Arnob Dey | ID : 203-15-3906| Section : PC - B | Subject: Artificial Intelligence Lab | Course Code: CSE316 |

List, tuple related experiments | Lab Report 1

```
#Tuple Related Experiment

tpl = ()
print('Enter an integer to add to the tuple. Enter 0 to stop.')
while True:
    num = input("Enter a number: ")
    if num == '0':
        break
    try:
        num = int(num)
    except:
        print("Invalid input")
        continue
    tpl = tpl + (num, )
print(tpl)

#List Related Experiment
lst = []
print('Enter an integer to add to the list. Enter 0 to stop.')
while True:
    num = input("Enter a number: ")
    if num == '0':
        break
    try:
        num = int(num)
    except:
        print("Invalid input")
        continue
    lst.append(num)
print(lst)

#List to tuple

lst = []
for i in range (10):
    lst.append(i)
print('List :', lst)
tpl = tuple(lst)
print('Tuple :', tpl)

#Tuple to list

tpl = ()
for i in range (10):
    tpl = tpl + (i, )
print('Tuple :', tpl)
```

```
lst = list(tpl)
print('List :', lst)
```

```
Tuple : (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
List : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Arnob Dey | ID : 203-15-3906| Section : PC - B | Subject: Artificial Intelligence Lab | Course Code: CSE316 |

```
#Array related experiments

arr = []
print('Enter 0 to stop the program')
num = int(input("Enter a number: "))
while num != 0000:
    arr.append(num)
    num = int(input("Enter a number: "))
print(arr)

#Set related experiments

s = set()
print('Enter 0 to stop the program')
num = int(input("Enter a number: "))
while num != 0:
    s.add(num)
    num = int(input("Enter a number: "))
print(s)

#Array to set conversion

arr = []
for i in range(1,11):
    arr.append(i)
print(arr)
s = set(arr)
print(s)

#Set to array conversion

s = set()
for i in range(1,11):
    s.add(i)
print(s)
arr = list(s)
print(arr)
```

Arnob Dey | ID : 203-15-3906| Section : PC - B | Subject: Artificial Intelligence Lab | Course Code: CSE316 |

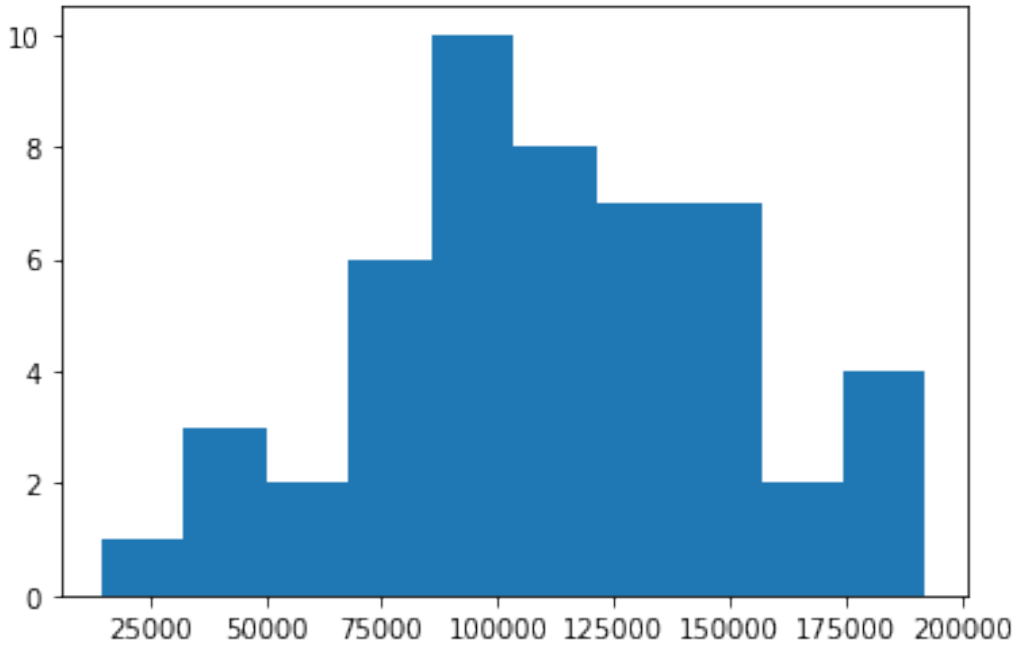
```
import numpy as np

#Arrange in numpy array
arr = np.arange(1,11)
print(arr)

#Square of the numbers in numpy array
arr2 = [x**2 for x in arr]
print(arr2)

#Cube of the numbers in numpy array
arr3 = [x**3 for x in arr]
print(arr3)

#Square Root of the numbers in numpy array
arr4 = [x**.5 for x in arr]
print(arr4)

#Reciprocal of the numbers in numpy array
arr5 = [1/x for x in arr]
print(arr5)

#Sin of the numbers numpy array
arr6 = [np.sin(x) for x in arr]
print(arr6)

#Cosin of the numbers numpy array
arr7 = [np.cos(x) for x in arr]
print(arr7)
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#Using same dataset from Aditi Dhali ma'ams class
data = pd.read_csv("https://raw.githubusercontent.com/Avik-Jain/100-Days-Of-ML-Code/master/datasets/50_Startups.csv")
data.head()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

