Assignment 3 [Part 1 of 2]

Questions 1-6 are pen-and-paper exercises (brief answers and justifications are expected). Questions 7-8 are coding exercises. In all responses, please show your workings (equations, justifications, or code when applicable).

1. What is the advantage of using the Apriori algorithm in comparison with computing the support of every subset of an itemset in order to find the frequent itemsets in a transaction dataset? [0.5 marks out of 5]
2. Let $\mathcal{L}_1$ denote the set of frequent 1-itemsets. For $k \geq 2$, why must every frequent k-itemset be a superset of an itemset in $\mathcal{L}_1$? [0.5 marks out of 5]
3. Let $\mathcal{L}_2 = \{\{1,2\},\{1,4\},\{2,3\},\{2,4\},\{3,5\}\}$. Compute the set of candidates $\mathcal{C}_3$ that is obtained by joining every pair of joinable itemsets from $\mathcal{L}_2$. [0.5 marks out of 5]
4. Let S_1 denote the support of the association rule $\{\text{boarding pass}, \text{passport}\} \Rightarrow \{\text{flight}\}$. Let S_2 denote the support of the association rule $\{\text{boarding pass}\} \Rightarrow \{\text{flight}\}$. What is the relationship between S_1 and S_2? [0.5 marks out of 5]
5. What is the support of the rule $\{\} \Rightarrow \{\text{Eggs}\}$ in the transaction dataset used in Section 1 of this lab notebook? [0.5 marks out of 5]
6. In the transaction dataset used in the tutorial presented above, what is the maximum length of a frequent itemset for a support threshold of 0.2? [0.5 marks out of 5]
7. Implement a function that computes the Kulczynski measure of two itemsets $\mathcal{A}$ and $\mathcal{B}$. Use your function to compute the Kulczynski measure for itemsets $\mathcal{A} = \{\text{Onion}\}$ and $\mathcal{B} = \{\text{Kidney Beans}, \text{Eggs}\}$ in the transaction dataset used in this lab notebook. [1 mark out of 5]
8. Implement a function that computes the imbalance ratio of two itemsets $\mathcal{A}$ and $\mathcal{B}$. Use your function to compute the imbalance ratio for itemsets $\mathcal{A} = \{\text{Onion}\}$ and $\mathcal{B} = \{\text{Kidney Beans}, \text{Eggs}\}$ in the transaction dataset used in this lab notebook. [1 mark out of 5]

Sample Solutions (brief)

1. Computational efficiency: the power set of items is too large.
2. Because of the Apriori property, stating that "all nonempty subsets of a frequent itemset must also be frequent".
3. $\mathcal{C}_3 = \{\{1,2,4\},\{2,3,4\}\}$. *(0.25 marks for each correct itemset)*
4. $S_1 \leq S_2$.
5. $S_{\{\}\Rightarrow\{\text{Eggs}\}} = S_{\{\}\cup\{\text{Eggs}\}} = S_{\{\text{Eggs}\}} = \frac{N_{\{\text{Eggs}\}}}{N} = \frac{4}{5} = 0.8.$
6. 6.

In [9]:
```
# Solution to exercise 7

# 0.75 marks to implement a function to compute the Kulczynski measure
def kulczynski(frequent_itemsets, A, B):
    support = {}
    for _, row in frequent_itemsets.iterrows():
        support[row['itemsets']] = row['support']

    supAB = support[A.union(B)]

    cAB = supAB/support[A]
    cBA = supAB/support[B]

    return (cAB + cBA)/2.


# 0.25 marks to correctly compute the Kulczynski measure for {Onion} and {Kidney Beans
A = frozenset(['Onion'])
B = frozenset(['Kidney Beans', 'Eggs'])
print(kulczynski(frequent_itemsets, A, B))
```

```
0.875
```

In [10]:
```
# Solution to exercise 8

# 0.75 marks to implement a function to compute the imbalance ratio
def imbalance_ratio(frequent_itemsets, A, B):
    support = {}
    for _, row in frequent_itemsets.iterrows():
        support[row['itemsets']] = row['support']

    numerator = abs(support[A] - support[B])
    denominator = support[A] + support[B] - support[A.union(B)]

    return numerator/denominator


# 0.25 marks to correctly compute the imbalance ratio for {Onion} and {Kidney Beans,Eg
A = frozenset(['Onion'])
B = frozenset(['Kidney Beans', 'Eggs'])
print(imbalance_ratio(frequent_itemsets, A, B))
```

```
0.2500000000000001
```

Assignment 3 [Part 2 of 2]

For your answers to the assignment, please include include your workings (e.g. equations, code) when this is relevant to the question. Questions 1-2 are pen-and paper exercises. Question 3 can be addressed either on paper or using code. Questions 4-5 are coding exercises.

1. For an application on credit card fraud detection, we are interested in detecting contextual outliers. Suggest 2 possible contextual attributes and 2 possible behavioural attributes that could be used for this application, and explain why each of your suggested attribute should be considered as either contextual or behavioural. [1 mark out of 5]

2. Assume that you are provided with the University of Wisconsin breast cancer dataset from the Week 3 lab, and that you are asked to detect outliers from this dataset. Additional information on the dataset attributes can be found online. Explain one possible outlier detection method that you could apply for detecting outliers for this particular dataset, explain what is defined as an outlier for your suggested approach given this particular dataset, and justify why would you choose this particular method for outlier detection. [1 mark out of 5]

3. The monthly rainfall in the London borough of Tower Hamlets in 2019 had the following amount of precipitation (measured in mm, values from January-December 2018): {22.93, 20.69, 25.75, 23.84, 25.34, 3.25, 23.55, 28.28, 23.72, 22.42, 26.83, 23.82}. Assuming that the data is based on a normal distribution, identify outlier values in the above dataset using the maximum likelihood method. [1 mark out of 5]

4. Using the stock prices dataset used in sections 1 and 2 of this lab notebook, estimate the outliers in the dataset using the one-class SVM classifier approach. As input to the classifier, use the percentage of changes in the daily closing price of each stock, as was done in section 1 of the notebook. Plot a 3D scatterplot of the dataset, where each object is color-coded according to whether it is an outlier or an inlier. Also compute a histogram and the frequencies of the estimated outlier and inlier labels. In terms of the plotted results, how does the one-class SVM approach for outlier detection differ from the parametric and proximity-based methods used in the lab notebook? What percentage of the dataset objects are classified as outliers? [1 mark out of 5]

5. This question will combine concepts from both data preprocessing and outlier detection. Using the house prices dataset from Section 3 of this lab notebook, perform dimensionality reduction on the dataset using PCA with 2 principal components (make sure that the dataset is z-score normalised beforehand, and remember that PCA should only be applied on the input attributes). Then, perform outlier detection on the pre-processed dataset using the k-nearest neighbours approach using k=2. Display a scatterplot of the two principal components, where each object is colour-coded according to the computed outlier score. [1 marks out of 5]

