9/2/2021 Lab 07 - CSE422.ipynb - Colaboratory



* import pandas as pd

2 import numpy as np

3

4 dataset = pd.read\_csv('/content/mushroom edibility classification dataset.csv') 5 dataset.head()

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | **sta** |  |
|  | **Unnamed:** | **class** | **cap-** | **cap-** | **cap-** | **bruises** | **odor** | **stalk- stalk- surfa** | | |  |
|  | **0** |  | **shape** | **surface** | **color** |  |  | **shape** | **root** | **abo** |  |
|  |  |  |  |  |  |  |  |  |  | **r** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **0** | 0 | poisonous | 5.0 | 2 | 4.0 | bruises | 6 | 0 | 3 |  |  |
| is there |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **1** | 1 | edible | 5.0 | 2 | 9.0 | bruises | 0 | 0 | 2 |  |  |
| is there |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **2** | 2 | edible | 0.0 | 2 | 8.0 | bruises | 3 | 0 | 2 |  |  |
| is there |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **3** | 3 | poisonous | 5.0 | 3 | 8.0 | bruises | 6 | 0 | 3 |  |  |
| is there |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **4** | 4 | edible | 5.0 | 2 | 3.0 | no | 5 | 1 | 3 |  |  |
| bruises |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | dataset.shape | |  |
|  |  |  |  |
|  | (3124, | 20) |  |
|  |  | |  |
|  |  | |  |
| 1 | dataset.isnull().sum() | |  |
|  |  | |  |
|  | Unnamed: 0 | | 0 |
|  | class |  | 0 |
|  | cap-shape | | 8 |
|  | cap-surface | | 0 |
|  | cap-color | | 8 |
|  | bruises |  | 0 |
|  | odor |  | 0 |
|  | stalk-shape | | 0 |
|  | stalk-root | | 0 |
|  | stalk-surface-above-ring | | 0 |
|  | stalk-surface-below-ring | | 0 |
|  | stalk-color-above-ring | | 0 |
|  | stalk-color-below-ring | | 0 |
|  | veil-type | | 0 |
|  | veil-color | | 0 |
|  | ring-number | | 0 |
|  | ring-type | | 0 |
|  | spore-print-color | | 0 |
|  | population | | 0 |
|  | habitat |  | 0 |
|  | dtype: int64 | |  |
|  |  | | |
|  |  | | |
| 1 | from sklearn.impute import SimpleImputer | | |

https://colab.research.google.com/drive/1\_-3VMTsgTrUa5Iyw0UPh9dXSkXhf6tAg#scrollTo=kyPhIVVbY1Pl&printMode=true 1/6

9/2/2021 Lab 07 - CSE422.ipynb - Colaboratory

* impute = SimpleImputer(missing\_values=np.nan, strategy='mean')

3 impute.fit(dataset[['cap-shape']])

4 dataset[['cap-shape']] = impute.transform(dataset[['cap-shape']])

5 impute.fit(dataset[['cap-color']])

6 dataset[['cap-color']] = impute.transform(dataset[['cap-color']])

7 dataset.isnull().sum()



|  |  |
| --- | --- |
| Unnamed: 0 | 0 |
| class | 0 |
| cap-shape | 0 |
| cap-surface | 0 |
| cap-color | 0 |
| bruises | 0 |
| odor | 0 |
| stalk-shape | 0 |
| stalk-root | 0 |
| stalk-surface-above-ring | 0 |
| stalk-surface-below-ring | 0 |
| stalk-color-above-ring | 0 |
| stalk-color-below-ring | 0 |
| veil-type | 0 |
| veil-color | 0 |
| ring-number | 0 |
| ring-type | 0 |
| spore-print-color | 0 |
| population | 0 |
| habitat | 0 |
| dtype: int64 |  |

* dataset.info()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <class 'pandas.core.frame.DataFrame'> | | | |  |
| RangeIndex: 3124 entries, 0 to 3123 | | |  |  |
| Data | columns (total 20 columns): | |  |  |
| # | Column | Non-Null Count | | Dtype |
| --- | ------ | -------------- | | ----- |
| 0 | Unnamed: 0 | 3124 | non-null | int64 |
| 1 | class | 3124 | non-null | object |
| 2 | cap-shape | 3124 | non-null | float64 |
| 3 | cap-surface | 3124 | non-null | int64 |
| 4 | cap-color | 3124 | non-null | float64 |
| 5 | bruises | 3124 | non-null | object |
| 6 | odor | 3124 | non-null | int64 |
| 7 | stalk-shape | 3124 | non-null | int64 |
| 8 | stalk-root | 3124 | non-null | int64 |

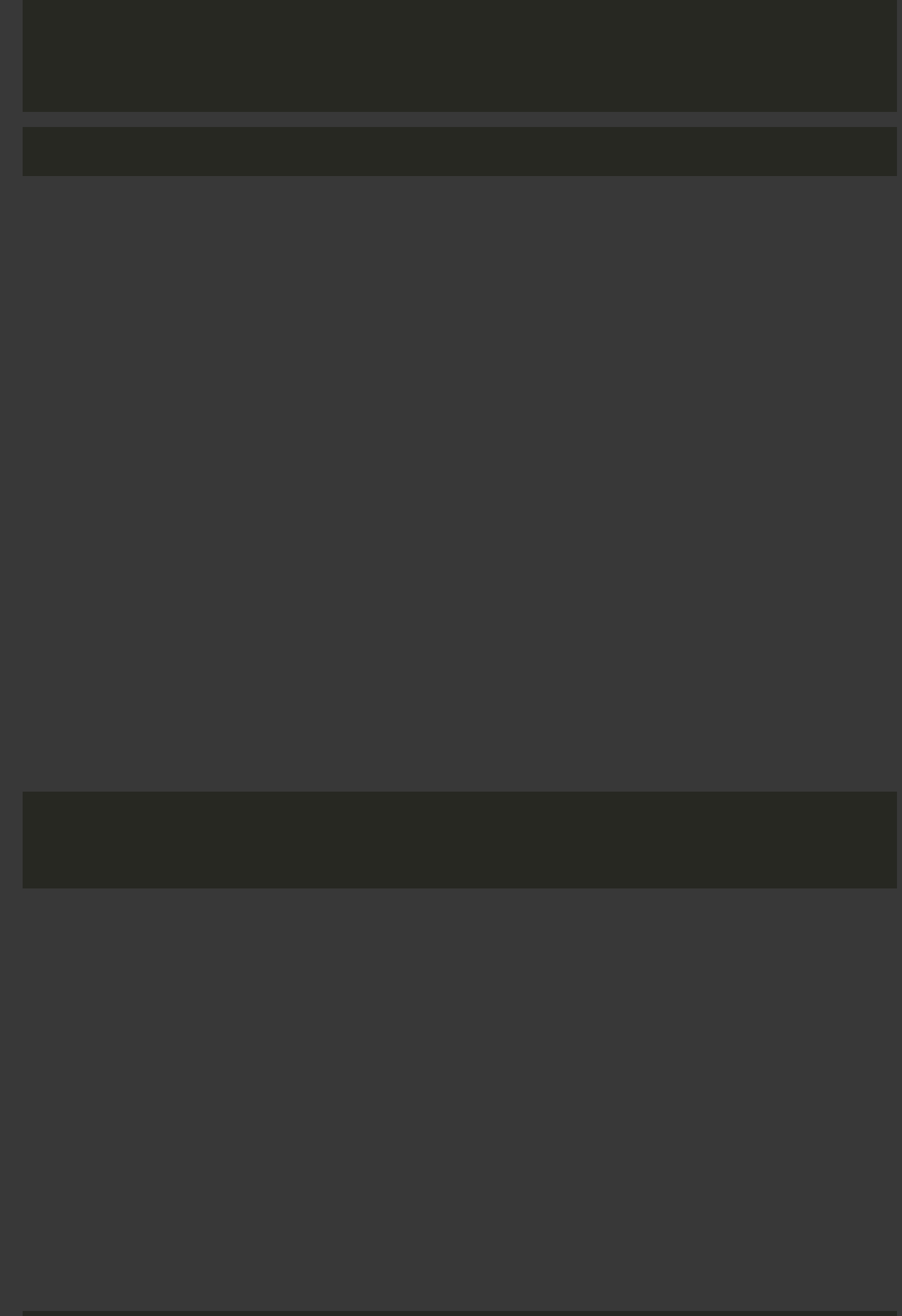
* stalk-surface-above-ring 3124 non-null int64

10 stalk-surface-below-ring 3124 non-null int64

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | stalk-color-above-ring | 3124 | non-null | int64 |
| 12 | stalk-color-below-ring | 3124 | non-null | int64 |
| 13 | veil-type | 3124 | non-null | int64 |
| 14 | veil-color | 3124 | non-null | int64 |
| 15 | ring-number | 3124 | non-null | int64 |
| 16 | ring-type | 3124 | non-null | int64 |
| 17 | spore-print-color | 3124 | non-null | int64 |
| 18 | population | 3124 | non-null | int64 |
| 19 | habitat | 3124 | non-null | int64 |
| dtypes: float64(2), int64(16), | | object(2) | |  |
| memory usage: 488.2+ KB | |  |  |  |

https://colab.research.google.com/drive/1\_-3VMTsgTrUa5Iyw0UPh9dXSkXhf6tAg#scrollTo=kyPhIVVbY1Pl&printMode=true 2/6

9/2/2021 Lab 07 - CSE422.ipynb - Colaboratory



* from sklearn.preprocessing import LabelEncoder

2 encoder = LabelEncoder()

3 dataset['class'] = encoder.fit\_transform(dataset['class'])

4 dataset['bruises'] = encoder.fit\_transform(dataset['bruises'])

* dataset.info()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <class 'pandas.core.frame.DataFrame'> | | | |  |
| RangeIndex: 3124 entries, 0 to 3123 | | |  |  |
| Data | columns (total 20 columns): | |  |  |
| # | Column | Non-Null Count | | Dtype |
| --- | ------ | -------------- | | ----- |
| 0 | Unnamed: 0 | 3124 | non-null | int64 |
| 1 | class | 3124 | non-null | int64 |
| 2 | cap-shape | 3124 | non-null | float64 |
| 3 | cap-surface | 3124 | non-null | int64 |
| 4 | cap-color | 3124 | non-null | float64 |
| 5 | bruises | 3124 | non-null | int64 |
| 6 | odor | 3124 | non-null | int64 |
| 7 | stalk-shape | 3124 | non-null | int64 |
| 8 | stalk-root | 3124 | non-null | int64 |

* stalk-surface-above-ring 3124 non-null int64

10 stalk-surface-below-ring 3124 non-null int64

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | stalk-color-above-ring | 3124 | non-null | int64 |
| 12 | stalk-color-below-ring | 3124 | non-null | int64 |
| 13 | veil-type | 3124 | non-null | int64 |
| 14 | veil-color | 3124 | non-null | int64 |
| 15 | ring-number | 3124 | non-null | int64 |
| 16 | ring-type | 3124 | non-null | int64 |
| 17 | spore-print-color | 3124 | non-null | int64 |
| 18 | population | 3124 | non-null | int64 |
| 19 | habitat | 3124 | non-null | int64 |

dtypes: float64(2), int64(18)

memory usage: 488.2 KB

* features\_list = dataset.columns.to\_list()

2 features\_list = features\_list[2:]

3 features\_list

['cap-shape',

'cap-surface',

'cap-color',

'bruises',

'odor',

'stalk-shape',

'stalk-root',

'stalk-surface-above-ring',

'stalk-surface-below-ring',

'stalk-color-above-ring',

'stalk-color-below-ring',

'veil-type',

'veil-color',

'ring-number',

'ring-type',

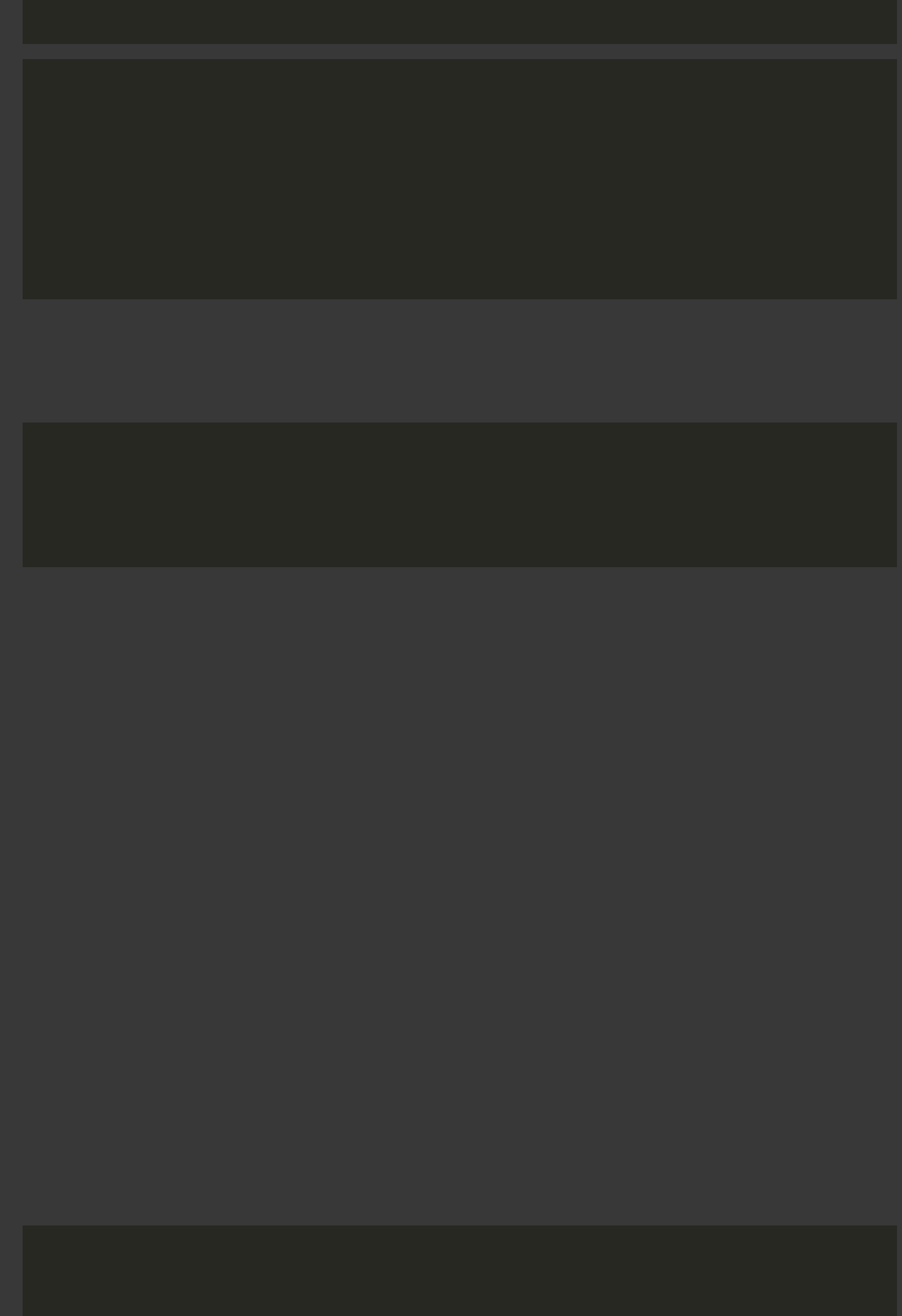
'spore-print-color',

'population',

'habitat']

https://colab.research.google.com/drive/1\_-3VMTsgTrUa5Iyw0UPh9dXSkXhf6tAg#scrollTo=kyPhIVVbY1Pl&printMode=true 3/6

9/2/2021 Lab 07 - CSE422.ipynb - Colaboratory



* label = ['class']
* x = dataset[features\_list]

2 y = dataset[label]

3

4 from sklearn.model\_selection import train\_test\_split

5 x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state 6 print(x\_train.shape)

7 print(x\_test.shape)

8 print(y\_train.shape)

9 print(y\_test.shape)

(2499, 18)

(625, 18)

(2499, 1)

(625, 1)

* from sklearn.linear\_model import LogisticRegression

2 regression = LogisticRegression(max\_iter=1000)

3 regression.fit(x\_train, y\_train.values.ravel())

4 predictions = regression.predict(x\_test)

