

```
In [17]: #8.1
import pandas as pd
df = pd.read_csv("Z:\ml\dataset.csv")
selected_columns = ['column1', 'column3', 'column5']
df_selected = df[selected_columns]
print("selected Attributes:")
print(df_selected.head())
```

```
selected Attributes:
   column1 column3 column5
0         1      Red      50
1         2  Yellow      30
2         3  Orange      40
3         4  Purple      60
4         5   Pink      70
```

```
In [18]: #8.2
import pandas as pd
df=pd.read_csv("Z:\ml\dataset.csv")
df_dropped=df.dropna()
print("Data after Dropping Missing Values:")
print(df_dropped.head())
```

```
Data after Dropping Missing Values:
   column1 column2 column3 column4 column5
0         1   Apple      Red      10      50
1         2  Banana  Yellow      20      30
2         3  Orange  Orange      15      40
3         4   Grape  Purple      25      60
4         5   Peach   Pink      30      70
```

```
In [21]: #8.3
import pandas as pd
from sklearn.preprocessing import KBinsDiscretizer
df=pd.read_csv("Z:\ml\dataset.csv")
kbins=KBinsDiscretizer(n_bins=3,encode='ordinal',strategy='uniform')
df['column1_binned']=kbins.fit_transform(df[['column1']])
print("Data After Discretization:")
print(df[['column1','column1_binned']].head())
```

```
Data After Discretization:
   column1 column1_binned
0         1             0.0
1         2             0.0
2         3             1.0
3         4             2.0
4         5             2.0
```

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\preprocessing\\_discretization.py:239: FutureWarning: In version 1.5 onwards, subsample=200\_000 will be used by default. Set subsample explicitly to silence this warning in the mean time. Set subsample=None to disable subsampling explicitly.

```
warnings.warn(
```

```
In [25]: #8.4
import pandas as pd
import numpy as np
from scipy import stats

df = pd.read_csv("Z:\\ml\\dataset.csv")
df['column1'] = pd.to_numeric(df['column1'], errors='coerce')
df['column2'] = pd.to_numeric(df['column2'], errors='coerce')
df = df.dropna(subset=['column1', 'column2'])

z_scores = np.abs(stats.zscore(df[['column1', 'column2']]))
df_no_outliers = df[(z_scores < 3).all(axis=1)]
print("Data after Eliminating Outliers:")
print(df_no_outliers.head())
```

Data after Eliminating Outliers:  
 Empty DataFrame  
 Columns: [column1, column2, column3, column4, column5]  
 Index: []

In [ ]:

```
In [41]: #9.1
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

from sklearn.datasets import load_iris

data = load_iris()
df = pd.DataFrame(data.data, columns = data.feature_names)
df['target'] = data.target
X = df.drop('target', axis=1)
y = df['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

knn_classifier = KNeighborsClassifier(n_neighbors=3)
knn_classifier.fit(X_train, y_train)

y_pred = knn_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of KNN classifier:{accuracy*100:.2f}%")
```

Accuracy of KNN classifier:42.22%

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names  
 warnings.warn(

```
In [46]: #9.2
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error

from sklearn.datasets import fetch_california_housing

data = fetch_california_housing()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

knn_classifier = KNeighborsClassifier(n_neighbors=5)
knn_classifier.fit(X_train, y_train)

y_pred = knn_classifier.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error of KNN regressor: {mse:.2f}")
```

Mean Squared Error of KNN regressor:1.38

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names  
warnings.warn(

```

In [44]: #10
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV

data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

X = df.drop('target', axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

dt_classifier = DecisionTreeClassifier(random_state=42)
dt_classifier.fit(X_train, y_train)
y_pred = dt_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of Decision Tree without tuning: {accuracy * 100:.2f}%")

param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30, 40],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2']
}

grid_search = GridSearchCV(estimator=dt_classifier, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
y_pred_best = best_model.predict(X_test)
best_accuracy = accuracy_score(y_test, y_pred_best)

print(f"Best model accuracy: {best_accuracy * 100:.2f}%")
print(f"Best hyperparameters: {best_params}")

