Machine Learning algorithms, perspectives, and real-world application: Empirical evidence from United States trade data

Aggarwal, Sakshi

Indian Institute of Foreign Trade

3 March 2023

Machine Learning algorithms, perspectives, and real-world application: Empirical evidence from United States trade data

Sakshi Aggarwal

Abstract

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed. It is one of today's most rapidly growing technical fields, lying at the crossroads of computer science and statistics, and at the core of artificial intelligence (AI) and data science. Various types of machine learning algorithms such as supervised, unsupervised, semi-supervised, and reinforcement learning exist in this area. Recent progress in ML has been driven both by the development of new learning algorithms theory, and by the ongoing explosion in the availability of vast amount of data (commonly known as "big-data") and low-cost computation. The adoption of data-intensive ML-based methods can be found throughout science, technology, and commerce, leading to more evidence-based decision-making across many walks of life, including finance, manufacturing, international trade, economics, education, healthcare, marketing, policymaking, and data governance. The present paper provides a comprehensive view on these machine learning algorithms that can be applied to enhance the intelligence and capabilities of an application. Moreover, the paper attempts to determine the accurate clusters of similar industries in United States that collectively account for more than 85 percent of economy's aggregate export and import flows over the period 2002-2021 through clustering algorithm (unsupervised learning). Four clusters of mapping labels have been used, namely the low investment (LL), category 1 medium investment (HL), category 2 medium investment (LH) and high investment (HH). The empirical results indicate that machinery and electrical equipment is classified as a high investment sector due to its efficient production mechanism. The analysis further underlines the need for upstream value chain integration through skill-augmentation and innovation especially in low investment industries. Overall, this paper aims to explain the trends of ML approaches and their applicability in various real-world domains, as well as serve as a reference point for academia, industry professionals and policymakers particularly from a technical, ethical, and regulatory point of view.

Keywords Machine learning, Artificial intelligence, Clustering, K-means, international trade

Introduction

Since the evolution of mankind, humans have been using various kinds of technologies to accomplish specific tasks in a simpler way. For societies to thrive and evolve, technological innovations have become necessary, leading to the invention of different machines. These machines have not only made human life easier by performing household tasks but also enabled the people to meet the needs of their lives including travelling, industries, computing, social media streaming etc. Machine Learning is the one among them that provides systems with the ability to learn and enhance from experience automatically without being explicitly programmed and is generally referred to as the most popular latest technologies in the fourth industrial revolution.

We are living in the age of big data, advanced analytics, and data science, where individuals' activity is, digitally recorded, connected to a data source. For instance, the current electronic world has a wealth of data, such as trade data, business data, financial data, COVID-19 data, smart city data, social media data, etc. The data can be classified as structured, semi-structured, or unstructured, which is undoubtedly increasing day-by-day. The trend of "big data growth" (Beyer and Laney 2012; Manyika et al., 2011; Bendre and Thool, 2016; Idrees et al., 2019) or "data deluge" (Hey and Trefethen, 2003; Baraniuk et al., 2011; Bevan 2015) has not only triggered tremendous hype and buzz, but more importantly presents enormous challenges that in turn bring incredible innovation and economic opportunities. The expansion of world trade between emerging economies, due to splitting of production processes into several stages carried out in varied geographical locations (Athukorala and Yamashita, 2006; Cingolani et al., 2018; Kano et al., 2020; Aggarwal et al., 2021), is one such positive outcome among numerous prevailing opportunities. Extracting information from the data can be used to create various smart applications as well as in conducting data analysis through advanced analytics in different fields, such as science, manufacturing, healthcare, cybersecurity, economic policy, and governance (Pugliese et al., 2021, Aggarwal et al., 2022; Aggarwal et al., 2023).

The present analysis provides a comprehensive view on various types of *machine learning algorithms* that can be applied to enhance the intelligence and the capabilities of an application. Thus, the key contribution of this study is to explain the principles of different machine learning techniques and their application in the field of international trade and economics. The purpose of this paper is, therefore, to provide a basic guide for those academia, industry professionals and policymakers who want to study, research, and develop data-driven automated and intelligent systems in the relevant areas based on the machine learning techniques.

The rest of the paper is organized along the following lines. First, a brief review of literature on ML methods is presented in the second section. The third section explains the types of real-world data and machine learning algorithms in a broader sense. Brief explanation of different machine learning algorithms is discussed in the subsequent section followed by data and methodology used in the analysis. The next section holds the empirical results and on the basis of that conclusions are drawn in the penultimate section, and the final section discusses the scope of future work.

Literature Survey

Artificial Intelligence (AI), and in particular, machine learning has progressed remarkably in recent years as key instruments to intelligently analyze such data and to develop the corresponding real-world applications (Chen et al., 2018; Liu et al., 2020; Koteluk et al., 2021; Sarker, 2021, Pugliese et al., 2021). Thus, machine learning has emerged as the state-of-the-art choice for developing practical software for robotics, autonomous vehicle control, human-computer interaction, computer vision, speech recognition, text mining and language processing (Dai et al., 2005; Jordan and Mitchell, 2015; Cummins et al., 2018; Hegde et al., 2019; Le Glaz et al., 2021, Chowdary et al., 2021) (Fig. 1). The effect of machine learning has also been widely felt across vertical production networks participating in global value chains (Hanson et al., 2005; Aggarwal, 2017; Aggarwal et al., 2021; Aggarwal et al., 2023), trade in tasks (Yi, 2003; Grossman and Rossi-Hansberg, 2008), the second great unbundling (Baldwin, 2013, Kowalski et al., 2015), formulation of trade and monetary policy (Das and Mandal, 2000; Ramachandran, 2004; Aggarwal, 2016, Aggarwal et al., 2022), and so on.

There has been a similar broad range of effects across sciences, as machine learning methods could assist scientists in their discovery of cancer classification through DNA microarray analysis (Tan and Gilbert, 2003; Wang et al., 2005, Sikora, 2015; Mirsadeghi et al., 2021), or that of predicting SARS-CoV-2 among the COVID-19 infected patients. The recent pandemic has forced the use of machine learning for discovering new candidate drugs and vaccines in silico (Alakus et al., 2020; Keshavarzi Arshadi et al., 2020; Lalmuanawma and Hussain, 2020; Clement et al., 2021; Zoabi et al., 2021).

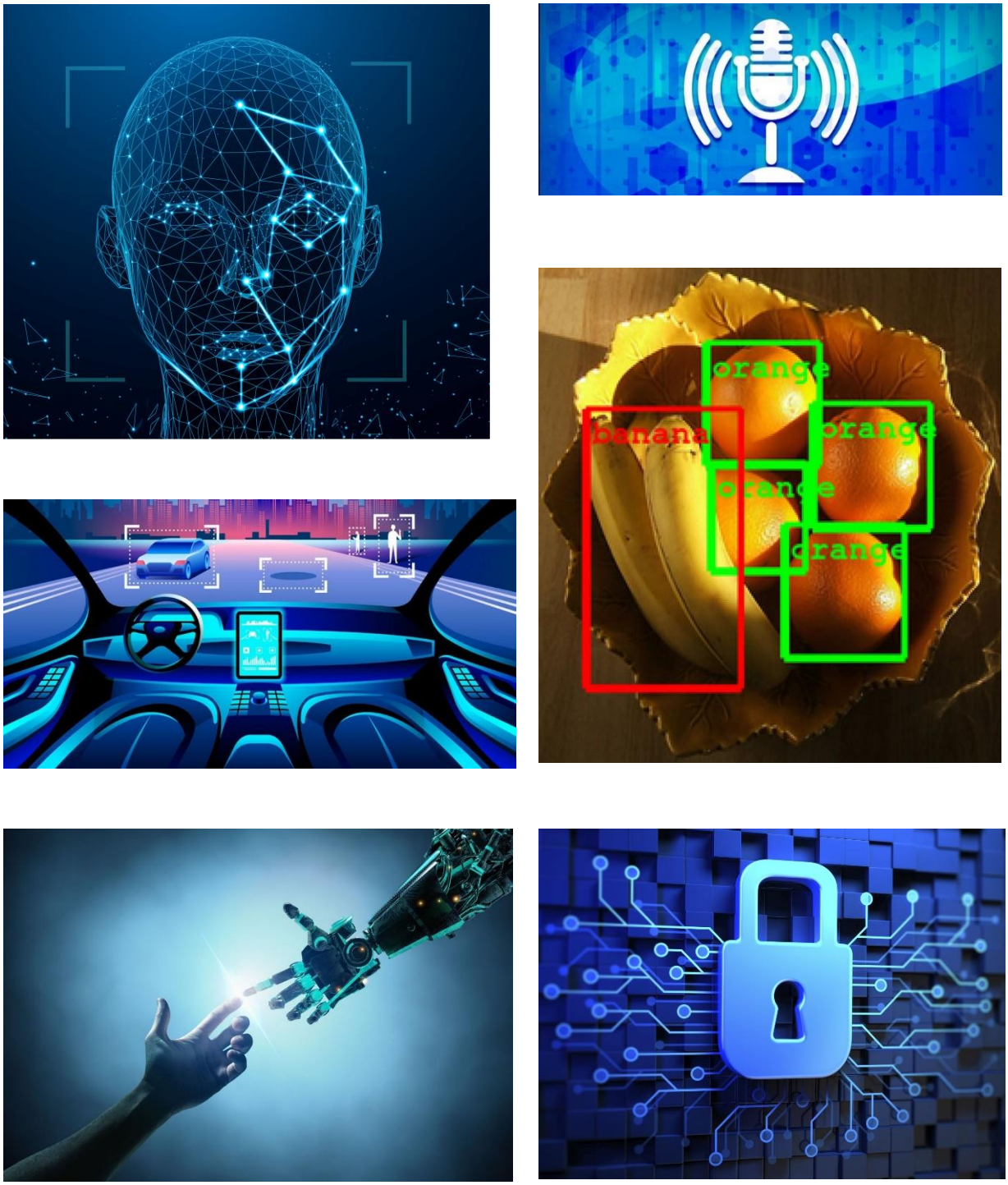


Fig 1. Real-world applications of machine learning. Machine learning is having a substantial effect on many areas of science and technology; examples of recent applied success stories include facial recognition and speech recognition (top panel), autonomous vehicle control and object recognition (middle panel), and human-computer interaction and cybersecurity (bottom panel).

In general, the efficacy of the machine learning model depends on the nature, characteristics, and pattern of data as well as the performance of the learning algorithm. For these reasons, a diverse

array of ML algorithms (such as supervised, unsupervised, semi-supervised and reinforcement learning) has been developed to cover the wide variety of data across different ML problems (Tsai et al., 2009; Das et al., 2015; Singh et al., 2016; Mohammed et al., 2016; Mahesh, 2020). Furthermore, data pre-processing and exploration are essential tasks in constructing meaningful summaries of the data. In the area of machine learning algorithms, classification analysis, regression, data clustering, feature engineering, dimensionality reduction and support vector machines exists to build data-driven systems (Witten et al., 2005; Han et al., 2011; Ray, 2019). Additionally, deep learning originated from the artificial neural network that can be used to intelligently analyze data, which is known as part of a wider family of machine learning approaches (Burrell, 2016; Cao, 2017; Sarker, 2021). Thus, selecting an appropriate learning algorithm that is suitable for target application in a specific domain is challenging. This is because different learning algorithms entails different purpose due to varying data characteristics. Therefore, it is important to understand the dynamics of data, through exploratory analysis, principles of various machine learning algorithms and their applicability before applying it in the real-world applications.

Types of real-world data

Machine learning algorithms typically consumes and process data to learn the key attributes, related patterns and trends about individuals, businesses, processes, transactions, events, and so on. The following section discusses various types of real-world data.

Data can be of various forms, such as structured, semi-structured, or unstructured, and the reliability of any machine learning model depends on the accuracy of data. Besides, the "metadata" is another type that typically represents data about data. It provides additional information about a specific set of data. In the following, we briefly discuss these types of data.

- Structured: It adheres to a pre-defined data model and conforms to a tabular format with relationship between different rows and columns, which is highly organized and easily accessed, and used by an entity or a computer program. Common examples of structured data are Excel files or SQL databases.
- Unstructured: On the other hand, there is no pre-defined data model or organization for unstructured data, that are typically text-heavy, resulting in ambiguities that make it more difficult to understand, process and analyze such datasets. For instance, audio files, images, videos, emails, sensor data, PDF files, No-SQL databases, and other multimedia materials are examples of unstructured data.
- Semi-structured: It is a form of a structured data that does not conform with the formal structure of data models associated with relational databases, but nonetheless contain tags or other markers to separate semantic elements and enforce hierarchies of records that make it easier to analyze. This property of semi-structured data is also known as "self-describing" structure. HTML, XML, JSON documents, NoSQL databases, etc., are some examples of semi-structured data.
- Metadata: It describes the relevant data information, giving it more significance for data users. A basic example of document's metadata might be author, file size, date generated by the document, keywords to define the document, etc.

In today's era of machine learning and data science, researchers have used plethora of datasets for conducting different kinds of analysis serving varied purposes. The recent studies not only highlight extensive research on building smart application, by incorporating cybersecurity datasets, smartphone datasets, etc. but have also conducted data-driven analysis on health, COVID data, etc. and many more in various application domain (Moustafa and Slay, 2015; Eagle and Pentland, 2006; Agarwal et al., 2017; Lade et al., 2017, Khadse et al., 2018; Harmon et al., 2020; Mohamadou et al., 2020). The data can be in different types as discussed above, which may vary from application to application when analyzing real-world datasets. To analyze such data in a particular problem domain, and to extract the meaningful insights and knowledgeable information from the data for building the real-world intelligent applications, different types of machine learning techniques can be used according to their learning capabilities as discussed in the following section.

Types of machine learning techniques

Machine learning involves the development and deployment of algorithms that, analyze the data and its properties, and determine the actions in response to specific inputs from the environment by using statistical tools. Generally, machine learning based analysis are dynamic in nature and improve its learning algorithms as more data is introduced (Duda et al., 2001; Bishop, 2006, Mohammed et al., 2016). Machine learning algorithms are broadly classified into four categories: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning as depicted in Fig 2.

- Supervised learning

  Supervised learning relies on machine learning tasks to learn a function that maps an input to an output based on sample input-output pairs (Han et al., 2011). It uses labeled training data and a collection of training examples to infer a function. Supervised learning is carried out when certain goals are identified to be accomplished from a certain set of inputs i.e., a *task-driven approach* (Sarker et al, 2020). In this learning process, computation of error is based on the difference between calculated and predicted output and adjusting the error for achieving the expected output. For instance, linear and logistic regression, naïve bayes classification, support vector machines (SVMs) are some examples of the supervised learning algorithms.

- Unsupervised learning

  Unsupervised learning analyzes unlabeled datasets without human interference, i.e., a *data-driven process*. In unsupervised learning, the algorithm optimally separates the samples into different classes based on the features of the training data alone, without corresponding labels (Figueiredo, 2002; Dike, 2018). Such type of analysis is widely used for extracting generative features, identifying meaningful trends and structures, groupings in results, and exploratory purposes. The most common unsupervised learning algorithms are clustering, dimensionality reduction, anomaly detection, density estimation, feature learning, etc.

