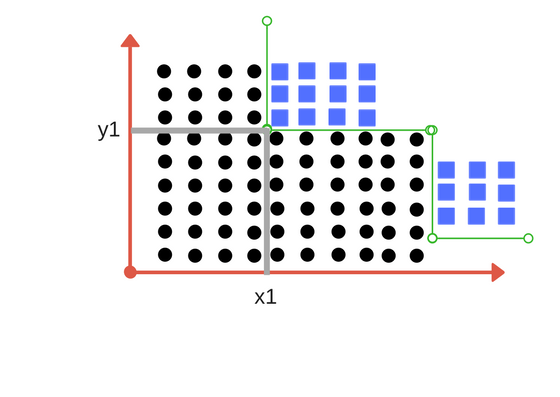
Decision Tree Classification

**Short note about Decision Tree Classification** :

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

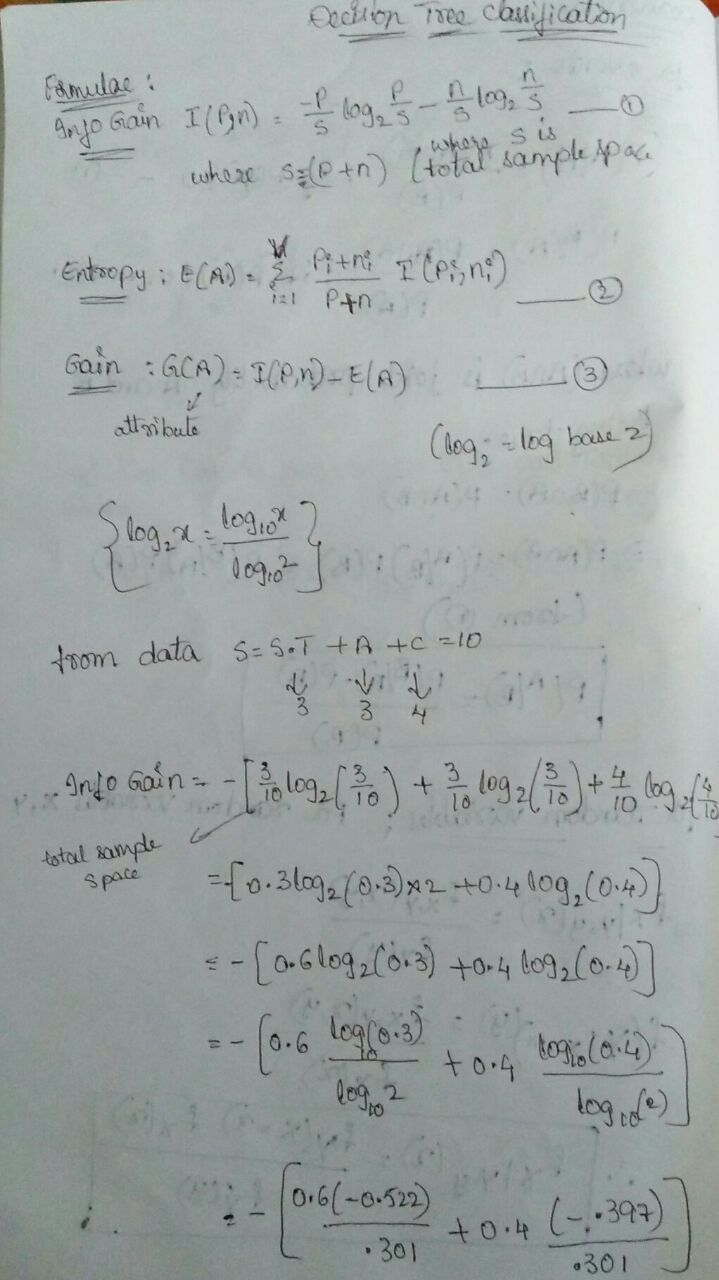
1. IN GRAPH :

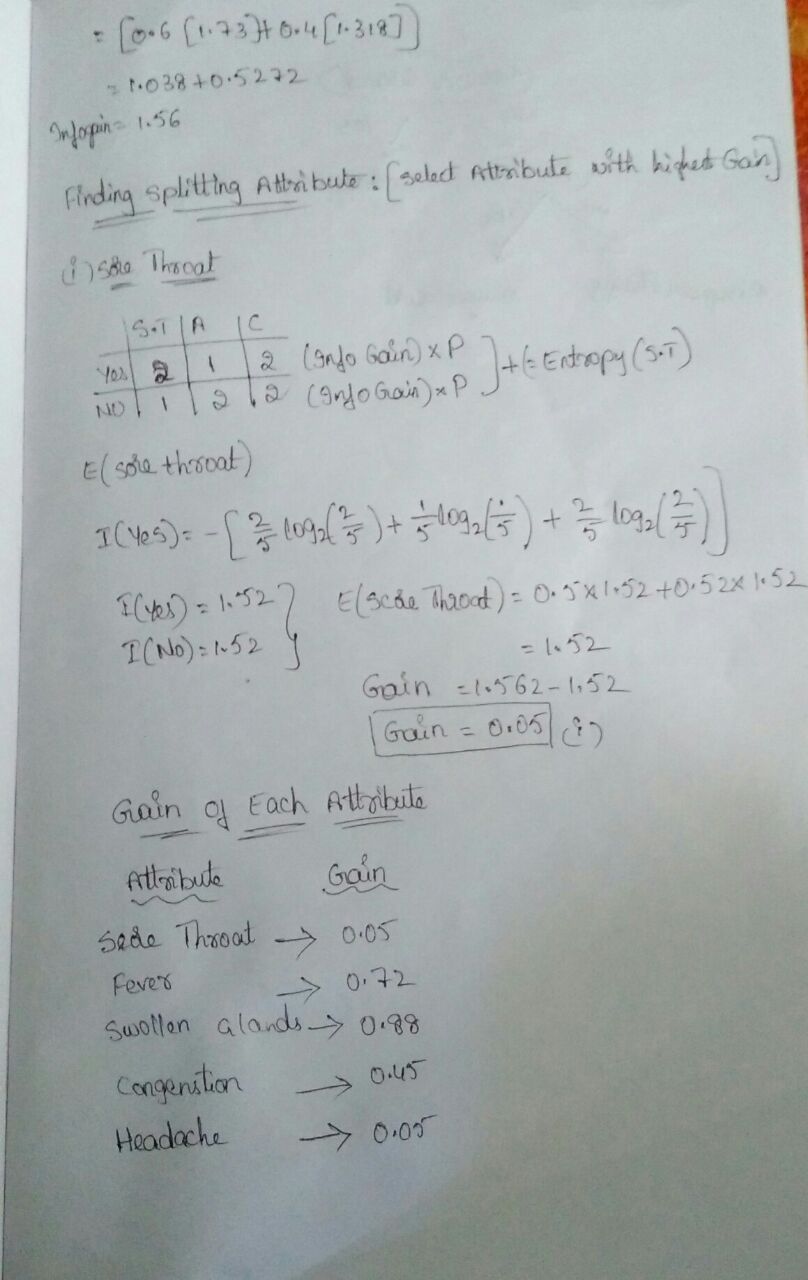


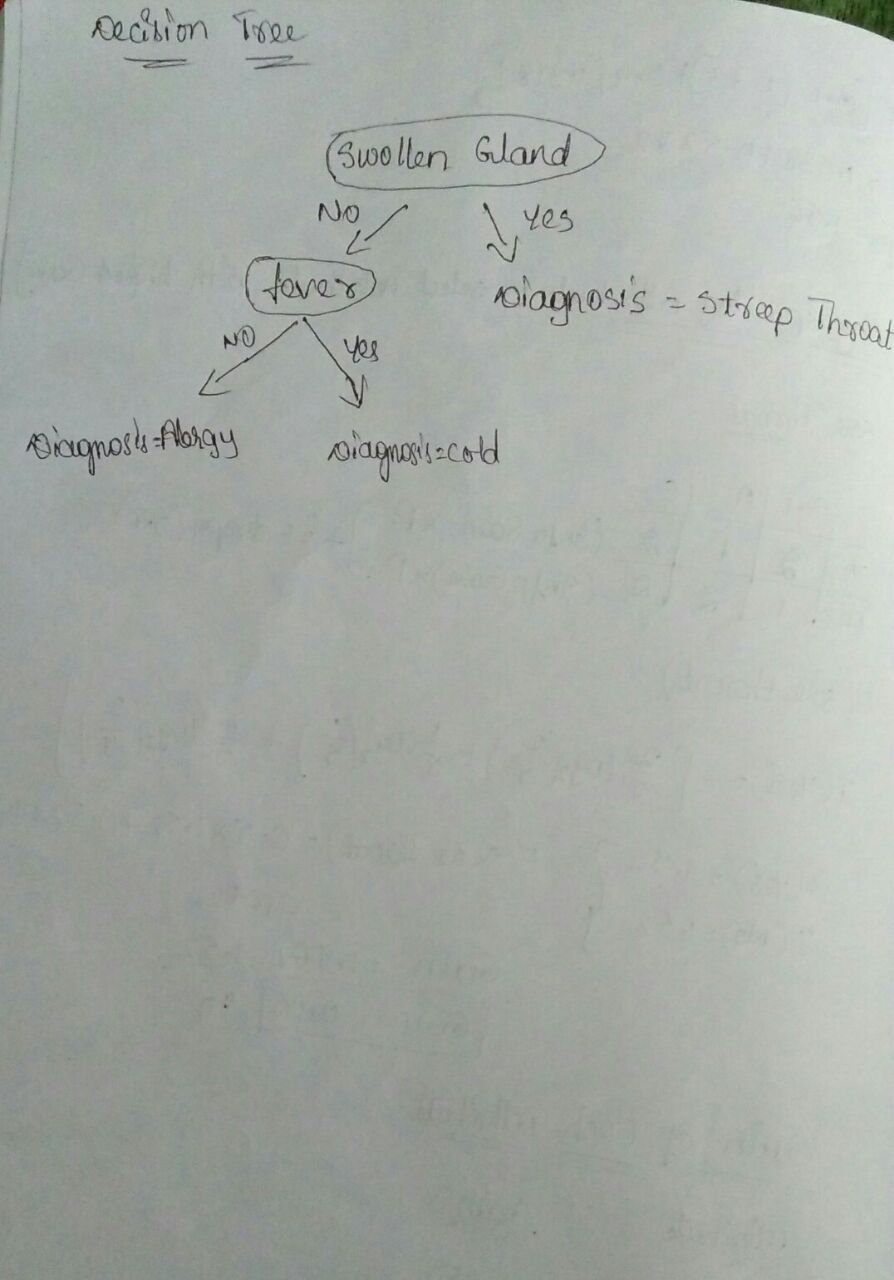
**Complex Problem-**

For the following Medical Diagnosis Data,Create Decision Tree

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SoreThreat | Fever | SwollenGlands | Congestion | HeadAche | Diagnosis |
| Yes | Yes | Yes | Yes | Yes | StrepThroat |
| No | No | No | Yes | Yes | Allergy |
| Yes | Yes | No | Yes | No | Cold |
| Yes | No | Yes | No | No | StrepThroat |
| No | Yes | No | Yes | No | Cold |
| No | No | No | Yes | No | Allergy |
| No | No | Yes | No | No | StrepThroat |
| Yes | No | No | No | yes | Allergy |
| No | Yes | No | Yes | yes | Cold |
| Yes | No | No | Yes | yes | Cold |

****

****

****

**CaseStudy-**

1. Finding the difference between approval of loan among Emp maximum and minimum salary.
2. Finding obese person with adiposity prone to heart disease
3. Finding Product Price & Sales
4. Finding Age & Mortality
5. Finding temperature vs. Number of cones sold at ice cream store
6. Finding Population vs Food consumption
7. Finding quantity with yield
8. Determining the chances to win cricket match .
9. Determining the chances of getting Jobs after Completing Graduation.
10. Speed and distance relationship
11. Finding rate of growth of the economy of a Institution

Application:

* **Agriculture:** Application of a range of machine learning methods to problems in agriculture and horticulture is described in.
* **Astronomy:** Astronomy has been an active domain for using automated classification techniques. Use of decision trees for filtering noise from Hubble Space Telescope images was reported recently in. Decision trees have helped in star-galaxy classification, determining galaxy counts and discovering quasars in the Second Palomar Sky Survey. Use of neural trees for ultraviolet stellar spectral classification is described in
* **Biomedical Engineering:** Use of decision trees for identifying features to be used in implantable devices can be found in
* **Control Systems:** Automatic induction of decision trees was recently used for control of nonlinear dynamical systems
* **Financial analysis:** Use of CART for asserting the attractiveness of buy-writes is reported in.
* **Medicine:** Medical research and practice have long been important areas of application for decision tree techniques. Recent uses of automatic induction of decision trees can be found in diagnosis , cardiology, psychiatry , gastroenterology , for detecting microcalcifications in mammography , to analyze Sudden Infant Death (SID) syndrome and for diagnosing thyroid disorders .
* **Molecular biology:** Initiatives such as the Human Genome Project and the GenBank database offer fascinating opportunities for machine learning and other data exploration methods in molecular biology.
* **Object recognition:** Tree based classification has been used recently for recognizing three dimensional objects and for high level vision .

**Python**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

## Importing the dataset

dataset = pd.read\_csv('C:\\Users\\Rama\\Desktop\\Social\_Network\_Ads.csv')

X = dataset.iloc[:, [2, 3]].values

y = dataset.iloc[:, 4].values

#

# Splitting the dataset into the Training set and Test set

from sklearn.cross\_validation import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25)

#

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

#fitting classifier to training set

#fitting classifier to training set

from sklearn.tree import DecisionTreeClassifier

classifier=DecisionTreeClassifier()

classifier=DecisionTreeClassifier(criterion='entropy',random\_state=0)

classifier.fit=(y\_test,y\_pred)

#

#

# Fitting K-NN Regression to the Training set

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors = 5, metric = 'minkowski', p = 2)

classifier.fit(X\_train, y\_train)

#

# Predicting the Test set results

from sklearn.linear\_model import LinearRegression

y\_pred = classifier.predict(X\_test)

#

# Making the Confusion Matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

#

# Visualising the Training set results

from matplotlib.colors import ListedColormap

X\_set, y\_set = X\_train, y\_train

X1, X2 = np.meshgrid(np.arange(start = X\_set[:, 0].min() - 1, stop = X\_set[:, 0].max() + 1, step = 0.01),

np.arange(start = X\_set[:, 1].min() - 1, stop = X\_set[:, 1].max() + 1, step = 0.01))

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(X1.min(), X1.max())

plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(X\_set[y\_set == j, 0], X\_set[y\_set == j, 1],

c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('DecisionTree Classification (Training set)')

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.legend()

plt.show()

#

# Visualising the Test set results

from matplotlib.colors import ListedColormap

X\_set, y\_set = X\_test, y\_test

X1, X2 = np.meshgrid(np.arange(start = X\_set[:, 0].min() - 1, stop = X\_set[:, 0].max() + 1, step = 0.01),

np.arange(start = X\_set[:, 1].min() - 1, stop = X\_set[:, 1].max() + 1, step = 0.01))

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(X1.min(), X1.max())

plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(X\_set[y\_set == j, 0], X\_set[y\_set == j, 1],

c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('Decision tree classification (Test set)')

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.legend()

plt.show()

