Practical 5

Aim: To implement Decision-Tree for restaurant wait problem.

Code:

*import* pandas *as* pd

*import* numpy *as* пр

*from* sklearn.preprocessing *import* LabelEncoder

*from* sklearn.feature\_extraction *import* DictVectorizer

*from* sklearn *import* tree

*import* warnings

warnings.filterwarnings(action='ignore',category=DeprecationWarning)

data=pd.read\_csv('data.csv')

cols\_to\_retain=['Alt','Bar','Fri','Hun','Pat','Price','Rain','Res','Type','Est']

X\_feature=data[cols\_to\_retain]

X\_dict=X\_feature.T.to\_dict().values()

vect=DictVectorizer(sparse=False)

X\_vector=vect.fit\_transform(X\_dict)

print(X\_vector)

X\_Train=X\_vector[:-1]

X\_Test=X\_vector[-1:]

print('Train Set')

print(X\_Train)

print('test set')

print(X\_Test)

le=LabelEncoder()

y\_Train=le.fit\_transform(data['Goal'][:-1])

clf=tree.DecisionTreeClassifier(criterion='entropy')

clf=clf.fit(X\_Train,y\_Train)

print('test data')

print(le.inverse\_transform(clf.predict(X\_Test)))

Output:

