AIM: Implementation of C4.5

THEORY:

The C4.5 algorithm is used in Data Mining as a Decision Tree Classifier which can be employed to generate a decision, based on a certain sample of data (univariate or multivariate predictors). C4.5 is the successor to ID3 and removed the restriction that features must be categorical by dynamically defining a discrete attribute (based on numerical variables) that partitions the continuous attribute value into a discrete set of intervals. C4.5 converts the trained trees (i.e. the output of the ID3 algorithm) into sets of if-then rules. This accuracy of each rule is then evaluated to determine the order in which they should be applied. Pruning is done by removing a rule's precondition if the accuracy of the rule improves without it.

SOURCE CODE:

1) INSTALLING CHEFBOOST FOR C4.5:

```
!pip install chefboost
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting chefboost
  Downloading chefboost-0.0.17-py3-none-any.whl (26 kB)
Requirement already satisfied: pandas>=0.22.0 in /usr/local/lib/python3.7/dist-packages (from chefboost) (1.3.5)
Requirement already satisfied: numpy>=1.14.0 in /usr/local/lib/python3.7/dist-packages (from chefboost) (1.21.6)
Requirement already satisfied: psutil>=5.4.3 in /usr/local/lib/python3.7/dist-packages (from chefboost) (5.4.8)
Requirement already satisfied: tqdm>=4.30.0 in /usr/local/lib/python3.7/dist-packages (from chefboost) (4.64.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.22.0->chefboost
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.22.0->chefboost) (2022.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>=0.22.0-
Installing collected packages: chefboost
Successfully installed chefboost-0.0.17
```

2) IMPORTING LIBRARIES AND READING DATA:

```
import pandas as pd
data = pd.read_csv('https://raw.githubusercontent.com/serengil/chefboost/master/tests/dataset/golf.txt')
data
```

	Outlook	Temp.	Humidity	Wind	Decision
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes

3) **BUILDING MODEL:**

```
from chefboost import Chefboost as chef
config = {'algorithm': 'C4.5'}
model = chef.fit(data, config = config, target_label = 'Decision')
```

```
[INFO]:  1 CPU cores will be allocated in parallel running
C4.5  tree is going to be built...
-------------------------
finished in  0.5118496417999268  seconds
-------------------------
Evaluate  train set
-------------------------
Accuracy:  100.0 % on  14  instances
Labels:  ['No' 'Yes']
Confusion matrix:  [[5, 0], [0, 9]]
Precision:  100.0 %, Recall:  100.0 %, F1:  100.0 %
```

4) **PREDICTING VALUES:**

```
for i in range(data.shape[0]):
  prediction = chef.predict(model, param = data.iloc[i])
  print(prediction)
```

```
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
Yes
No
```

CONCLUSUION:

From this practical, I have learned the implementation of C4.5 in python.