CSC411H1 L0101, Fall 2018
Assignment 1

Name: JingXi, Hao
Student Number: 1000654188

Due Date: 26 March, 2018 11:59pm

1. **Nearest Neighbours and the Curse of Dimensionality.** In this question, you will verify the claim from lecture that 'most' points in a high-dimensional space are far away from each other, and also approximately the same distance. There is a very neat proof of this fact which uses the properties of expectation and variance. If it's been a long time since you've studied these, you may wish to review the Tutorial 1 *slides*[3], or the Metacademy *resources*[4].

   (a) First, consider two independent univariate random variables X and Y sampled uniformly from the unit interval $[0, 1]$. Determine the expectation and variance of the random variable Z, defined as the squared distance $Z = (X - Y)^2$. You are allowed to evaluate integrals numerically (e.g. using scipy.integrate.quad or scipy.integrate.dblquad), but you should explain what integral(s) you are evaluating, and why.

   Solution:

   Firstly, we compute the expected value of the random variable Z. Then, we have that

   $$\begin{aligned}
   E(Z) &= E((X-Y)^2) \\
   &= E(X^2 - 2XY + Y^2) \\
   &= E(X^2) - 2E(XY) + E(Y^2) \\
   &= E(X^2) - 2E(X)E(Y) + E(Y^2) \quad \# \; E(XY) = E(X)E(Y) \text{ since } X \text{ and } Y \text{ are two independent univariate random variables} \\
   &= \int_0^1 x^2 p(x)dx - 2(\int_0^1 xp(x)dx)(\int_0^1 yp(y)dy) + \int_0^1 y^2 p(y)dy \quad \# \text{ since } X \text{ and } Y \text{ are sampled uniformly from unit interval } [0,1]], \text{ thus the density functions } p(x) = 1 \text{ and } p(y) = 1 \\
   &= \int_0^1 x^2 dx - 2(\int_0^1 xdx)(\int_0^1 ydy) + \int_0^1 y^2 dy \\
   &= \tfrac{1}{3}x^3|_0^1 - 2(\tfrac{1}{2}x^2|_0^1)(\tfrac{1}{2}y^2|_0^1) + \tfrac{1}{3}y^3|_0^1 \\
   &= \tfrac{1}{3} - 2(\tfrac{1}{2})(\tfrac{1}{2}) + \tfrac{1}{3} \\
   &= \tfrac{1}{6}
   \end{aligned}$$

   Secondly, we compute the variance of the random variable Z. Then, we have that

   $$\begin{aligned}
   Var(Z) &= E(((X-Y)^2)^2) - (E((X-Y)^2))^2 \\
   &= E((X-Y)^4) - (\tfrac{1}{6})^2 \quad \# \; E((X-Y)^2) = E(Z) = \tfrac{1}{6} \\
   &= E(X^4 - 4X^3Y + 6X^2Y^2 - 4XY^3 + Y^4) - \tfrac{1}{36} \\
   &= E(X^4) - 4E(X^3)E(Y) + 6E(X^2)E(Y^2) - 4E(X)E(Y^3) + E(Y^4) - \tfrac{1}{36} \quad \# \text{ since } X \text{ and } Y
   \end{aligned}$$

are two independent univariate random variables

$= \int_0^1 x^4 dx - 4(\int_0^1 x^3 dx)(\int_0^1 y dy) + 6(\int_0^1 x^2 dx)(\int_0^1 y^2 dy) - 4(\int_0^1 x dx)(\int_0^1 y^3 dy) + \int_0^1 y^4 dy - \frac{1}{36}$

\# as before, $p(x) = p(y) = 1$

$= \frac{1}{5}x^5|_0^1 - 4(\frac{1}{4}x^4|_0^1)(\frac{1}{2}y^2|_0^1) + 6(\frac{1}{3}x^3|_0^1)(\frac{1}{3}y^3|_0^1) - 4(\frac{1}{2}x^2|_0^1)(\frac{1}{4}y^4|_0^1) + \frac{1}{5}y^5|_0^1 - \frac{1}{36}$

$= \frac{1}{5} - 4(\frac{1}{4})(\frac{1}{2}) + 6(\frac{1}{3})(\frac{1}{3}) - 4(\frac{1}{2})(\frac{1}{4}) + \frac{1}{5} - \frac{1}{36}$

$= \frac{1}{5} - \frac{1}{2} + \frac{2}{3} - \frac{1}{2} + \frac{1}{5} - \frac{1}{36}$

$= \frac{2}{5} - 1 + \frac{2}{3} - \frac{1}{36}$

$= \frac{7}{180}$

Therefore, $E(Z) = \frac{1}{6}$ and $Var(Z) = \frac{7}{180}$.

(b) Now suppose we sample two points independently from a unit cube in d dimensions. Observe that each coordinate is sampled independently from $[0, 1]$, i.e. we can view this as sampling random variables $X_1, \ldots, X_d$, $Y_1, \ldots, Y_d$ independently from $[0; 1]$. The squared Euclidean distance can be written as $R = Z_1 + \cdots + Z_d$, where $Z = (X - Y)^2$. Using the properties of expectation and variance, determine E[R] and Var[R]. You may give your answer in terms of the dimension d, and E[Z] and Var[Z] (the answers from part (a)).

