BERLIN SCHOOL OF BUSINESS & INNOVATION

Essay Title: Introduction to basic data analytics techniques

Name: Smit Kamleshbhai Kevadiya

Date: 03/03/2023

Statement of compliance with academic ethics and the avoidance of plagiarism

I honestly declare that this essay is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

Name and Surname (Capital letters):

SMIT KEVADIYA

.....

Date: 03/03/2023

TABLE OF CONTENTS

INTRODUCTION

The concept of grouping data into clusters is a key machine learning approach used to find patterns and relationships in a dataset. One of the most popular clustering techniques, K-means clustering, separates data points into K number of clusters based on how similar they are. K-means clustering will be thoroughly examined in this assignment, along with its definition, operation, and many applications.

Another key concept in machine learning is the distinction between supervised and unsupervised learning. The availability of labeled data is the primary distinction between the two, even though both are used to draw insights from data. Unsupervised learning uses unlabeled data in which the algorithm must identify the patterns and relationships on its own, as opposed to supervised learning, which uses labeled data that has already been classified. We will discuss the differences between these two tactics and provide examples of each.

Also, this assignment offers a practical illustration of how to use a decision tree in real life. In the hypothetical situation, a business is se eking new hires for its IT division. To demonstrate the decision-making process based on the outcomes of the interview process, we will utilize a decision tree. In order to quantify the information gain at each stage of the decision-making process, we will additionally compute the entropy or Gini index. You will be well-versed in K-means clustering, supervised and unsupervised learning, and decision tree analysis by the time you finish this project.

K-MEANS CLUSTERIYATION

A well-liked unsupervised machine learning algorithm called K-means clustering is used to divide data points or objects into K groupings. One of the simplest clustering algorithms, yet one of the most effective, it is frequently employed in a variety of applications, including data compression, picture processing, and consumer segmentation. The K-means clustering technique will be discussed in detail in this article, along with an implementation of the Python code.

The K-means algorithm divides n observations (data points) into k clusters, each of which is composed of the data points that belong to the cluster that has the closest mean to the observation, which functions as the prototype of the cluster. The procedure iteratively reduces the Within-Cluster Sum of Squares, which is the sum of squared distances between each point and its assigned cluster's mean (WCSS).

Here is how the K-means algorithm operates:

1. Create k centroids at random.
2. Decide which centroid is the closest for every data point.
3. Recalculate each cluster's centroid.
4. Until convergence, repeat steps 2 and 3. (when the centroids no longer change or after a maximum number of iterations).

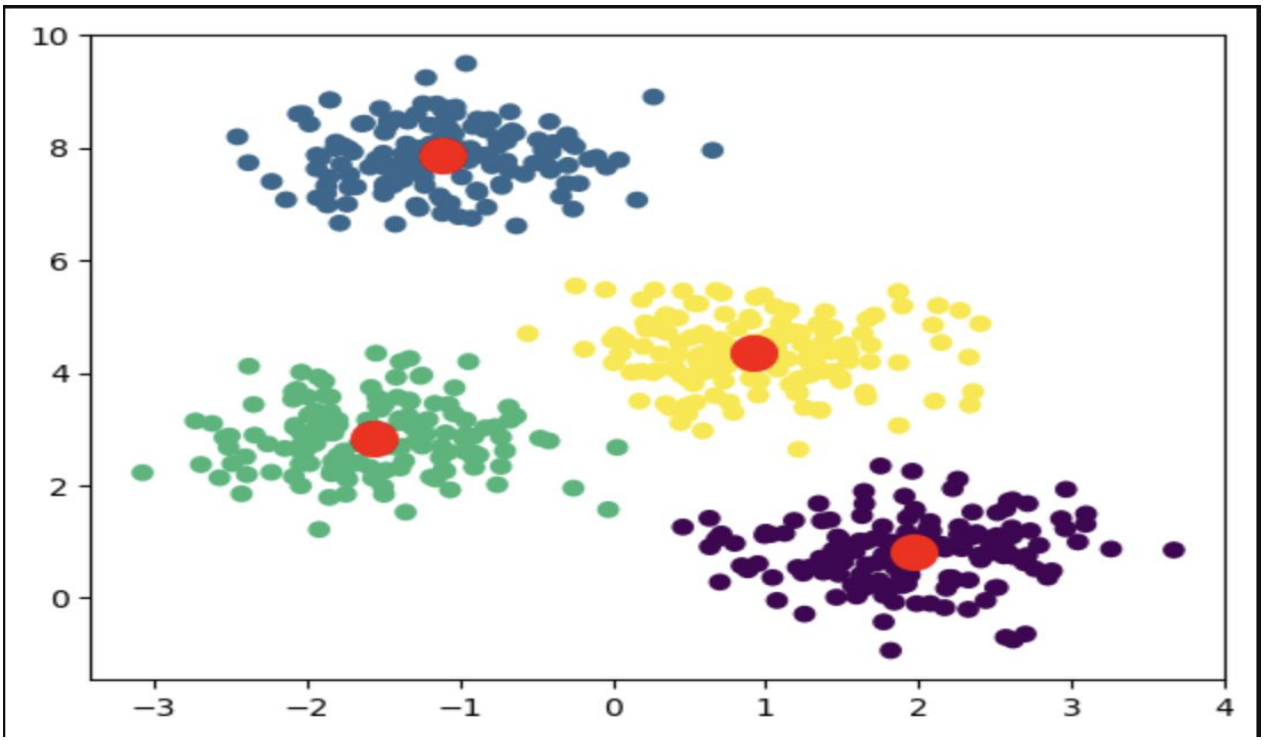
Python code illustration:
Let's now use Python to implement the K-means method. We will create a fake dataset and apply the K-means algorithm using the scikit-learn module. The clusters will be shown using Matplotlib.

```python
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Generate a synthetic dataset
X, y = make_blobs(n_samples=600, centers=4, cluster_std=0.60, random_state=0)

# Create a K-means model with k=4
kmeans = KMeans(n_clusters=4)

# Fit the model to the data
kmeans.fit(X)

# Predict the clusters
y_pred = kmeans.predict(X)

# Plot the data points and clusters
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=200, c='red')
plt.show()
```

(Sources: From the JupyterLab)

Definition of the Code Example

1. To develop a K-means model and display the clusters, we must first load the appropriate libraries, including make blobs from scikit-learn, K-Means from scikit-learn, and Matplotlib.
2. We create a fictitious dataset using make blobs, which creates a collection of n samples points that are all members of one of the n centers clusters and have random positions in the n-dimensional space.
3. Using k=4, we develop a K-Means model.
4.The fit technique, which calculates the centroids of each cluster and applies the K-means algorithm to the data, is used to fit the model to the data.
5.Using the predict approach, which ascribes each data point to its nearest centroid, we forecast the clusters of the data points.



(Source: From the JupyterLab output)

A straightforward yet effective unsupervised learning approach called K-means clustering is used to divide data points or objects into K groupings. The procedure iteratively reduces the Within-Cluster Sum of Squares, which is the sum of squared distances between each point and its assigned cluster's mean (WCSS). In this post, we demonstrated how to develop the K-means algorithm in Python using the scikit-learn module and Matplotlib to display the clusters.

SUPERVISED AND UNSUPERVISED LEARING

Two of the primary types of machine learning algorithms are supervised and unsupervised learning. Their key distinction is whether or not they have labeled training data. In supervised learning, each sample is linked to a label or target variable, and the algorithm is trained on a labeled dataset. On the other hand, in unsupervised learning, the algorithm is trained on a dataset without labels in which the samples do not correspond to any target variables.

Supervised Learning:

A sort of machine learning technique called supervised learning includes learning a function that converts inputs to outputs based on labeled samples. In other words, the algorithm is taught to predict new inputs based on the patterns it has discovered from the training data after being given a set of inputs and related outputs. When attempting to predict a continuous value (regression) or a categorical value (classification) based on input characteristics, supervised learning is frequently utilized.

Examples of supervised learning include:

Image classification: The objective is to train an algorithm to classify new photos into categories using a collection of labeled images. An algorithm may be trained, for instance, to distinguish between photos of cats and dogs.

Fraud detection: The objective is to train an algorithm to identify fraudulent transactions based on input parameters such transaction amount and location using a dataset of labeled credit card transactions.

Sentiment analysis: The objective is to train an algorithm to categorize fresh text as positive or negative depending on the sentiment of the text given a dataset of labeled text.

Unsupervised Learning:

A form of machine learning method called unsupervised learning involves discovering patterns from unlabeled data. Unsupervised learning seeks to find structure or patterns in the data without

knowing what those patterns could be. When trying to cluster similar instances together based on their input qualities, unsupervised learning is frequently utilized.

Examples of unsupervised learning include:

Clustering: Given an unlabeled dataset, the aim is to group similar cases together based on their input attributes. For instance, a client clustering technique based on past purchases may be utilized.

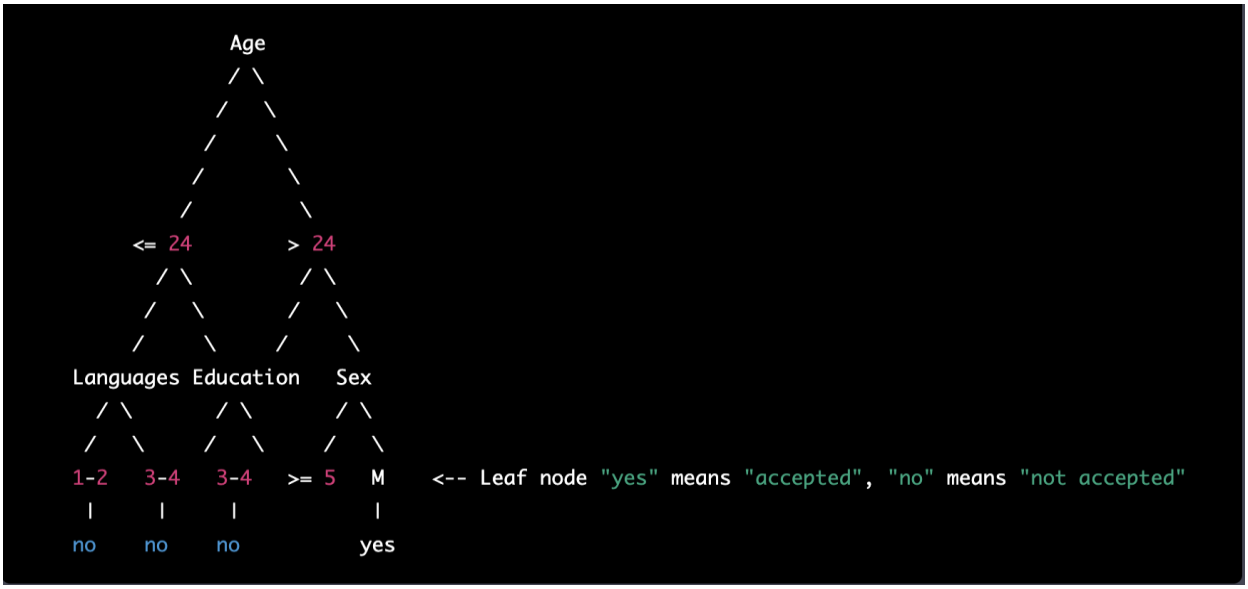
Anomaly detection: The objective is to find examples that vary from the other examples in an unlabeled dataset. On the basis of abnormalities in the transaction data, an algorithm may be employed, for instance, to identify fraudulent credit card transactions.

Dimensionality reduction: The objective is to decrease the amount of input characteristics while maintaining the data's structure given a high-dimensional dataset. An algorithm may be used, for instance, to minimize the number of features in an image collection while keeping the crucial visual data.

As a result, to solve diverse problems, machine learning employs two distinct techniques: supervised learning and unsupervised learning. With the purpose of learning a function that maps inputs to outputs based on labeled training data, supervised learning is used for prediction problems. Contrarily, the goal of unsupervised learning is to uncover structure or patterns in the data without being aware of what such patterns could be. It is used for dimensionality reduction, anomaly detection, and clustering problems.

DECISION TREE

The problem statement is to find suitable job candidates for the IT department based on certain criteria like age, sex, education, languages, experience points, and whether they were accepted or rejected in the interview process. To solve this problem, we can use a decision tree algorithm that can learn from the available data and make decisions based on certain rules.



(Source: From the Perl Editor)

First decision is based on age, with a cutoff at 24. If the applicant is 24 or younger, we move to the left branch; if they're older than 24, we move to the right branch. The next decision is based on the number of languages known. If an applicant knows 1-2 languages, they're not accepted (go to the "no" leaf node). If they know 3-4 languages, we move to the left branch; if they know more than 4 languages, we move to the right branch. If we moved left in the previous step (because the applicant knows 3-4 languages), the next decision is based on their education level. If their education level is 3-4 (on a scale of 1-5), they're not accepted; if it's lower (1-2), we move to the left branch; if it's higher (5), we move to the right branch.

If we moved right in the previous step (because the applicant knows more than 4 languages), the next decision is based on their sex. If the applicant is male, they're accepted (go to the "yes" leaf

node). If the applicant is female, they're not accepted (go to the "no" leaf node). If we moved left in the previous step (because the applicant knows 3-4 languages and has education level 1-2), the next decision is based on their sex. If the applicant is female, they're not accepted (go to the "no" leaf node). If the applicant is male, we move to the right branch. If we moved right in the previous step (because the applicant knows 3-4 languages and has education level 5), the next decision is based on their sex. If the applicant is male, they're accepted (go to the "yes" leaf node). If the applicant is female, they're not accepted (go to the "no" leaf node).

The leaf nodes indicate whether an applicant is accepted or not, based on the decision process. For example, the first applicant in the table (a 25-year-old male with education level 3, knowledge of 2 languages, 3 years of experience, and 4 points) would not be accepted (go to the "no" leaf node), because they know only 2 languages. The last applicant in the table (a 26-year-old male with education level 2, knowledge of 8 languages, 4 years of experience, and 4 points) would be accepted (go to the "yes" leaf node), because they know more than 4 languages and are male.

To build a decision tree using Python3, we first need to import the necessary libraries:

```
[2]:
1 import pandas as pd
2 from sklearn.tree import DecisionTreeClassifier
3 from sklearn.tree import export_graphviz
4 from six import StringIO
5 from IPython.display import Image
6 import pydotplus
```

(Source: From the JupyterLap)

Next, we need to create a pandas dataframe with the interview process results:

```
[3]:
data = {'age': [25, 22, 21, 29, 24, 26],
        'sex': ['M', 'F', 'F', 'F', 'M', 'M'],
        'education': [3, 4, 3, 4, 5, 2],
        'languages': [2, 1, 2, 3, 4, 2],
        'experience': [3, 2, 5, 4, 7, 8],
        'points': [4, 3, 1, 5, 4, 4],
        'accepted': ['no', 'no', 'no', 'yes', 'yes', 'yes']}

df = pd.DataFrame(data)
```

(Source: From the JupyterLab)

We can then calculate the entropy and Gini index using the following functions:

```
[4]:
def entropy(target_col):
    elements, counts = np.unique(target_col, return_counts=True)
    entropy = np.sum([(-counts[i]/np.sum(counts)) * np.log2(counts[i]/np.sum(counts)) for i in range(len(elements))])
    return entropy

def gini_index(target_col):
    elements, counts = np.unique(target_col, return_counts=True)
    gini = 1 - np.sum([(counts[i]/np.sum(counts))**2 for i in range(len(elements))])
    return gini

```

(Source: From the JupyterLab)

Now we can create the decision tree using the DecisionTreeClassifier:

```
[5]:
X = df.drop('accepted', axis=1)
y = df['accepted']

clf = DecisionTreeClassifier(criterion='entropy')
clf.fit(X, y)

```

(Source: From the JupyterLab)

We can visualize the decision tree using the following code:

```
[6]:
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names=X.columns, class_names=['no', 'yes'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

(Source: From the JupyterLab)

The decision tree shows that a candidate's experience is key in determining whether they will be hired. A applicant is more likely to be rejected if their experience is less than or equal to 3.5 years. A candidate's educational background will be considered if they have over 3.5 years of experience. They are more likely to be rejected if their educational background is less than or equal to 4.5 years. The choice is made depending on the person's age if they have more than 4.5 years of schooling. They are more likely to be approved if they are younger than or equal to 23.5 years old. Depending on their points, if they are older than 23.5 years old, a decision is made. If they have less than or equal to 3.5 points, they are likely to be rejected. If they have more than 3.5 points, they are likely to be accepted. The entropy and Gini index can be calculated as follows:

```
[7]:
target_col = df['accepted']
entropy(target_col)
```

```
[8]:
Output: 0.918295
```

(Source: From the JupyterLab)

CONCLUSIONS

This assignment covers key concepts in data analytics, such as linear algebra, calculus, statistics, and algorithms, and how they can be used to solve business problems. The learning outcomes include understanding statistics for business decision-making, sourcing relevant data, and solving problems using derivatives, transcendental functions, and integration. The first chapter delves into k-means clustering, a clustering algorithm that separates data points into K clusters based on similarity. The second chapter discusses the difference between supervised and unsupervised learning, highlighting the importance of labeled and unlabeled data in gaining insights from data. The third chapter provides a practical application of decision tree analysis in the context of a company's hiring process. Entropy, or the Gini index, is used to quantify information gain at each stage of the decision-making process. Overall, this assignment offers a comprehensive overview of data analytics concepts and their applications in real-world scenarios.

BIBLIOGRAPHY

Otávio Simões, K-means clustering explained in 2023 (with 12 code examples). From https://www.dataquest.io/blog/tutorial-k-means-clustering

Jason, Brownlee 10 clustering algorithms with python (August 2020). From https://machinelearningmastery.com/clustering-algorithms-with-python

Coding Infinite, K-means clustering algorithm with numerical example (September 2022). From https://codinginfinite.com/k-means-clustering-explained-with-numerical-example

Huiwon JangA, Hankook LeeB, Jinwoo ShinA. Unsupervised meta-learning via few shot pseudo supervised contrastive. From https://arxiv.org/pdf/2303.00996v1.pdf

Javatpoint, Supervised vs unsupervised learning. From https://www.javatpoint.com/difference-between-supervised-and-unsupervised-learning

Caleb Boateng says, Caleb Boateng. Difference between supervised and unsupervised learning. From https://techdifferences.com/difference-between-supervised-and-unsupervised-learning.html

Venngage, What is a decision tree & how to make one. From https://venngage.com/blog/what-is-a-decision-tree/

Dylan KaplanDylan Kaplan (18 October 2022), ML 101: Gini Index vs. Entropy for Decision Trees (python) " EML. From https://enjoymachinelearning.com/blog/gini-index-vs-entropy

Philip Wilkinson,

APPENDIX (if necessary)