- Semi-supervised learning

Semi-supervised learning can be defined as a hybridization of the above-mentioned supervised and unsupervised methods, as it deploys both labeled and unlabeled data. Thus, this kind of mechanism falls between learning "without supervision" and learning "with supervision". In the real world, there are several instances pertaining to shortages of labeled data and existence of numerous unlabeled data, where semi-supervised learning is useful (Van Engelen and Hoos, 2020; Zhou and Zhou, 2021). The ultimate goal of a semi-supervised learning model is to provide a better outcome for prediction than that produced using the labeled data alone from the model. Such a method is commonly used in fraud detection, machine translation and text classification.

- Reinforcement learning

  Reinforcement learning is a type of machine learning algorithm that typically operate sequentially to automatically evaluate the optimal behavior in a particular environment to improve its efficiency, i.e., an environment driven approach (Kaelbling et al., 1996; Buşoniu et al., 2010). This type of learning is based on reward or penalty, and its ultimate goal is to use insights obtained from environmental activists to take action to increase the reward or minimize the risk. It is a powerful tool for training AI models that could optimize the operational efficiency of sophisticated systems, such as robotics, autonomous driving tasks, manufacturing and supply chains, however, not preferable to use it for solving the basic or straightforward problems (Mohammed et al, 2016; Pugliese et al., 2021; Sarker, 2021).

Thus, different types of machine learning techniques can help to build effective models, in various application areas, according to their learning capabilities, depending on the nature and characteristics of the data as discussed earlier. In Table 1, we summarized various types of machine learning techniques with examples that provide a comprehensive view of machine learning algorithms that can be applied to enhance the intelligence and learning capabilities of a data-driven application.

