp                891 non-null    int64
 7   Parch                891 non-null    int64
 8   Ticket               891 non-null    object
 9   Fare                 891 non-null    float64
 10  Cabin                891 non-null    object
 11  Embarked             889 non-null    object
 12  Family_Size          891 non-null    int64
 13  Family_Size_Grouped  891 non-null    object
 14  Age_Cut              714 non-null    category
 15  Fare_Cut             891 non-null    category
 16  Title                891 non-null    object
 17  Name_Length          891 non-null    int64
 18  Name_LengthGB        891 non-null    category
 19  Name_Size            891 non-null    float64
 20  TicketNumber         891 non-null    object
 21  TicketNumberCounts   891 non-null    int64
 22  TicketLocation       891 non-null    object
 23  Cabin_Assigned       891 non-null    int64
dtypes: category(3), float64(3), int64(9), object(9)
memory usage: 149.8+ KB
```

```
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 23 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   PassengerId          418 non-null    int64
 1   Pclass               418 non-null    int64
 2   Name                 418 non-null    object
 3   Sex                  418 non-null    object
 4   Age                  332 non-null    float64
 5   SibSp                418 non-null    int64
 6   Parch                418 non-null    int64
 7   Ticket               418 non-null    object
 8   Fare                 418 non-null    float64
 9   Cabin                418 non-null    object
 10  Embarked             418 non-null    object
 11  Family_Size          418 non-null    int64
 12  Family_Size_Grouped  418 non-null    object
 13  Age_Cut              332 non-null    category
 14  Fare_Cut             418 non-null    category
 15  Title                418 non-null    object
 16  Name_Length          418 non-null    int64
 17  Name_LengthGB        418 non-null    category
 18  Name_Size            418 non-null    float64
 19  TicketNumber         418 non-null    object
 20  TicketNumberCounts   418 non-null    int64
 21  TicketLocation       418 non-null    object
 22  Cabin_Assigned       418 non-null    int64
dtypes: category(3), float64(3), int64(8), object(9)
memory usage: 67.5+ KB
```

```
train_df.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked', 'Family_Size',
       'Family_Size_Grouped', 'Age_Cut', 'Fare_Cut', 'Title', 'Name_Length',
       'Name_LengthGB', 'Name_Size', 'TicketNumber', 'TicketNumberCounts',
       'TicketLocation', 'Cabin_Assigned'],
      dtype='object')
```

```
train_df['Age'] = train_df['Age'].fillna(train_df['Age'].mean())
test_df['Age'] = test_df['Age'].fillna(test_df['Age'].mean())
# train_df['Age'].fillna(train_df['Age'].mean(),inplace=True) not the best use/practises
# test_df['Age'].fillna(test_df['Age'].mean(),inplace=True)
# test_df['Fare'].fillna(test_df['Fare'].mean(),inplace=True) ini baru ditulis di video tutori
```

```
ohe = OneHotEncoder(sparse_output=False)
ode = OrdinalEncoder
SI = SimpleImputer(strategy='most_frequent')
```

```
ode_cols =['Family_Size_Grouped']
ohe_cols = ['Sex', 'Embarked']
```

```
X = train_df.drop(['Survived'], axis=1)
y = train_df['Survived']
```

```
X_test = test_df.drop(['Age_Cut', 'Fare_Cut'], axis=1)
```

```
X.head(1)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Family_Size
0	1	3	Braund, Mr. Owen Harris	male	2.00	1	0	A/5 21171	0.00	U	S	2

```
y.head(10)
```

```
0    0
1    1
2    1
3    1
4    0
5    0
6    0
7    0
8    1
9    1
Name: Survived, dtype: int64
```

```
X_test.head(1)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Family_Size
0	892	3	Kelly, Mr. James	male	5.00	0	0	330911	1.00	U	Q	1

```
# X = train_df.drop(['Survived'], axis=1)
# y = train_df['Survived']
# X_test = test_df.drop(['Age_Cut', 'Fare_Cut'], axis=1)
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2, stratify = y, randc
```

```
ordinal_pipeline = Pipeline(steps=[
     ('impute', SimpleImputer(strategy='most_frequent')),
     ('ord', OrdinalEncoder(handle_unknown= 'use_encoded_value', unknown_value=-1))
])
```

```
ohe_pipeline = Pipeline(steps=[
     ('impute', SimpleImputer(strategy='most_frequent')),
     ('one-hot ', OneHotEncoder(handle_unknown='ignore', sparse_output=False))
])
```

```
col_trans = ColumnTransformer(transformers=[
    ('impute', SI, ['Age']),
    ('ord_pipeline', ordinal_pipeline, ode_cols),
```

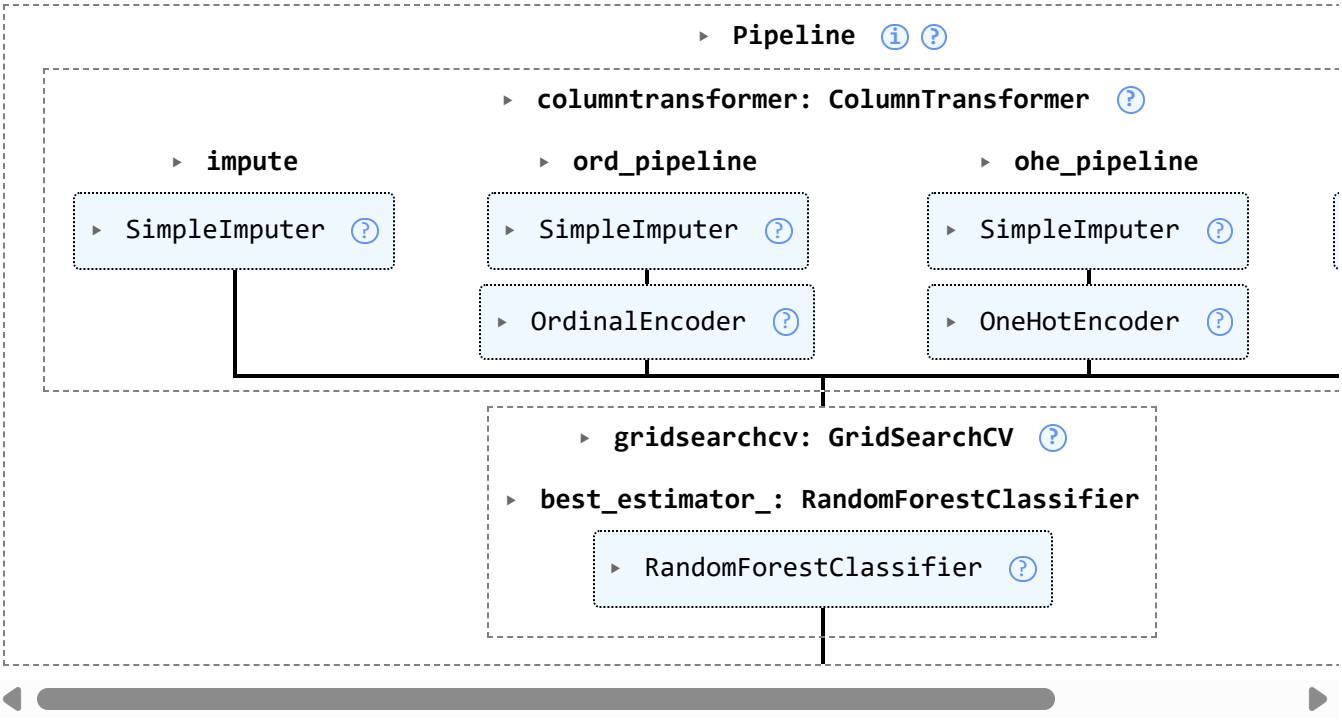
```
    ('ohe_pipeline', ohe_pipeline, ohe_cols),
    ('passthrough', 'passthrough', ['Pclass', 'TicketNumberCounts', 'Cabin_Assigned', 'Name_Si
    ],
    remainder='drop',
    n_jobs=-1)
```

```
rfc = RandomForestClassifier()
```

```
param_grid = {
    'n_estimators': [100,150,200],
    'min_samples_split': [5,10,15],
    'max_depth': [8,9,10,15,20],
    'min_samples_leaf': [1,2,4],
    'criterion': ['gini', 'entropy'],
}
```

```
CV_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv=StratifiedKFold(n_splits=5))
```

```
pipefinalrfc = make_pipeline(col_trans, CV_rfc)
pipefinalrfc.fit(X_train, y_train)
```




```
print(CV_rfc.best_params_)
print(CV_rfc.best_score_)
```

```
{'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_
estimators': 100}
0.8328572835615089
```

```
Y_pred = pipefinalrfc.predict(X_test)
```

```
submission = pd.DataFrame({
    'PassengerID' : test_df['PassengerId'],
```

```
    'Survived': Y_pred
})
```

```
submission.to_csv('../data/submissionTitanic_1.csv', index=False)
```