Sample Solutions

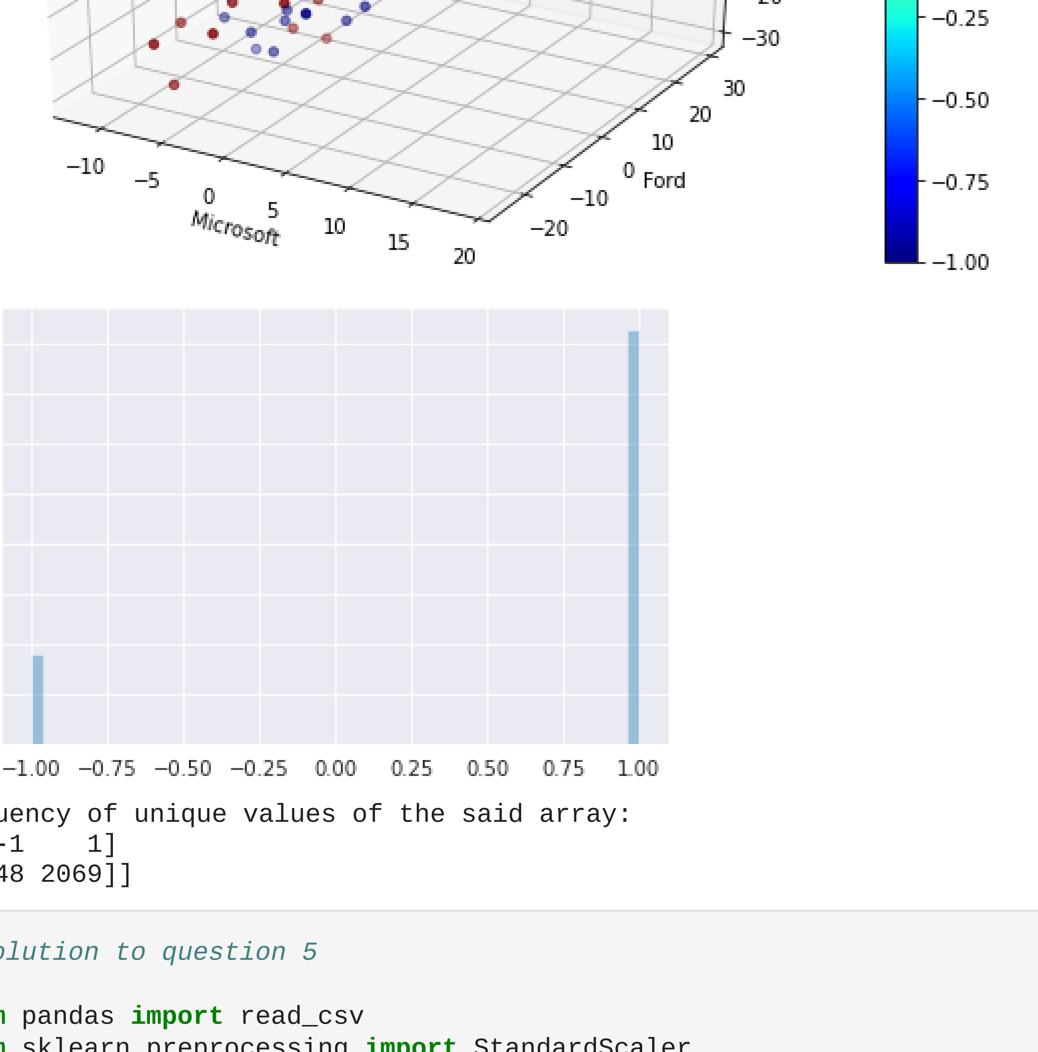
1. Contextual attributes could be age group and postal code. Behavioural attributes could be number of transacitons per month, and monthly total transactions amount. Age group and postal code can group customers into different groups, i.e. contexts. Number and amount of transactions for a given context can be used to infer credit card fraud outliers. *(0.5 marks for appropriate contextual attributes, 0.5 marks for appropriate behavioural attributes)*

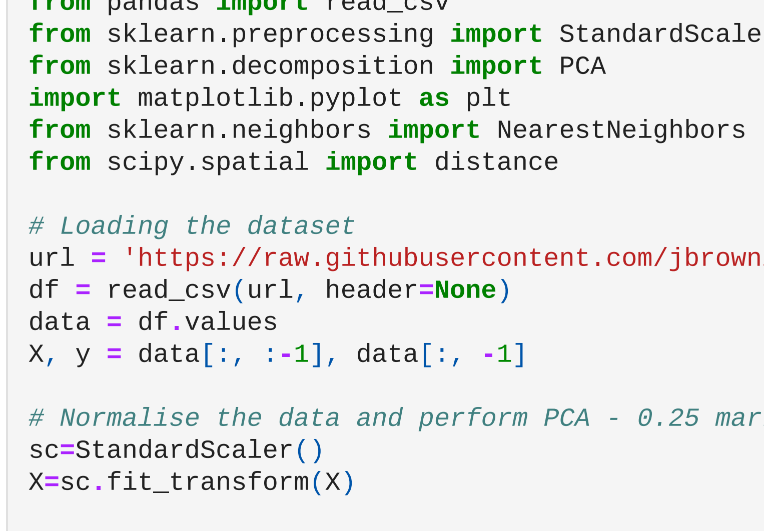
2. Approach 1: The breast cancer dataset contains class labels, denoting each data object as having either a benign or malignant tumor. We could therefore adopt a classification-based outlier detection method, where we learn a model for the "benign" class. Any data object that does not fit into the model of the "benign" class could be viewed as an outlier (and possibly as a malignant tumor). Approach 2: A different approach might be on performing clustering on the dataset, using 2 clusters (assuming one cluster for the benign tumors, and one for the malignant tumors). So we could use a clustering-based approach for outlier detection. Any data object that does not belong to any of the two clusters is identified as an outlier. *(0.25 marks for stating an appropriate outlier detection method, 0.25 marks for stating what is an outlier for this dataset, and 0.5 marks for appropriate justifications.)*

In [11]:
```
# Solution for question 3

# Marking scheme: 0.5 marks for correct result. 0.5 marks for showing workings (math o

import numpy as np

rainfall = [22.93, 20.69, 25.75, 23.84, 25.34, 3.25, 23.55, 28.28, 23.72, 22.42, 26.83

print(np.sum(rainfall))
print('Mean rainfall:', np.mean(rainfall))
print('Standard deviation of rainfall:', np.std(rainfall))

print( abs(rainfall-np.mean(rainfall))/np.std(rainfall) )

# Solution: Value in June (3.25) is an outlier since it is outside the mu+/3*std regio
# of the data (its fraction value is 3.14 > 3) under the assumption of a normal distri
```

```
270.41999999999996
Mean rainfall: 22.534999999999997
Standard deviation of rainfall: 6.130045540885756
[0.06443672 0.30097656 0.52446592 0.21288586 0.45758224 3.14597989
 0.16557789 0.93718716 0.19331015 0.01876006 0.70064732 0.20962324]
```

In [13]:
```
# Solution for question 4

import pandas as pd
import numpy as np
from sklearn.svm import OneClassSVM
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
%matplotlib inline

# Load CSV file, set the 'Date' values as the index of each row, and display the first
stocks = pd.read_csv('stocks.csv', header='infer')
stocks.index = stocks['Date']
stocks = stocks.drop(['Date'],axis=1)

# Compute delta, which denotes the percentage of changes in the daily closing price of
N,d = stocks.shape
delta = pd.DataFrame(100*np.divide(stocks.iloc[1:,:].values-stocks.iloc[:N-1,:].values
                     columns=stocks.columns, index=stocks.iloc[1:].index)
delta.head()

# Train a one-class SVM for the dataset and return outlier labels - 0.25 marks
ee = OneClassSVM(nu=0.01,gamma='auto')
X = delta.values
yhat = ee.fit_predict(X)

# Plot 3D scatterplot of outlier scores - 0.25 marks
fig = plt.figure(figsize=(10,6))
ax = fig.add_subplot(111, projection='3d')
p = ax.scatter(delta.MSFT,delta.F,delta.BAC,c=yhat,cmap='jet')
ax.set_xlabel('Microsoft')
ax.set_ylabel('Ford')
ax.set_zlabel('Bank of America')
fig.colorbar(p)
plt.show()

# Display histogram and value counts - 0.25 marks
import seaborn as sns
sns.set_style('darkgrid')
sns.distplot(yhat, bins=None, kde=False)
plt.show()
unique_elements, counts_elements = np.unique(yhat, return_counts=True)
print("Frequency of unique values of the said array:")
print(np.asarray((unique_elements, counts_elements)))

# 0.25 marks:
# One-class classifier produces a too coarse decision - a data point is either an out
# From the histogram and value counts, we see that 17.7% of the objects have been ide
```





```
Frequency of unique values of the said array:
[[   -1    1]
 [ 448 2069]]
```

In [14]:
```
# Solution to question 5

from pandas import read_csv
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.neighbors import NearestNeighbors
from scipy.spatial import distance

# Loading the dataset
url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/housing.csv'
df = read_csv(url, header=None)
data = df.values
X, y = data[:, :-1], data[:, -1]

# Normalise the data and perform PCA - 0.25 marks
sc=StandardScaler()
X=sc.fit_transform(X)

# Perform PCA - 0.25 marks
pca=PCA(n_components=2)
principalComponents=pca.fit_transform(X)
print(principalComponents.shape)

# Implement a k-nearest neighbour approach using k=2 neighbours - 0.25 marks
knn = 2
nbrs = NearestNeighbors(n_neighbors=knn, metric=distance.euclidean).fit(principalCompo
distances, indices = nbrs.kneighbors(principalComponents)
outlier_score = distances[:,knn-1]

# Plot scatterplot of principal components, colored by the outlier score - 0.25 marks
fig = plt.figure(figsize=(10,6))
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
p = plt.scatter(principalComponents[:,0], principalComponents[:,1], c = outlier_score,
fig.colorbar(p)
```

```
(506, 2)
```

Out[14]: `<matplotlib.colorbar.Colorbar at 0x7faf5f3b0240>`