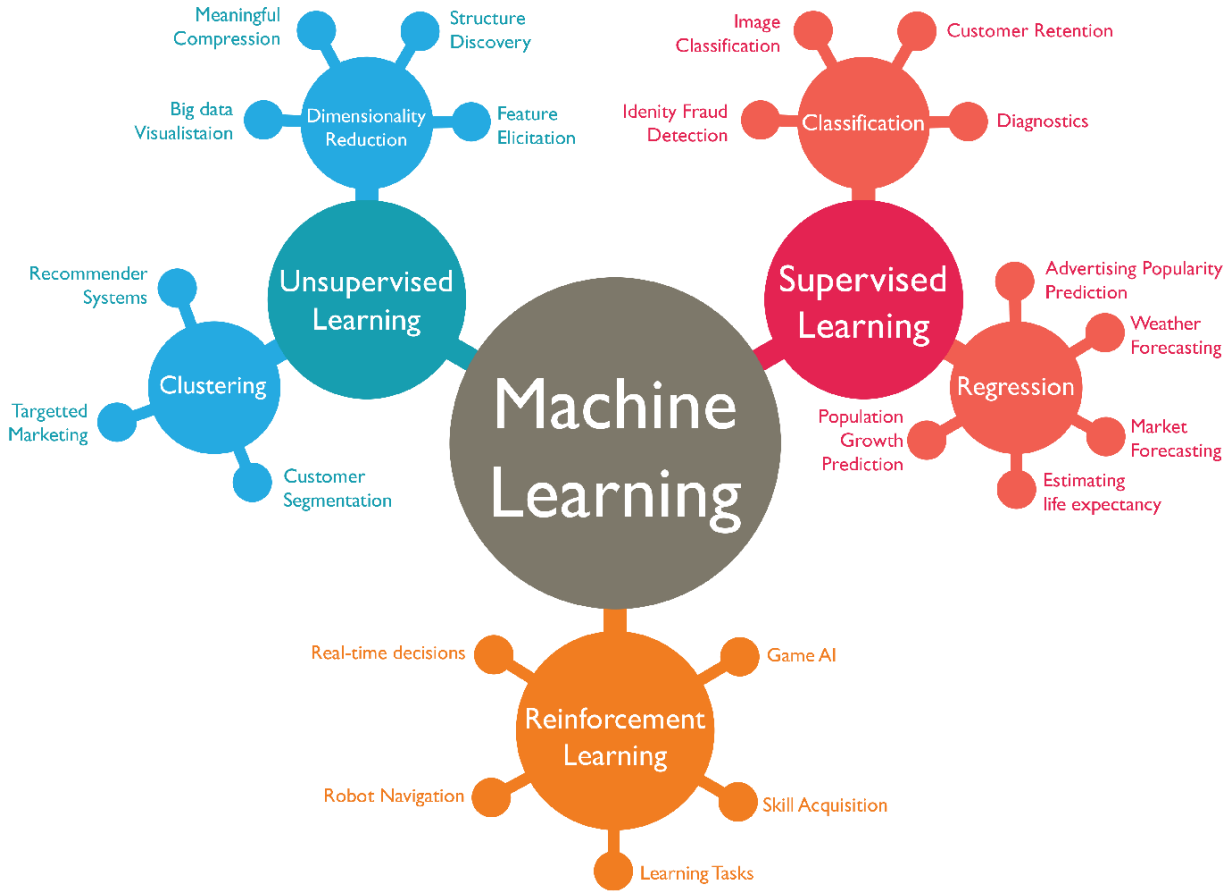


Fig 2. Machine learning algorithm classification. Classification of the main machine learning techniques, namely supervised learning, unsupervised learning, and reinforcement learning with some examples.

Table 1. Types of machine learning algorithms and real-world application examples.

Learning type	Description	Model building	Examples	Pros & cons
Supervised learning	Data is labeled (task-driven approach)	Models learn from labeled data. This type of learning comprises regression, classification, decision trees, random forests, Naïve Bayes classification and neural network.	Predicting house prices, cancer patients' classification, face and speech recognition.	Pros: exact idea about the classes in the training set; helpful in prediction and classification problems from given data and labels. Cons: cannot handle complex tasks in machine learning, cannot classify data by discovering its feature on its own, higher computation time in case of large datasets.
Unsupervised learning	Data are not labeled (data-driven approach)	Models learn from unlabeled data on the basis of the features of the training data alone, without corresponding labels. This type of learning comprises k-means clustering, hierarchical clustering, and principal component analysis (PCA).	Market segmentation, social network analysis, DNA classification	Pros: It can detect what human eyes cannot understand, the potential of hidden patterns can be powerful for the business especially for the purpose of fraud detection, unexplored territories, and new ventures for businesses. Cons: It is expensive as it might require human intervention to understand the patterns and correlate them with the domain knowledge. It is heavily dependent on the machine and the results often have lesser accuracy.
Semi-supervised learning	The algorithm works with labeled and unlabeled data	Models are built using combined labeled and unlabeled data. This type of learning includes both classification and clustering.	Text document classifier, text filtering, fraud detection.	Pros: It is easy to understand, it reduces the amount of annotated data used. Cons: Iteration results are not stable, it is not applicable to network level analysis, it has lower accuracy.
Reinforcement learning	The algorithm operates sequentially to automatically evaluate the optimal behavior in a particular context or environment to improve its efficiency (environment – driven approach).	Models are based on reward or penalty. This type of learning uses classification.	Traffic forecasting service, computer games, machinery application, medicine, surgery.	Pros: It can be used to solve the complex problems that cannot be solved by other techniques. Cons: Too much reinforcement learning can lead to an overhead of states, which can diminish the results, it needs a lot of data and a lot of computation.

Source: Author's compilation

Machine Learning Algorithms

In this section, various machine learning algorithms that include classification analysis, regression analysis, data clustering, feature engineering for dimensionality reduction is discussed. A general framework of machine learning based predictive model has been demonstrated in Fig 3. Wherein the model is trained from the historical data using machine learning algorithms to predict a model in the training phase and thereafter, the predictions is generated from the new data based on the predictive model of the previous phase.