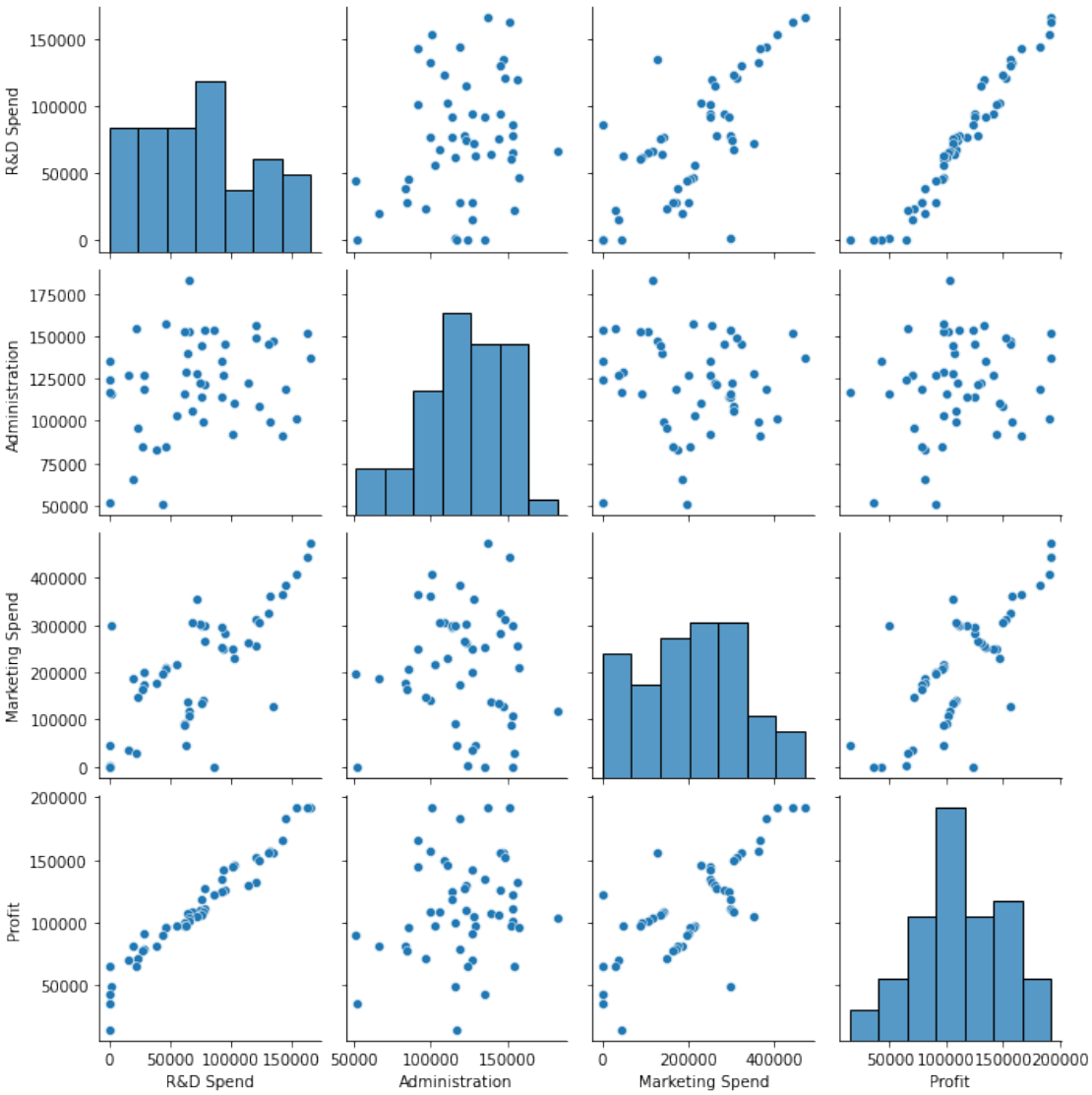
```
print(data['Profit'].head())
plt.hist(data['Profit'])
```

```
0    192261.83
1    191792.06
2    191050.39
3    182901.99
4    166187.94
Name: Profit, dtype: float64

(array([ 1.,  3.,  2.,  6., 10.,  8.,  7.,  7.,  2.,  4.]),
 array([ 14681.4  ,  32439.443,  50197.486,  67955.529,  85713.572,
        103471.615, 121229.658, 138987.701, 156745.744, 174503.787,
        192261.83 ]),
 <a list of 10 Patch objects>)
```



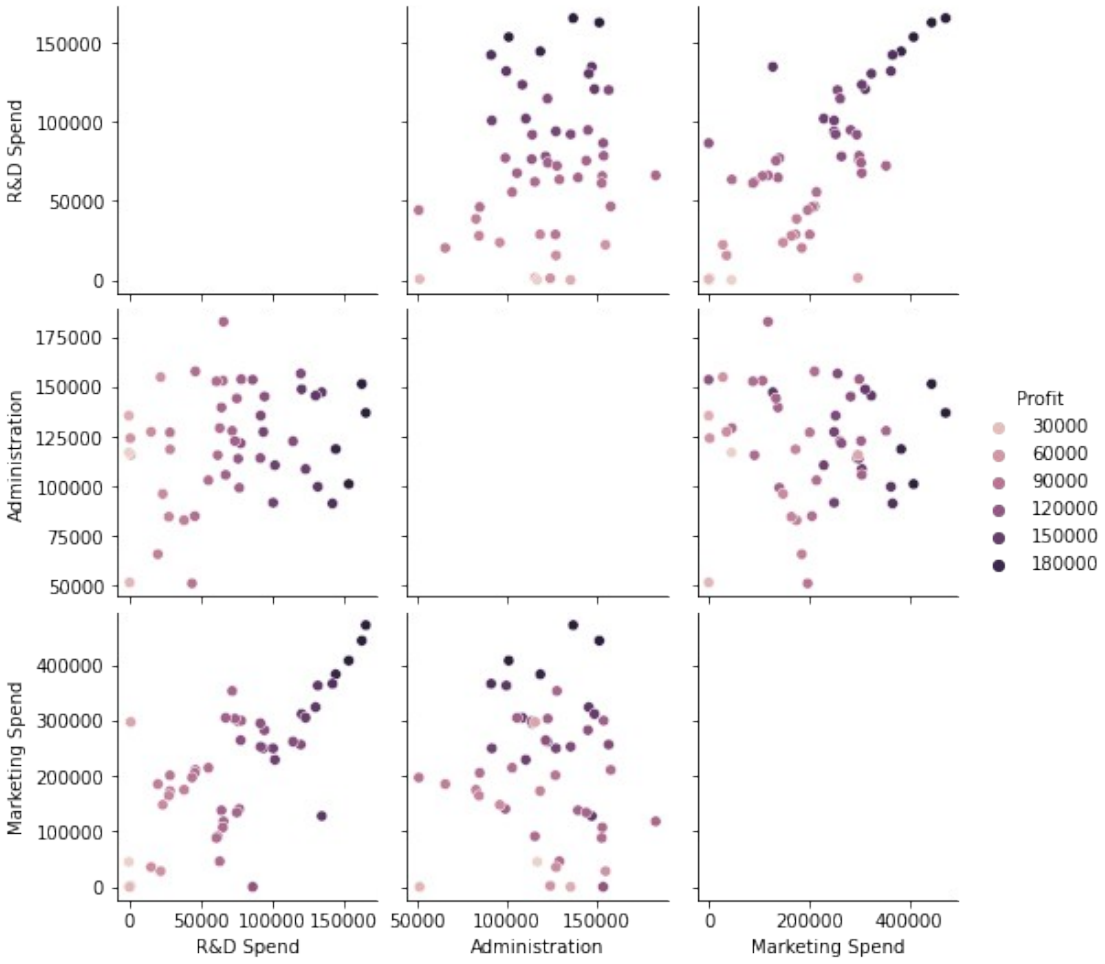

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7fbf34b735e0>
```



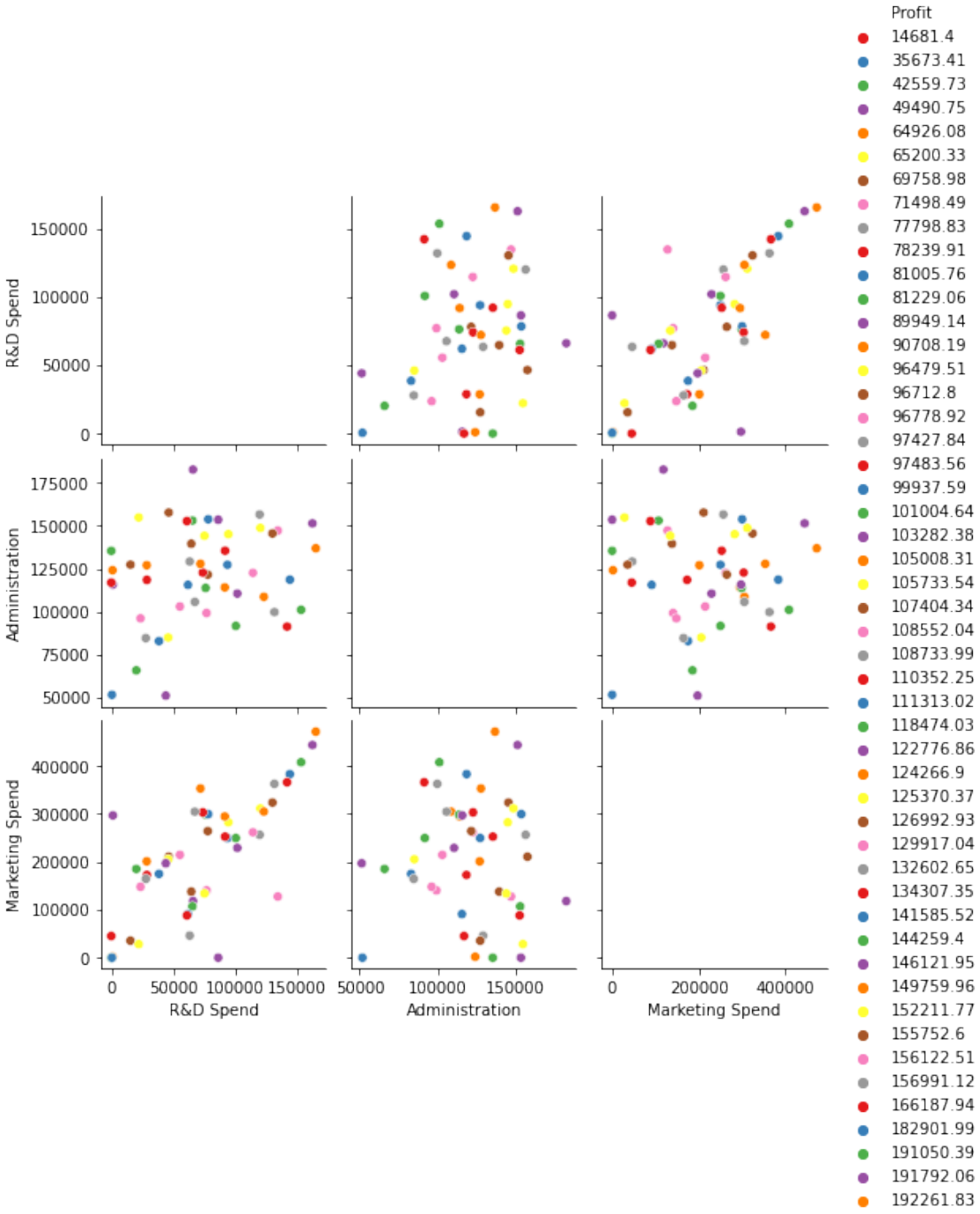

```
data.head()
sns.pairplot(data, hue="Profit")
```

```
<seaborn.axisgrid.PairGrid at 0x7fbf318e8e20>
```




```
sns.pairplot(data, hue="Profit", palette="Set1")
```

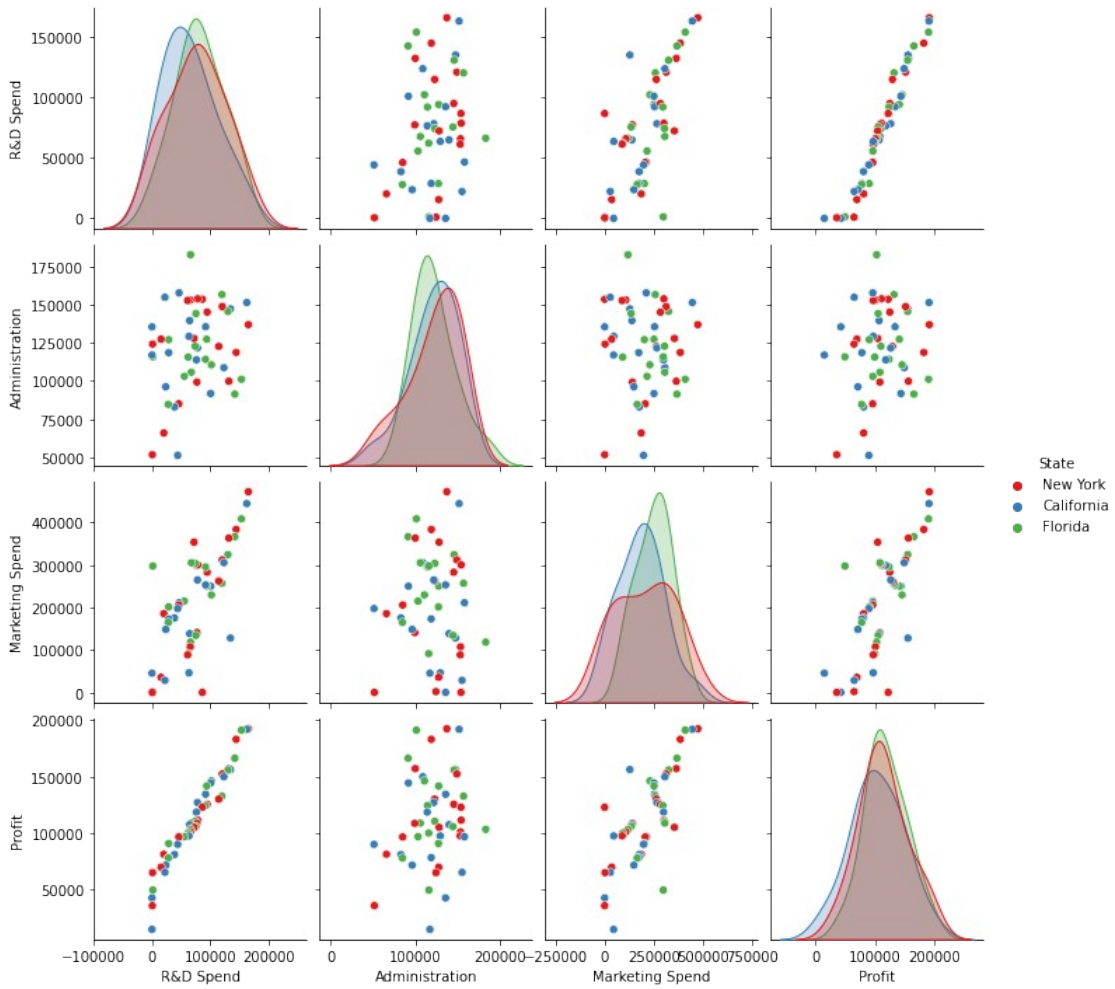
```
<seaborn.axisgrid.PairGrid at 0x7fbf318791f0>
```