```

Accuracy of Decision Tree without tuning: 100.00%  
 Fitting 5 folds for each of 270 candidates, totalling 1350 fits  
 Best model accuracy: 97.78%  
 Best hyperparameters: {'criterion': 'gini', 'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10}

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:425: FitFailedWarning:
450 fits failed out of a total of 1350.
The score on these train-test partitions for these parameters will be set
to nan.
If these failures are not expected, you can try to debug them by setting e
rror_score='raise'.
```

Below are more details about the failures:

```
-----
-----
258 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in wrapper
    estimator._validate_params()
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features'
parameter of DecisionTreeClassifier must be an int in the range [1, inf),
a float in the range (0.0, 1.0], a str among {'sqrt', 'log2'} or None. Got
'auto' instead.
```

```
-----
-----
192 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in wrapper
    estimator._validate_params()
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features'
parameter of DecisionTreeClassifier must be an int in the range [1, inf),
a float in the range (0.0, 1.0], a str among {'log2', 'sqrt'} or None. Got
'auto' instead.
```

```
warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: UserWarning: One or more of the test scores are non-finite: [
nan          nan          nan          nan          nan          nan
      nan          nan          nan 0.91428571 0.92380952 0.93333333
0.92380952 0.93333333 0.93333333 0.92380952 0.92380952 0.92380952
0.91428571 0.92380952 0.93333333 0.92380952 0.93333333 0.93333333
0.92380952 0.92380952 0.92380952          nan          nan          nan
      nan          nan          nan          nan          nan          nan
0.91428571 0.92380952 0.93333333 0.92380952 0.93333333 0.93333333]
```

```
0.92380952 0.92380952 0.92380952 0.91428571 0.92380952 0.93333333
0.92380952 0.93333333 0.93333333 0.92380952 0.92380952 0.92380952
      nan      nan      nan      nan      nan      nan
      nan      nan      nan 0.91428571 0.92380952 0.93333333
0.92380952 0.93333333 0.93333333 0.92380952 0.92380952 0.92380952
0.91428571 0.92380952 0.93333333 0.92380952 0.93333333 0.93333333
0.92380952 0.92380952 0.92380952      nan      nan      nan
      nan      nan      nan      nan      nan      nan
0.91428571 0.92380952 0.93333333 0.92380952 0.93333333 0.93333333
0.92380952 0.92380952 0.92380952 0.91428571 0.92380952 0.93333333
0.92380952 0.93333333 0.93333333 0.92380952 0.92380952 0.92380952
      nan      nan      nan      nan      nan      nan
      nan      nan      nan 0.91428571 0.92380952 0.93333333
0.92380952 0.93333333 0.93333333 0.92380952 0.92380952 0.92380952
0.91428571 0.92380952 0.93333333 0.92380952 0.93333333 0.93333333
0.92380952 0.92380952 0.92380952      nan      nan      nan
      nan      nan      nan      nan      nan      nan
0.9047619 0.91428571 0.92380952 0.92380952 0.93333333 0.92380952
0.91428571 0.91428571 0.91428571 0.9047619 0.91428571 0.92380952
0.92380952 0.93333333 0.92380952 0.91428571 0.91428571 0.91428571
      nan      nan      nan      nan      nan      nan
      nan      nan      nan 0.9047619 0.91428571 0.92380952
0.92380952 0.93333333 0.92380952 0.91428571 0.91428571 0.91428571
0.9047619 0.91428571 0.92380952 0.92380952 0.93333333 0.92380952
0.91428571 0.91428571 0.92380952 0.92380952 0.93333333 0.92380952
0.91428571 0.91428571 0.91428571      nan      nan      nan
      nan      nan      nan      nan      nan      nan
0.9047619 0.91428571 0.92380952 0.92380952 0.93333333 0.92380952
0.91428571 0.91428571 0.91428571 0.9047619 0.91428571 0.92380952
0.92380952 0.93333333 0.92380952 0.91428571 0.91428571 0.91428571]
warnings.warn(
```

In [ ]: