ARTIFICIAL INTELLIGENCE

EXPERIMENT NO: 9

IMPLEMENTATION OF UNCERTAIN METHODS FOR AN APPLICATION

Nikith Kumar Seemakurthi

RA1911003020480

AIM:

To implement the uncertain methods for an application using python.

ALGORITHM:

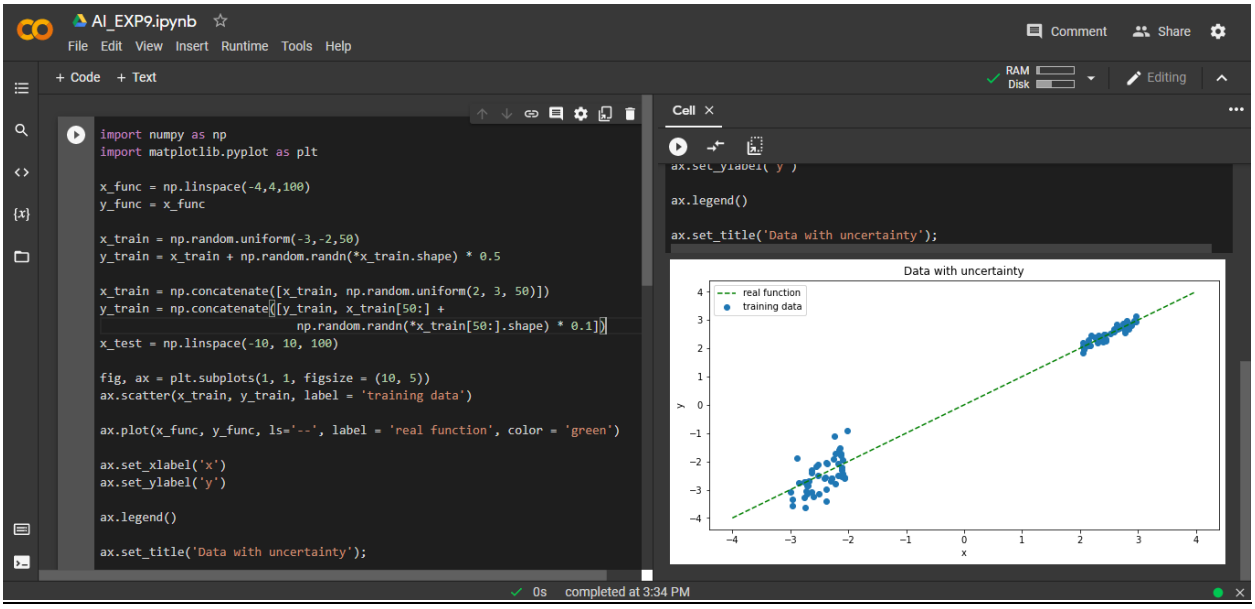
1. Import the packages numpy and matplotlib.
2. Two ndarray objects x_func and y_func are created to store the values of x-axis and y-axis respectively.
3. The values on x-axis and y-axis are ranged from -4to4 with num as 100.
4. Initialize the value of x_train and y_train using random.uniform() and random.randn() respectively.
5. The values of x_train and y_train are updated by concatenating the new values.
6. The values on x_test are ranged between -10 to 10 with num as 100.
7. By using subplot() the layout of the figure is set to 1, 1 and figure size as 10, 5.
8. BY using scatter() the values of x_train and y_train are inserted into the graph with the label training data.
9. By using plot() the values of x_func and y_func are plotted inside the graph with line style as dash lines and labeled as real function.
10. The real function and training data are represented in top left corner by using legend().
11. The x-axis and y-axis are labeled as x and y respectively and the figure is titled as Data with uncertainty.

SOURCE CODE:

```
import numpy as np

import matplotlib.pyplot as plt


x_func = np.linspace(-4,4,100)

y_func = x_func


x_train = np.random.uniform(-3,-2,50)

y_train = x_train + np.random.randn(*x_train.shape) * 0.5

x_train = np.concatenate([x_train, np.random.uniform(2, 3, 50)])
```

```
y_train = np.concatenate([y_train, x_train[50:] + np.random.randn(*x_train[50:].shape) * 0.1])

x_test = np.linspace(-10, 10, 100)

fig, ax = plt.subplots(1, 1, figsize = (10, 5))
ax.scatter(x_train, y_train, label = 'training data')

ax.plot(x_func, y_func, ls='--', label = 'real function', color = 'green')

ax.set_xlabel('x')
ax.set_ylabel('y')
ax.legend()

ax.set_title('Data with uncertainty')
```

OUTPUT:



RESULT:

The Uncertain methods for an application using python has been implemented.