```
data.head()
```

```
sns.pairplot(data, hue="State", palette="Set1", diag_kind="kde")
```

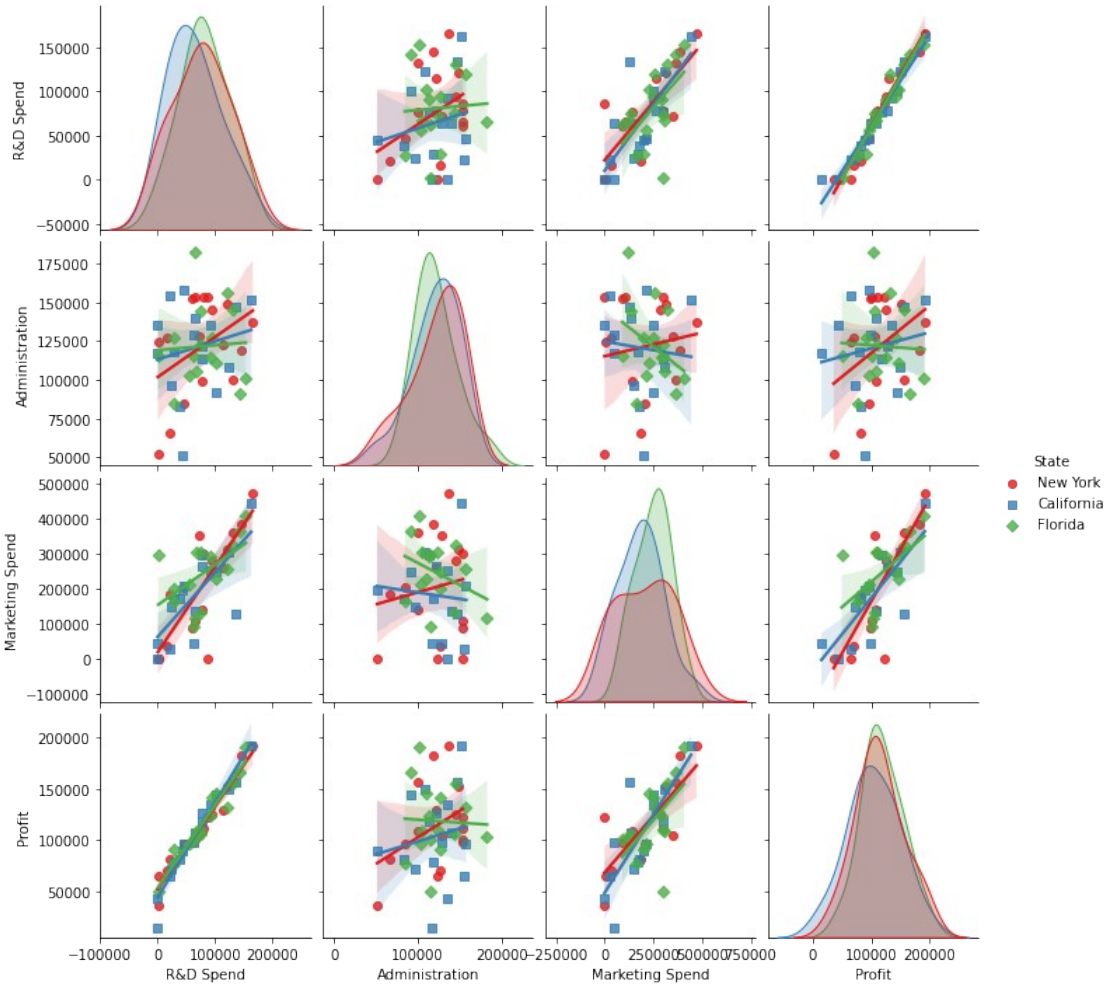
```
<seaborn.axisgrid.PairGrid at 0x7fbf318df700>
```




```
data.head()
```

```
sns.pairplot(data, hue="State", palette="Set1", diag_kind="kde",
markers=["o", "s", "D"], kind="reg")
```

```
<seaborn.axisgrid.PairGrid at 0x7fbf2cc79220>
```

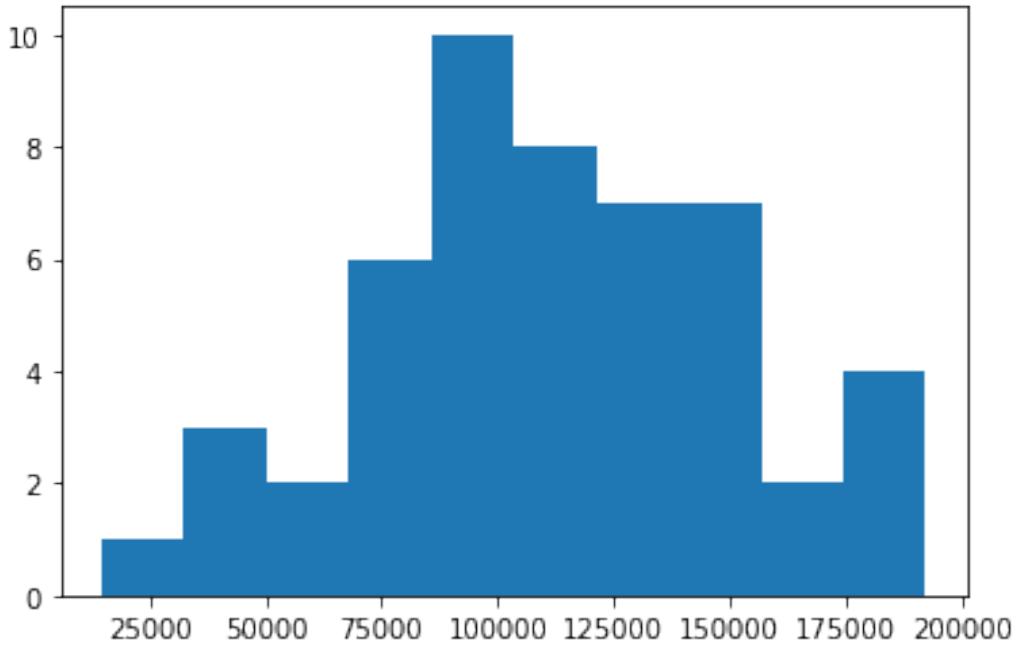

R&D Spend
Administration
Marketing Spend
Profit
State
New York
California
Florida

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
%matplotlib inline

#Using same datasetset from Aditi Dhali ma'ams class
dataset =
pd.read_csv("https://raw.githubusercontent.com/arnob016/PyClass/master
/AI%20Lab%20Report/data.csv")
dataset.head()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

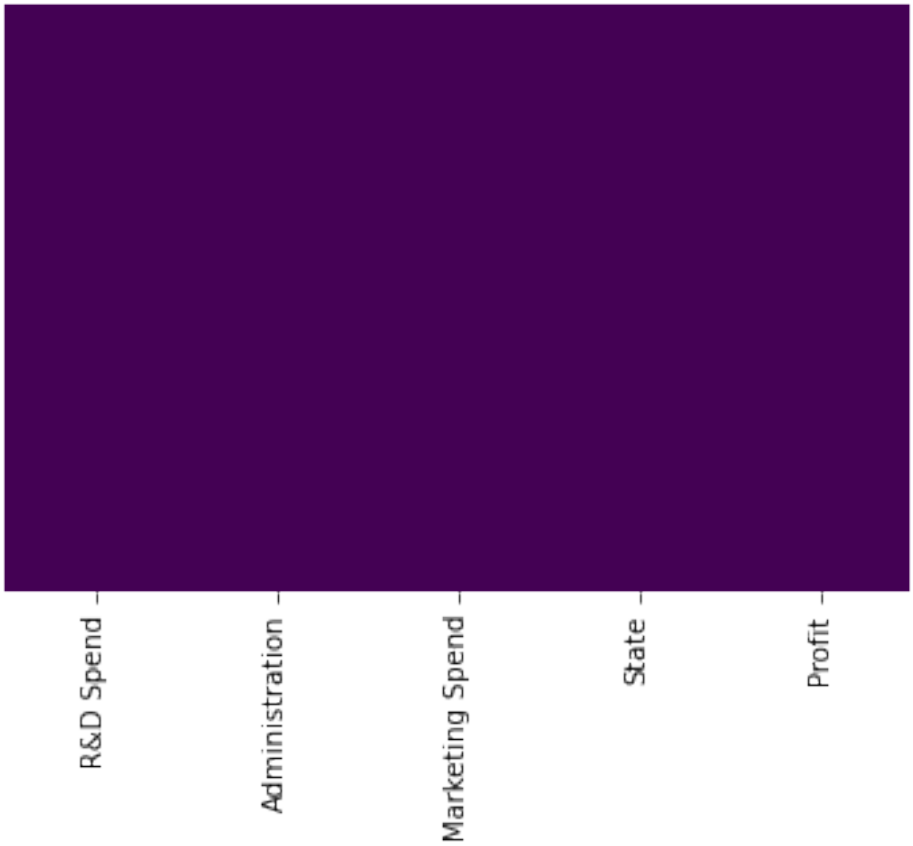
```
print(dataset['Profit'].head())
plt.hist(dataset['Profit'])
```

```
0    192261.83
1    191792.06
2    191050.39
3    182901.99
4    166187.94
Name: Profit, dtype: float64

(array([ 1.,  3.,  2.,  6., 10.,  8.,  7.,  7.,  2.,  4.]),
 array([ 14681.4  ,  32439.443,  50197.486,  67955.529,  85713.572,
        103471.615, 121229.658, 138987.701, 156745.744, 174503.787,
        192261.83 ]),
 <a list of 10 Patch objects>)
```




```
import seaborn as sns

sns.heatmap(dataset.isnull(),yticklabels = False, cbar = False, cmap =
'viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1987c2430>
```




```
nullset =
pd.read_csv("https://raw.githubusercontent.com/arnob016/PyClass/master
/AI%20Lab%20Report/nullcopy.csv")
nullset.head(10)
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	NaN	Florida	191050.39
3	144372.41	118671.85	NaN	New York	182901.99
4	142107.34	91391.77	NaN	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	NaN	NaN	NaN	California	156122.51
7	NaN	NaN	NaN	Florida	155752.60
8	NaN	NaN	NaN	New York	152211.77
9	NaN	NaN	NaN	California	149759.96

```
pd.isnull(nullset).head()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	True	False	False

```
3      False           False             True  False   False
4      False           False             True  False   False
```

```
nulldata = nullset[pd.isnull(nullset).any(axis=1)]
nulldata.head()
```

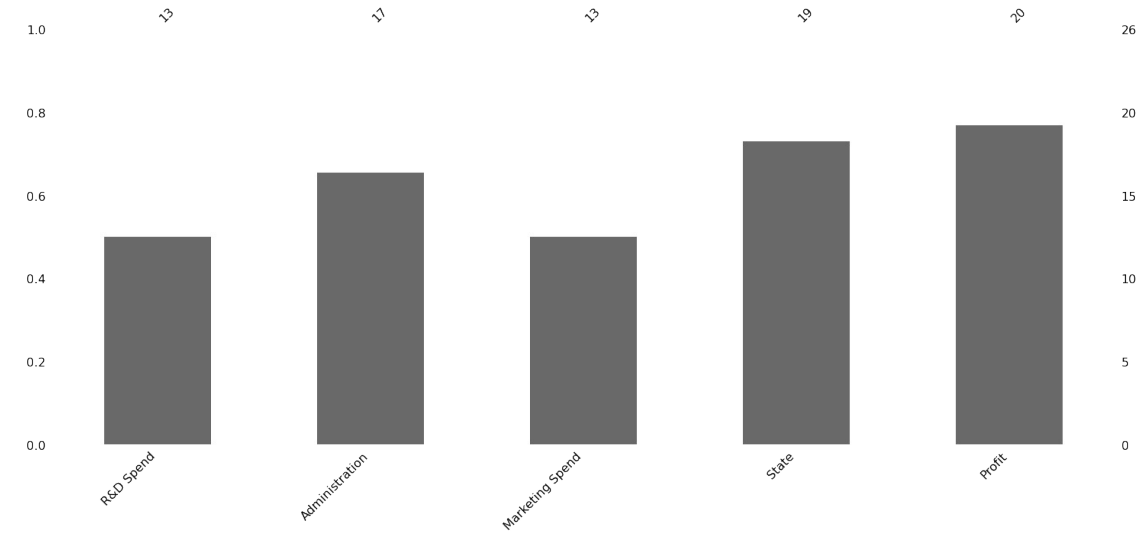
```
   R&D Spend  Administration  Marketing Spend       State     Profit
2  153441.51       101145.55              NaN     Florida  191050.39
3  144372.41       118671.85              NaN    New York  182901.99
4  142107.34        91391.77              NaN     Florida  166187.94
6        NaN             NaN              NaN  California  156122.51
7        NaN             NaN              NaN     Florida  155752.60
```

```
nulldata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26 entries, 2 to 47
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   R&D Spend        13 non-null     float64
 1   Administration   17 non-null     float64
 2   Marketing Spend  13 non-null     float64
 3   State            19 non-null     object 
 4   Profit           20 non-null     float64
dtypes: float64(4), object(1)
memory usage: 1.2+ KB
```

```
msno.bar(nulldata)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff195dcc520>
```



```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

sns.set()

df =
pd.read_csv('https://raw.githubusercontent.com/arnob016/PyClass/master
/AI%20Lab%20Report/data.csv')
df.head()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

```
df['Profit'] = df['R&D Spend'] + df['Administration'] + df['Marketing
Spend']
df.head()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	774031.10
1	162597.70	151377.59	443898.53	California	757873.82
2	153441.51	101145.55	407934.54	Florida	662521.60
3	144372.41	118671.85	383199.62	New York	646243.88
4	142107.34	91391.77	366168.42	Florida	599667.53

```
from sklearn.tree import DecisionTreeRegressor

X = df[['R&D Spend']]
y = df['Profit']

train1 = DecisionTreeRegressor(max_depth=2, random_state=1)

train1.fit(X, y)

DecisionTreeRegressor(max_depth=2, random_state=1)

sns.scatterplot(x=df['R&D Spend'],
                y=df['Profit'],
                label='data')

plt.plot(df['R&D Spend'].sort_values(),
         train1.predict(df['R&D Spend'].sort_values().to_frame()),
```

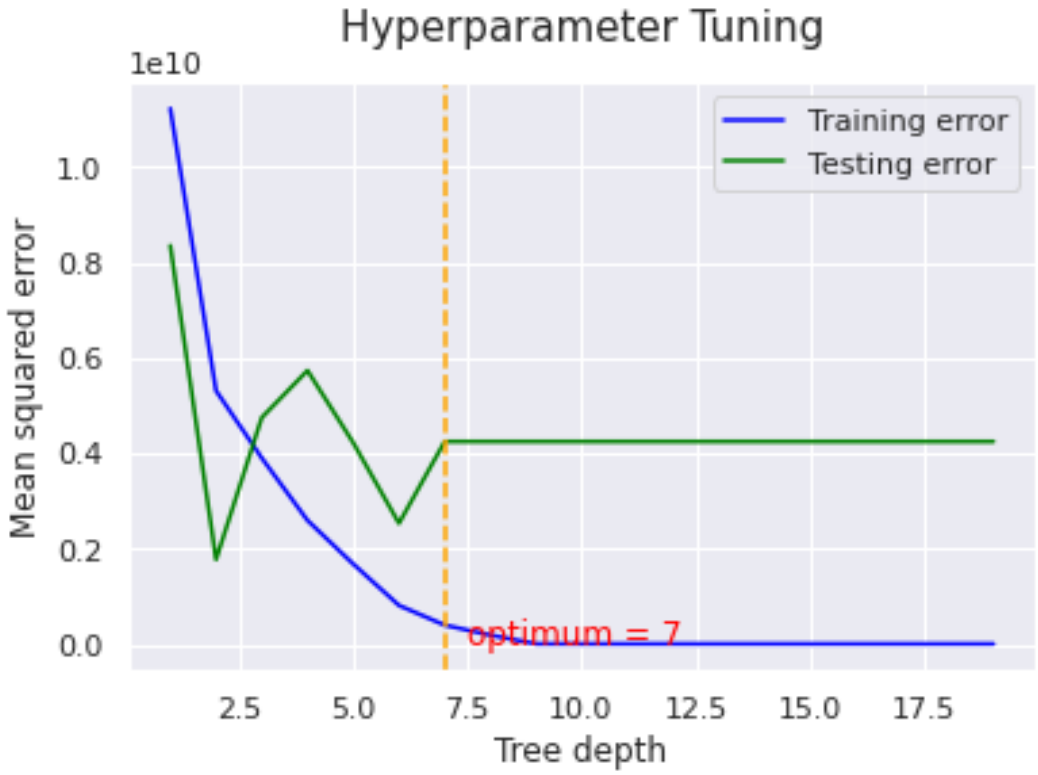
```
          color='Green', label='model',
          linewidth=2)

plt.legend()

plt.savefig('tree1.png')
```


```
from sklearn.tree import export_graphviz
import graphviz

dot_data = export_graphviz(train1, feature_names=['R&D Spend'],
filled=True, rounded=True)

graph = graphviz.Source(dot_data)
graph.render("tree")
```

```
{"type":"string"}
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.10, random_state=0, shuffle=True)

from sklearn.metrics import mean_squared_error as mse

max_depths = range(1, 20)
training_error = []
```

```
for max_depth in max_depths:
    model_1 = DecisionTreeRegressor(max_depth=max_depth)
    model_1.fit(X, y)
    training_error.append(mse(y, model_1.predict(X)))

testing_error = []
for max_depth in max_depths:
    model_2 = DecisionTreeRegressor(max_depth=max_depth)
    model_2.fit(X_train, y_train)
    testing_error.append(mse(y_test, model_2.predict(X_test)))

plt.plot(max_depths, training_error, color='blue', label='Training
error')
plt.plot(max_depths, testing_error, color='green', label='Testing
error')
plt.xlabel('Tree depth')
plt.axvline(x=7, color='orange', linestyle='--')
plt.annotate('optimum = 7', xy=(7.5, 1.17), color='red')
plt.ylabel('Mean squared error')
plt.title('Hyperparameter Tuning', pad=15, size=15)
plt.legend()
plt.savefig('error.png')
```

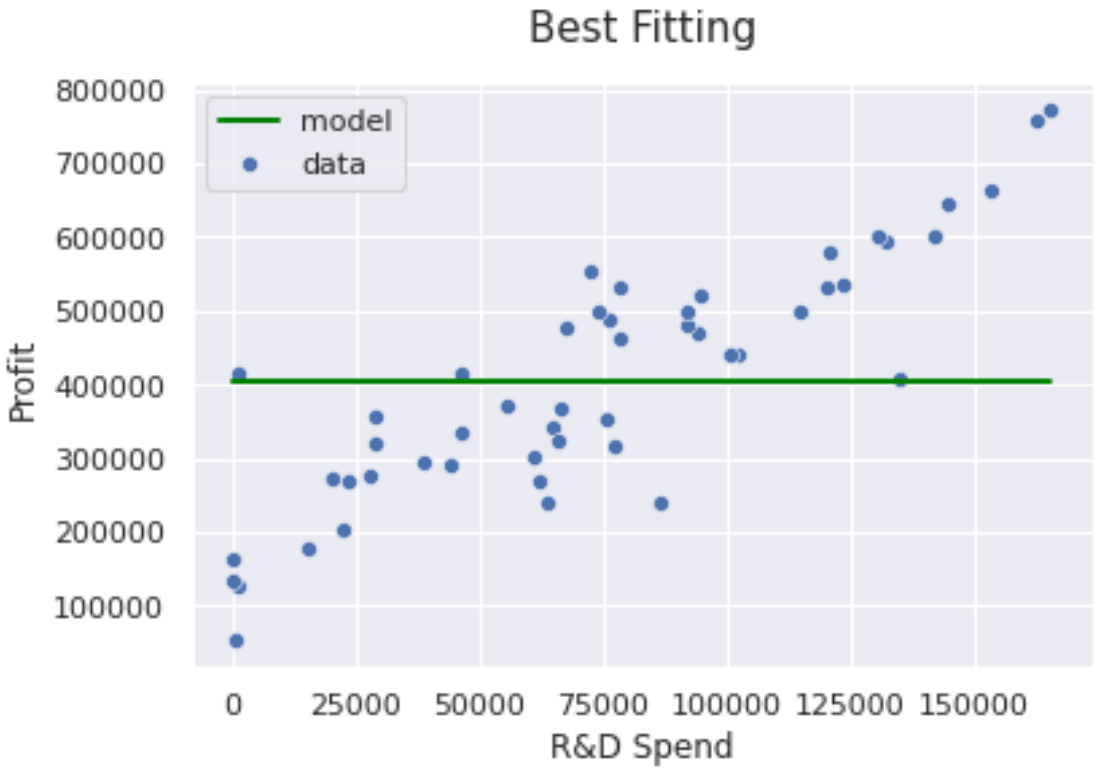


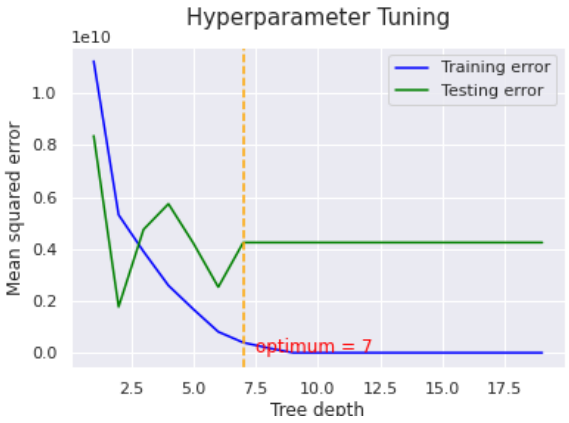

```
from sklearn.model_selection import GridSearchCV

model = DecisionTreeRegressor()
```

```
gs = GridSearchCV(model,
                  param_grid = {'max_depth': range(1, 11),
'min_samples_split': range(10, 60, 10)}, cv=5,
                  n_jobs=1, scoring='neg_mean_squared_error')

gs.fit(X_train, y_train)

print(gs.best_params_)
print(-gs.best_score_)
```

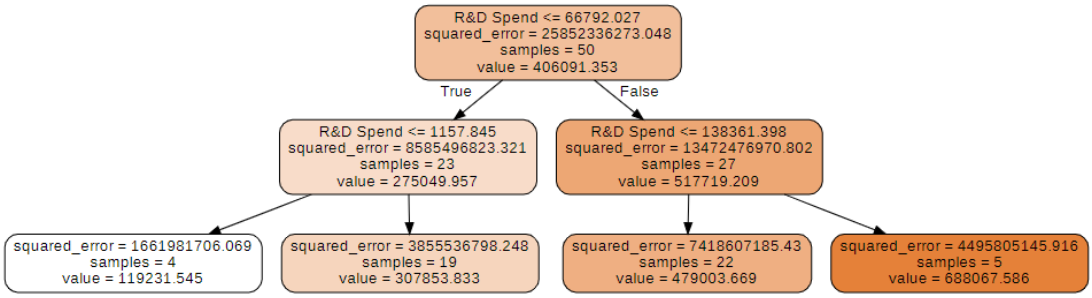
```
{'max_depth': 4, 'min_samples_split': 10}
9604135398.303028
```

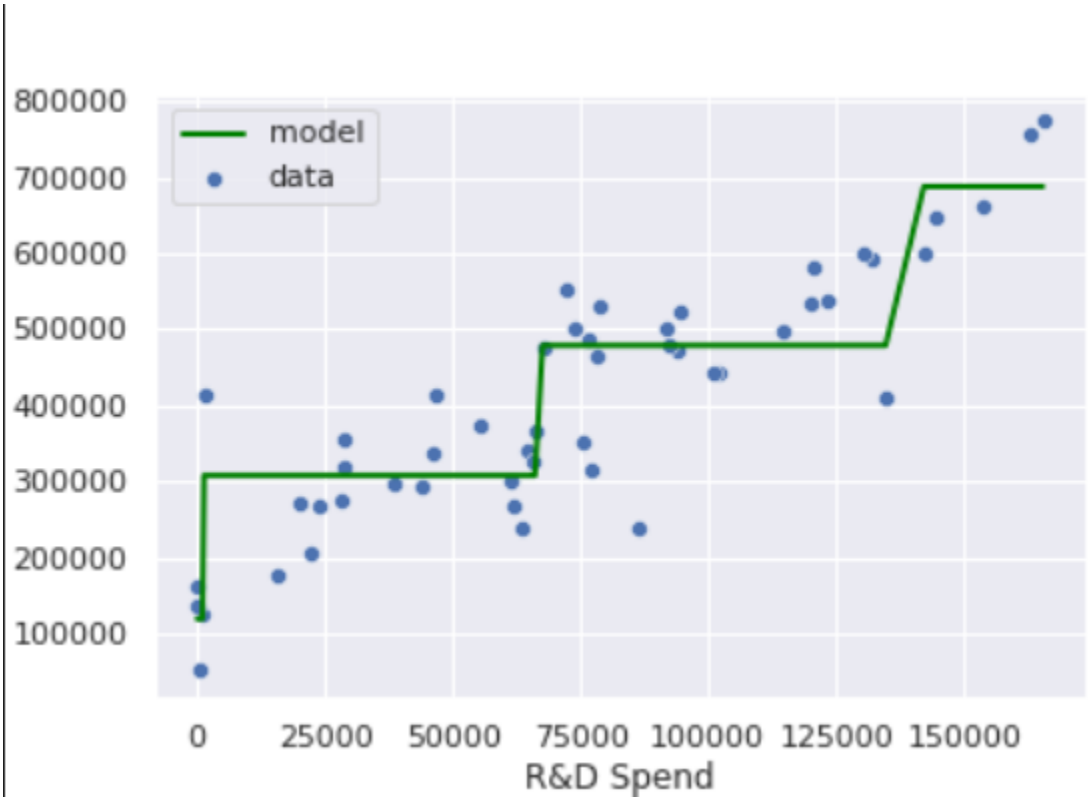
```
sns.scatterplot(x=df['R&D Spend'],
                y=df['Profit'],
                label='data')

new_model = DecisionTreeRegressor(max_depth=9,
                                  min_samples_split=50)

new_model.fit(X_train, y_train)

plt.plot(df['R&D Spend'].sort_values(),
         new_model.predict(df['R&D Spend'].sort_values().to_frame()),
         color='green', label='model', linewidth=2)

plt.legend()
plt.title('Best Fitting', pad=15, size=15)
plt.savefig('decisiontree.png')
```


Best Fitting
model
data
Profit
R&D Spend
800000
700000
600000
500000
400000
300000
200000
100000
0
25000
50000
75000
100000
125000
150000


Hyperparameter Tuning
1e10
Training error
Testing error
Mean squared error
optimum = 7
Tree depth


Best Fitting
model
data
R&D Spend


R&D Spend <= 66792.027
squared_error = 25852336273.048
samples = 50
value = 406091.353
True
False
R&D Spend <= 1157.845
squared_error = 8585496823.321
samples = 23
value = 275049.957
R&D Spend <= 138361.398
squared_error = 13472476970.802
samples = 27
value = 517719.209
squared_error = 1661981706.069
samples = 4
value = 119231.545
squared_error = 3855536798.248
samples = 19
value = 307853.833
squared_error = 7418607185.43
samples = 22
value = 479003.669
squared_error = 4495805145.916
samples = 5
value = 688067.586


model
data
R&D Spend

Arnob Dey | ID : 203-15-3906| Section : PC - B | Subject: Artificial Intelligence Lab | Course Code: CSE316 |

```
import random

while True:
    try:
        userinp = int(input("Enter the number of times to roll the
dice: "))
        break
    except ValueError:
        print("Invalid input, integer required")

dice1 = 0
dice2 = 0
dice3 = 0
dice4 = 0
dice5 = 0
dice6 = 0
for i in range(userinp):
    dice = random.randint(1,6)
    if dice == 1:
        dice1 += 1
    elif dice == 2:
        dice2 += 1
    elif dice == 3:
        dice3 += 1
    elif dice == 4:
        dice4 += 1
    elif dice == 5:
        dice5 += 1
    elif dice == 6:
        dice6 += 1
print("You rolled the dice",dice,"times")
print("You rolled 1",dice1,"times")
print("You rolled 2",dice2,"times")
print("You rolled 3",dice3,"times")
print("You rolled 4",dice4,"times")
print("You rolled 5",dice5,"times")
print("You rolled 6",dice6,"times")
```