5 predictions

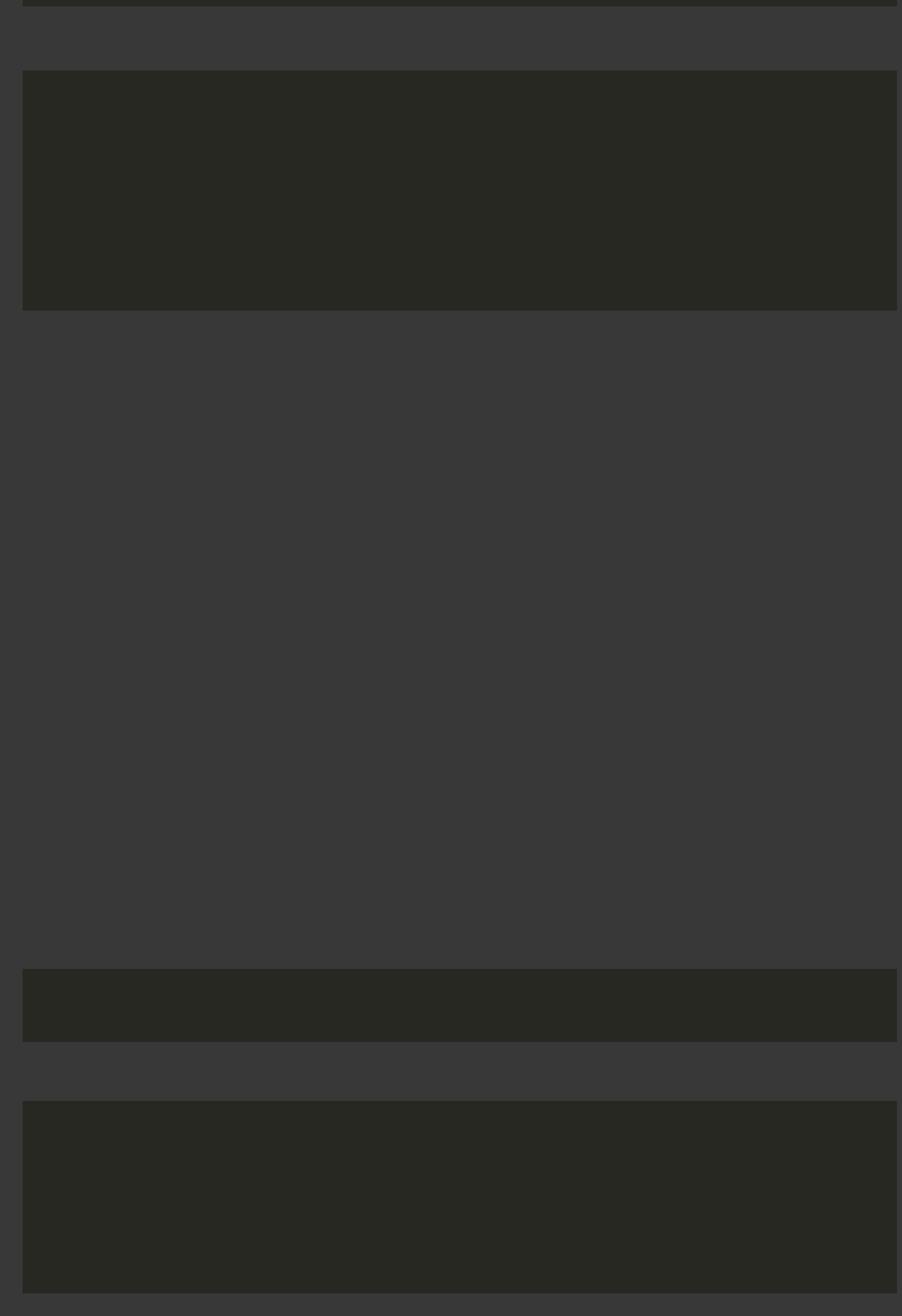
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0])

* from sklearn.metrics import accuracy\_score
* regression\_accuracy = accuracy\_score(y\_test, predictions)

3 print(f"Accuracy using LogisticRegression: {regression\_accuracy \* 100}%")

https://colab.research.google.com/drive/1\_-3VMTsgTrUa5Iyw0UPh9dXSkXhf6tAg#scrollTo=kyPhIVVbY1Pl&printMode=true 4/6

9/2/2021 Lab 07 - CSE422.ipynb - Colaboratory



Accuracy using LogisticRegression: 98.72%

* from sklearn.tree import DecisionTreeClassifier
* from sklearn.model\_selection import train\_test\_split

3 X = dataset[features\_list]

4 Y = dataset[label]

5 X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state 6 classifier = DecisionTreeClassifier(criterion='entropy', random\_state=3)

7 classifier.fit(X\_train,Y\_train)

8 predictions\_new = classifier.predict(X\_test)

9 predictions\_new

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0])

* classifier\_accuracy = accuracy\_score(Y\_prediction,Y\_test)
* print(f"Accuracy using DecisionTreeClassifier: {classifier\_accuracy \* 100}%")

Accuracy using DecisionTreeClassifier: 100.0%

* import matplotlib.pyplot as plt

2 plt.xlabel("Algorithm Name")

3 plt.ylabel("Score")

4 x\_axis=["Logistic Regression", "Decision Tree Classifier"]

