Student: Jing Li

1. Nearest Neighbours and the Curse of Dimensionality.

a) From the mathematical deduction, for random variable $Z = (X - Y)^2$, its expectation is:

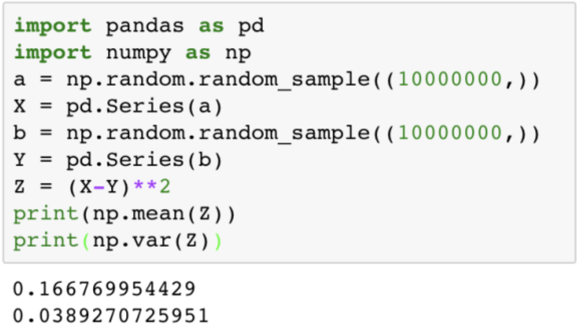
$$
\begin{aligned}
E(Z) &= \int_0^1 \int_0^1 (X-Y)^2 \, dXdY \\
&= \int_0^1 \int_0^1 (X^2 + Y^2 - 2XY) \, dXdY \\
&= \int_0^1 \left[\left(\frac{X^3}{3} + XY^2 - X^2Y \right) \Bigg|_0^1 \right] dY \\
&= \int_0^1 \left(\frac{1}{3} + Y^2 - Y \right) dY \\
&= \frac{1}{3}Y + \frac{1}{3}Y^3 - \frac{1}{2}Y^2 \Bigg|_0^1 \\
&= \frac{1}{6}
\end{aligned}
$$

and variance:

$$E(Z^2) = E[(X-Y)^2] = \int_0^1 \int_0^1 (X-Y)^2 dXdY = \frac{1}{15}$$

$$Var(Z) = E(Z^2) - E(Z)^2 = \frac{7}{180}$$

Verify the deduction using python:



```
import pandas as pd
import numpy as np
a = np.random.random_sample((10000000,))
X = pd.Series(a)
b = np.random.random_sample((10000000,))
Y = pd.Series(b)
Z = (X-Y)**2
print(np.mean(Z))
print(np.var(Z))
```

```
0.166769954429
0.0389270725951
```

b) For squared Euclidean distance $R = Z_1 + \cdots + Z_d$, where $Z_i = (X_i - Y_i)^2$, its expectation is:

$E(R) = E(Z_1) + \cdots + E(Z_d) = d * E(Z)$

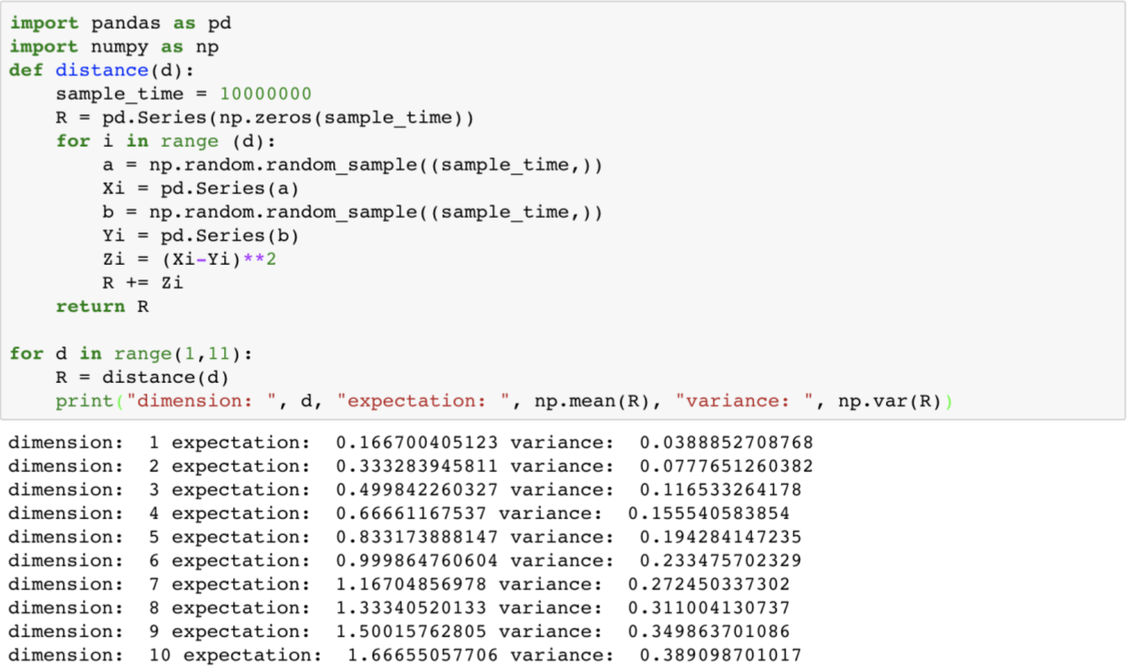
and variance:

$Var(R) = Var(Z_1) + \cdots + Var(Z_d) + d * Cov(Z_1, \ldots, Z_d)$

as Z_1, ..., Z_d is independent, $Cov(Z_1, \ldots, Z_d) = 0$

$Var(R) = Var(Z_1) + \cdots + Var(Z_d) = d * Var(Z)$

Verify the deduction using python:



```
import pandas as pd
import numpy as np
def distance(d):
    sample_time = 10000000
    R = pd.Series(np.zeros(sample_time))
    for i in range (d):
        a = np.random.random_sample((sample_time,))
        Xi = pd.Series(a)
        b = np.random.random_sample((sample_time,))
        Yi = pd.Series(b)
        Zi = (Xi-Yi)**2
        R += Zi
    return R

for d in range(1,11):
    R = distance(d)
    print("dimension: ", d, "expectation: ", np.mean(R), "variance: ", np.var(R))
```

```
dimension:  1 expectation:  0.166700405123 variance:  0.0388852708768
dimension:  2 expectation:  0.333283945811 variance:  0.0777651260382
dimension:  3 expectation:  0.499842260327 variance:  0.116533264178
dimension:  4 expectation:  0.66661167537 variance:  0.155540583854
dimension:  5 expectation:  0.833173888147 variance:  0.194284147235
dimension:  6 expectation:  0.999864760604 variance:  0.233475702329
dimension:  7 expectation:  1.16704856978 variance:  0.272450337302
dimension:  8 expectation:  1.33340520133 variance:  0.311004130737
dimension:  9 expectation:  1.50015762805 variance:  0.349863701086
dimension:  10 expectation:  1.66655057706 variance:  0.389098701017
```

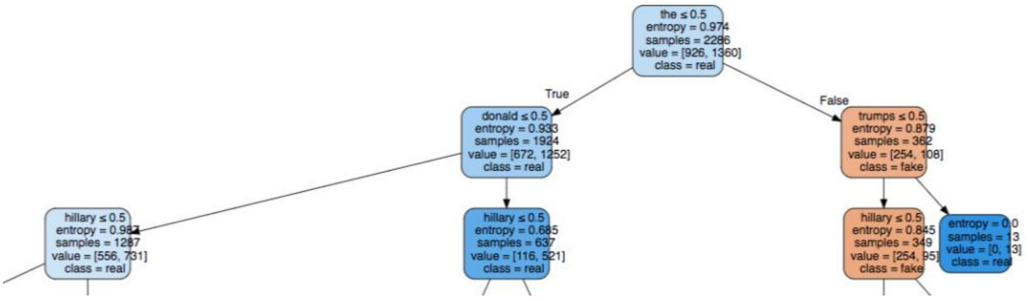
2. Decision Trees

a) load_data(): the headlines in the cleaned-up data forms a series of strings df_x, and the label of 0 (fake) and 1 (real) is stored respectively in df_y.
df_x is vectorized using *CountVectorizer*. Features of each string in the series is extracted and stored in the matrix x_cv.

In order to randomly split the dataset into training, validation and test while keeping the pairing between x_cv and df_y, an index is used. The indices are randomized and split to 70%-15%-15%. Afterwards, training, validation and test datasets with features and its labels is populated according to the randomized spliced indices.

b) In select_model, DecisionTreeClassifier with different max_depth ([5,10,20,30,40]) and split criteria (['gini', 'entropy']) is trained by training dataset and evaluated with validation dataset. The accuracy is calculated based on the percentage of matching between predicted and actual label in the validation dataset.
Based on the result, the optimal hyperparameters is (max_depth=40, criteria='entropy')

c) The first two layers of the trained decision tree is shown below



d) According to the formula of information gain, we use different keywords to split the data and computed the entropy before the split and the entropies of the two split segments. The result is shown below:

keyword	Information gain
donald	0.0880059388076393
hillary	0.038757989444715424
trump	0.04820630647640067
clinton	0.009466668624697139
love	8.474076757902793e-05

From the observation on few selected words, though sample size is too small, the topmost split did achieve the highest information gain, which align with the principle of decision tree.