Solution:

Firstly, we compute the expectation of R, which is

$E(R) = E(Z_1 + Z_2 + \cdots + Z_d)$

$= E(Z_1) + E(Z_2) + \cdots + E(Z_d)$

$= E((X_1 - Y_1)^2) + E((X_2 - Y_2)^2) + \ldots E((X_i - Y_i)^2) + \cdots + E((X_d - Y_d)^2)$

$= \frac{1}{6} + \frac{1}{6} + \cdots + \frac{1}{6}$ \# $E((X_i - Y_i)^2) = \frac{1}{6}$ where i is a positive integer and $i \in [1, d]$

$= d(\frac{1}{6})$

Secondly, we compute the variance of R, which is

$Var(R) = Var(Z_1 + Z_2 + \cdots + Z_d)$

$= Var(Z_1) + Var(Z_2) + \cdots + Var(Z_d) + Cov(Z_1, Z_2) + \cdots + Cov(Z_i, Z_j) + \cdots + Cov(Z_{d-1}, Z_d)$

\# add all $Cov(Z_i, Z_j)$, where $i \neq j$ but do not add covariance between two variables twice

$= d(\frac{7}{180}) + 0$ \# since Z_i and Z_j are mutually independent, $Cov(Z_i, Z_j) = 0$

$= d(\frac{7}{180})$

Therefore, the expectation of R is $\frac{d}{6}$ and the variance of R is $\frac{7d}{180}$.

2. **Decision Trees.** This question is taken from a project by Lisa Zhang and Michael Guerzhoy.

   In this question, you will use the scikit-learn decision tree classifier to classify real vs. fake news headlines. The aim of this question is for you to read the scikit-learn API and get comfortable with training/validation splits.

   All code should be included in the *hw1_code.py* which you submit through MarkUs.

(a) Write a function load_data which loads the data, preprocesses it using a vectorizer (http://scikit-learn.org/stable/modules/class.html#module-sklearn.feature_extraction.text), and splits the entire dataset randomly into 70% training, 15% validation, and 15% test examples.

Solution:

The code snippets of function, *load_data*, for this question can be found in *hw1_code.py*.

(b) Write a function select_model which trains the decision tree classifier using at least 5 different values of max_depth, as well as two different split criteria (information gain and Gini coefficient), evaluates the performance of each one on the validation set, and prints the resulting accuracies of each model. You should use DecisionTreeClassifier, but you should write the validation code yourself. Include the output of this function in your solution PDF (hw1_writeup.pdf).

Solution:

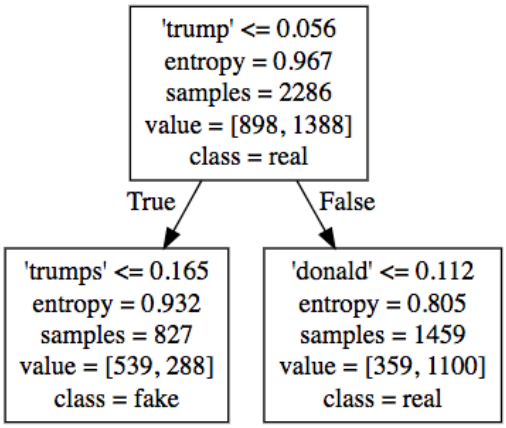
The code snippets of function, *select_model*, for this question can be found in *hw1_code.py*. Then, we show the output of this function below.

```
The accuracy for tree decision classifier with criterion Gini and max_depth 5 is 0.7995910020449898
The accuracy for tree decision classifier with criterion Entropy and max_depth 5 is 0.8057259713701431

The accuracy for tree decision classifier with criterion Gini and max_depth 25 is 0.7852760736196319
The accuracy for tree decision classifier with criterion Entropy and max_depth 25 is 0.8139059304703476

The accuracy for tree decision classifier with criterion Gini and max_depth 45 is 0.7914110429447853
The accuracy for tree decision classifier with criterion Entropy and max_depth 45 is 0.8241308793456033

The accuracy for tree decision classifier with criterion Gini and max_depth 55 is 0.7995910020449898
The accuracy for tree decision classifier with criterion Entropy and max_depth 55 is 0.7832310838445807

The accuracy for tree decision classifier with criterion Gini and max_depth 65 is 0.7730061349693251
The accuracy for tree decision classifier with criterion Entropy and max_depth 65 is 0.787321063394683
```

(c) Now let's stick with the hyperparameters which achieved the highest validation accuracy. Extract and visualize the first two layers of the tree. Your visualization may look something like what is shown below, but it does not have to be an image: it is perfectly fine to display text. It may be hand-drawn. Include your visualization in your solution PDF (hw1_writeup.pdf).

Solution:

Based on the output from question (*b*), we choose criterion to be $Information\ Gain$ ($Entropy$) and max_depth to be 45 since with these hyperparameters, it produces the highest validation accuracy. We show the first two layers of the tree below.



(d) Write a function compute_information_gain which computes the information gain of a split on the training data. That is, compute $I(Y, x_i)$, where Y is the random variable signifying whether the headline is real or fake, and x_i is the keyword chosen for the split.

Report the outputs of this function for the topmost split from the previous part, and for several other keywords.

Solution:

The code snippets of function, $compute\_information\_gain$, are able to be found in $hw1\_code.py$. The outputs of this function for the topmost split ($'trump'$) from the previous part and several other keywords are shown below.

```
The information gain for keyword trump is 0.0321365720404
The information gain for keyword trumps is 0.0474052330519
The information gain for keyword donald is 0.0450423455928
The information gain for keyword military is 0.00178895209188
The information gain for keyword obama is 0.00662025093148
```