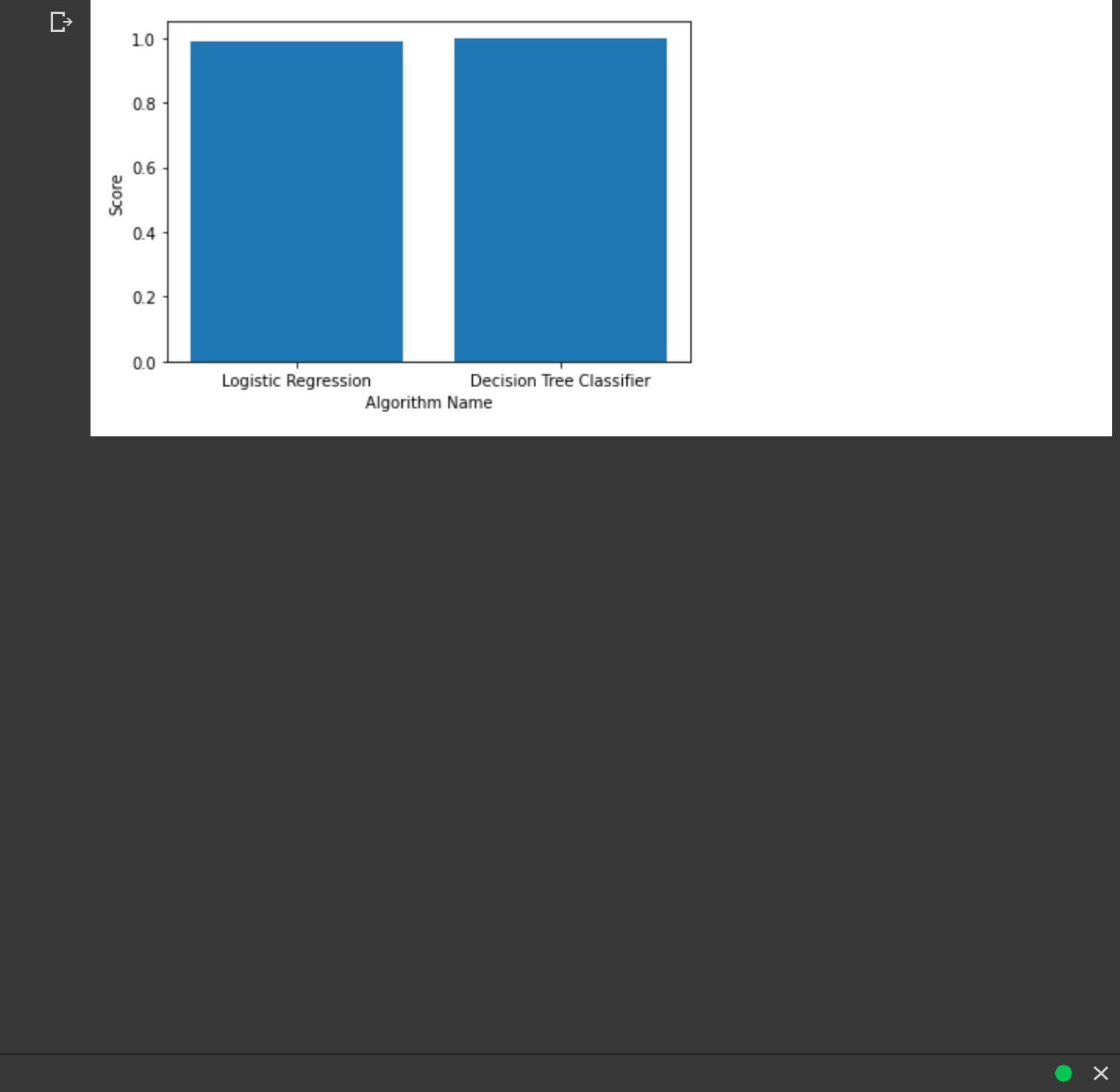
5 y\_axis=[regression\_accuracy, classifier\_accuracy]

6 plt.bar(x\_axis, y\_axis)

7 plt.show()

https://colab.research.google.com/drive/1\_-3VMTsgTrUa5Iyw0UPh9dXSkXhf6tAg#scrollTo=kyPhIVVbY1Pl&printMode=true 5/6

9/2/2021 Lab 07 - CSE422.ipynb - Colaboratory



* 0s completed at 10:43 PM

https://colab.research.google.com/drive/1\_-3VMTsgTrUa5Iyw0UPh9dXSkXhf6tAg#scrollTo=kyPhIVVbY1Pl&printMode=true 6/6