```
You rolled the dice 5 times
The number 1 was rolled 2 times
The number 2 was rolled 1 times
The number 3 was rolled 0 times
The number 4 was rolled 1 times
The number 5 was rolled 1 times
The number 6 was rolled 0 times
```

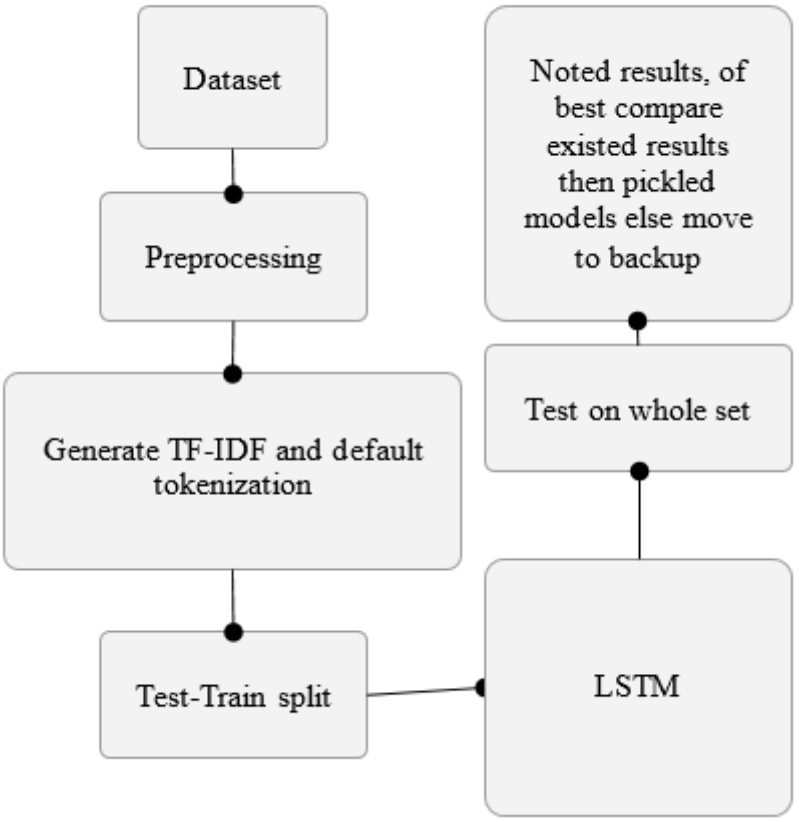
Team: | >_ SpeedOut

Members	Student ID	Section
Biplob Kumar Sutradhar	203-15-3923	PC - B
Arnob Dey	203-15-3906	PC - B
Farabi Ahnaf Akib	203-15-3920	PC - B

Objective:

Image classification is a complex procedure which relies on different components. Image classification is the task of categorizing and assigning labels to groups of pixels or vectors within an image dependent on particular rules. Meaning that, our deep learning and machine learning algorithm will be able to detect which nationality (south asian, indian, east asian , caucasian etc) the person from the image belongs to.

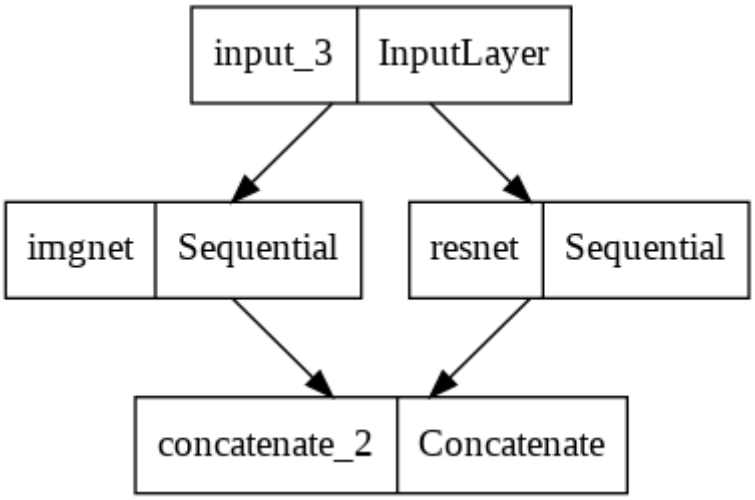
Probable Methodology:



We take an image as an input

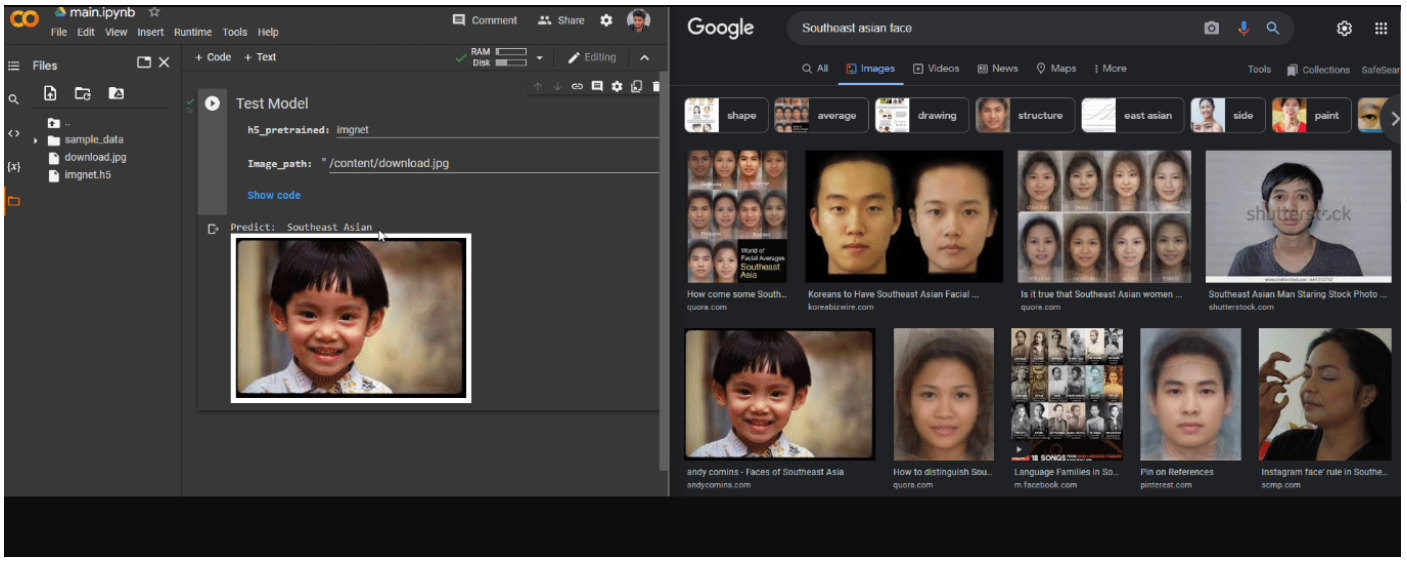
We have trained our model in a pretrained model : InceptionV3 (imagenet), Resnet(imagenet), **InceptionV3**

WE combine these models (resnet & imagenet)



5. Now we test the data on our whole dataset

6. Based on the result, we compare existing results and keep the best result, delete all others and clear our cache.



Novelty of Image classifier:

1. We can use our image classifier model in a visual search engine
2. We can use our model to increase security in infrastructures.
3. In visual learning & visual listening
4. In image recognition
5. We can use it to guess nationality of a deceased person
6. To detect criminal identity