# Project 5

**1)** Group Eta: Roman Formicola

**2) a)**

$P(class = 1) = 1/4$
$P(class = 2) = 1/2$
$P(class = 3) = 1/4$
$|V| = 14$
$\hat{P}(t|c) = (N_{ct} + 1)/(N_c + 2)$

**i)** $P(X_{peony} = T|class = 2) = 3/4$

**ii)** $P(X_{crocus} = T|class = 2) = 1/2$

**iii)** $P(X_{peony} = T|class = 1) = 2/3$

**b)**

$P(t|c) = (count(t, c) + 1)/(count(t) + |v|)$

**i)** $P(X = peony|class = 2) = (4 + 1)/(14 + 14) = 5/28$

**ii)** $P(X = crocus|class = 2) = (1 + 1)/(14 + 14) = 2/28 = 1/14$

**iii)** $P(X = peony|class = 1) = (1 + 1)/(8 + 14) = 2/22 = 1/11$

**c)**

$P(d|class = 1) = P(1) * P(X_{daffodil} = T|class = 1) * P(X_{crocus} = T|class = 1) * P(X_{daisy} = T|class = 1)$
$* P(X_{tulip} = T|class = 1)$
$* P(X_{clematis} = T|class = 1) * P(X_{peony} = T|class = 1) = (1/4) * (1/3) * (1/3) * (1/3) * (2/3) * (2/3)$
$* (2/3) \approx 0.002743$

$P(d|class = 2) = P(2) * P(X_{daffodil} = T|class = 2) * P(X_{crocus} = T|class = 2) * P(X_{daisy} = T|class = 2)$
$* P(X_{tulip} = T|class = 2)$
$* P(X_{clematis} = T|class = 2) * P(X_{peony} = T|class = 2) = (1/2) * (1/2) * (1/2) * (1/4) * (1/4) * (3/4)$
$* (3/4) \approx 0.0043945$

$P(d|class = 3) = P(3) * P(X_{daffodil} = T|class = 3) * P(X_{crocus} = T|class = 3) * P(X_{daisy} = T|class = 3)$
$* P(X_{tulip} = T|class = 3)$
$* P(X_{clematis} = T|class = 3) * P(X_{peony} = T|class = 3) = (1/4) * (1/3) * (1/3) * (2/3) * (2/3) * (1/3)$
$* (1/3) \approx 0.00137174$

Predicted class for document: daffodil crocus daisy tulip clematis peony = 2

**d)**

$P(d|class = 1) = P(X = daffodil|class = 1) * P(X = crocus|class = 1) * P(X = daisy|class = 1)$
$* P(X = tulip|class = 1) * P(X = clemantis|class = 1) * P(X = peony|class = 1) = (1/4) * (1/22)$
$* (1/22) * (1/22) * (1/11) * (1/11) * (1/11) \approx 1.76398 * 10^{-8}$

$P(d|class = 2) = P(X = daffodil|class = 2) * P(X = crocus|class = 2) * P(X = daisy|class = 2)$
$* P(X = tulip|class = 2) * P(X = clemantis|class = 2) * P(X = peony|class = 2) = (1/2) * (1/14)$
$* (1/14) * (1/28) * (1/28) * (5/28) * (5/28) \approx 1.03758 * 10^{-7}$

$P(d|class = 3) = P(X = daffodil|class = 3) * P(X = crocus|class = 3) * P(X = daisy|class = 3)$
$* P(X = tulip|class = 3) * P(X = clemantis|class = 3) * P(X = peony|class = 3) = (1/4) * (1/21)$
$* (1/21) * (2/21) * (3/21) * (1/21) * (1/21) \approx 1.74894 * 10^{-8}$

Predicted class for document: daffodil crocus daisy tulip clematis peony = 2

**3)**

**a)**

```
In [6]: import pandas as pd

        term_doc_matrix = [[1, 1, 1, 1, 0, 0, 0], [0, 1, 1, 0, 1, 1, 0], [1, 0, 1, 1, 1, 1, 1]]
        df = pd.DataFrame(term_doc_matrix, columns=["cat", "bat", "rat", "fat", "mat", "pat", "sat"])
        df.style.set_caption("Term Document Matrix")
```

Out[6]:

Term Document Matrix

| | cat | bat | rat | fat | mat | pat | sat |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| **1** | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| **2** | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

**b)** Note: I used $Log_{10}(1 + tf_{t,d}) * Log_{10}(N/df_t)$ for TF-IDF weights

```
In [8]: TF_IDF = [[0.106, 0.053, 0, 0.053, 0, 0, 0], [0, 0.106, 0, 0, 0.053, 0.053, 0],
                  [0.053, 0, 0, 0.053, 0.053, 0.053, 0.1436]]
        df = df = pd.DataFrame(TF_IDF, columns=["cat", "bat", "rat", "fat", "mat", "pat", "sat"])
        df.style.set_caption("TF-IDF Matrix")
```

Out[8]:

TF-IDF Matrix

| | cat | bat | rat | fat | mat | pat | sat |
|---|---|---|---|---|---|---|---|
| **0** | 0.106000 | 0.053000 | 0 | 0.053000 | 0.000000 | 0.000000 | 0.000000 |
| **1** | 0.000000 | 0.106000 | 0 | 0.000000 | 0.053000 | 0.053000 | 0.000000 |
| **2** | 0.053000 | 0.000000 | 0 | 0.053000 | 0.053000 | 0.053000 | 0.143600 |

**c)** The term-document pair witht the highest TF-IDF weight is (Doc 3, "sat")