In [1]:

```
# 1.1
age = ["young","young","young","young","young","prepresbyopic","prepresbyopic",
       "prepresbyopic","prepresbyopic","prepresbyopic","presbyopic",
       "presbyopic","presbyopic","presbyopic","presbyopic"]
spec = ["near","near","near","near","far","near","near","near","near","far",
        "near","far","far","far","far"]
astig = ["no","no","yes","yes","no","no","no","yes","yes","no","yes","no",
         "no","yes","yes"]
tear = ["reduced","normal","reduced","normal","reduced","reduced","normal",
        "reduced","normal","reduced","normal","reduced","normal","reduced",
        "normal"]
lenses = ["no contact lenses","soft contact lenses","no contact lenses",
          "hard contact lenses","no contact lenses","no contact lenses",
          "soft contact lenses","no contact lenses","hard contact lenses",
          "no contact lenses","hard contact lenses","no contact lenses",
          "soft contact lenses","no contact lenses","no contact lenses"]
```

In [2]:

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
```

In [3]:

```
age_encoded = le.fit_transform(age)
print("age : ", age_encoded)
```

```
age :  [2 2 2 2 2 0 0 0 0 0 1 1 1 1 1]
```

In [4]:

```
spec_encoded = le.fit_transform(spec)
print("spec : ", spec_encoded)
```

```
spec :  [1 1 1 1 0 1 1 1 1 0 1 0 0 0 0]
```

In [5]:

```
astig_encoded = le.fit_transform(astig)
print("astig : ", astig_encoded)
```

```
astig :  [0 0 1 1 0 0 0 1 1 0 1 0 0 1 1]
```

In [6]:

```
tear_encoded = le.fit_transform(tear)
print("tear : ", tear_encoded)
```

```
tear :  [1 0 1 0 1 1 0 1 0 1 0 1 0 1 0]
```

In [7]:

```
lenses_encoded = le.fit_transform(lenses)
print("lenses : ", lenses_encoded)
```

```
lenses :  [1 2 1 0 1 1 2 1 0 1 0 1 2 1 1]
```

In [8]:

```
feature = list(zip(age_encoded, spec_encoded, astig_encoded, tear_encoded))
print(feature)
```

```
[(2, 1, 0, 1), (2, 1, 0, 0), (2, 1, 1, 1), (2, 1, 1, 0), (2, 0, 0, 1), (0, 1,
0, 1), (0, 1, 0, 0), (0, 1, 1, 1), (0, 1, 1, 0), (0, 0, 0, 1), (1, 1, 1, 0),
(1, 0, 0, 1), (1, 0, 0, 0), (1, 0, 1, 1), (1, 0, 1, 0)]
```

In [9]:
```
x = feature
label = lenses_encoded
y = label
```

In [10]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
                                                    random_state = 1)
```

In [11]:
```
print(x_train)
```

```
[(2, 0, 0, 1), (2, 1, 0, 0), (1, 0, 1, 1), (2, 1, 0, 1), (1, 0, 1, 0), (0, 0,
0, 1), (0, 1, 1, 0), (1, 0, 0, 0), (1, 0, 0, 1), (0, 1, 0, 1)]
```

In [12]:
```
print(x_test)
```

```
[(2, 1, 1, 0), (0, 1, 1, 1), (0, 1, 0, 0), (2, 1, 1, 1), (1, 1, 1, 0)]
```

In [13]:
```
print(y_train)
```

```
[1 2 1 1 1 1 0 2 1 1]
```

In [14]:
```
print(y_test)
```

```
[0 1 2 1 0]
```

In [15]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model = model.fit(x_train, y_train)
print(model)
```

```
GaussianNB()
```

In [16]:
```
from sklearn import metrics
y_pred = model.predict(x_test)
y_score = metrics.accuracy_score(y_test, y_pred)
```

In [17]:
```
print(y_pred)
```

```
[1 1 2 1 1]
```

In [18]:
```
print(y_score)
```

```
0.6
```

In [19]:
```
# 1.2
predicted = model.predict([[0, 0, 1, 1]])
print(predicted)
```

```
[1]
```