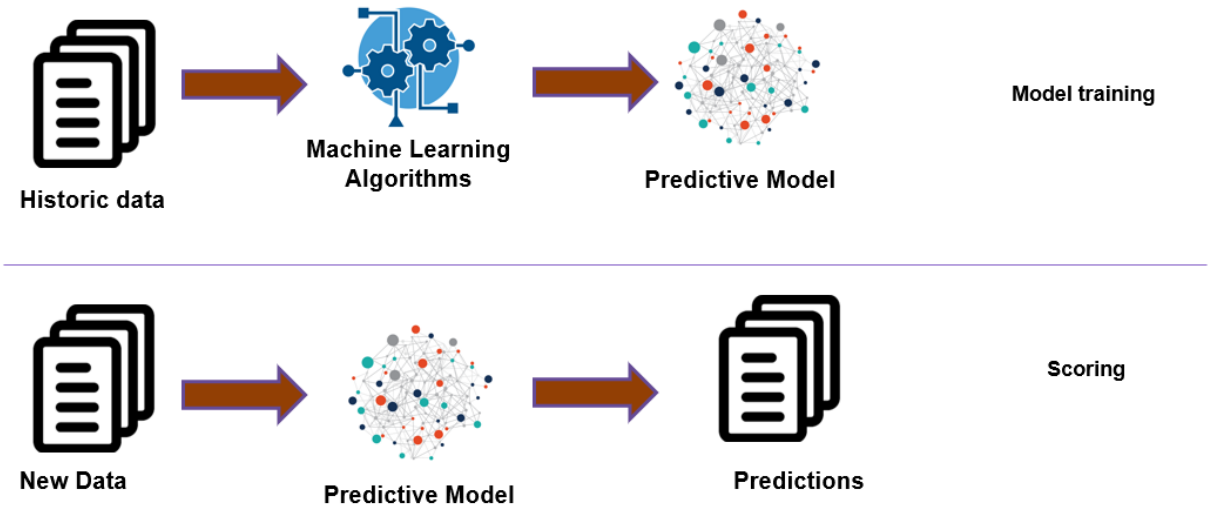


Fig 3. A general structure of machine learning based predictive model depicting both the training and testing phase.

1. Logistic regression

Logistic regression uses a logistic function (also known as the sigmoid function) to map any real-valued input to a value between 0 and 1. The logistic function is defined as:

$$\Phi(z) = \frac{1}{1 + e^{-z}}$$

where z is the input and Φ(z) is the output of the function. Once the logistic model is trained, new features of the model can be predicted. This has been illustrated in Fig 4 wherein the predicted probability value can be obtained by passing the input features (Kleinbaum et al., 2002; Nick and Campbell, 2007). If the predicted probability is greater than a certain threshold (usually 0.5), then the positive class (1) is predicted, otherwise the negative class (0) is predicted.

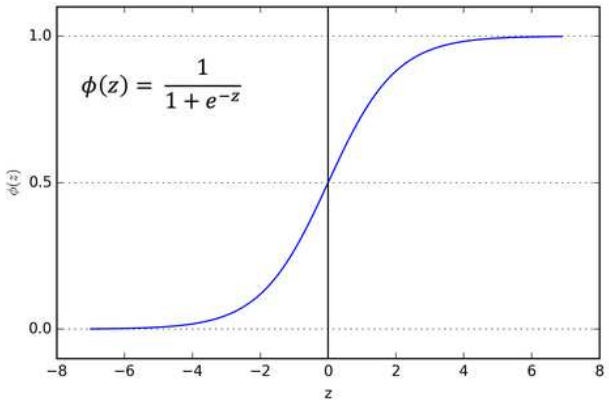


Fig 4. Sigmoid function

Logistic Regression Pseudo Code:

```
function logistic_regression(X, y, learning_rate, num_iterations):
    # Step 1: Initialize weights and bias to zeros
    w = zeros(X.shape[1])
    b = 0

    # Step 2: Gradient descent loop
    for i in range(num_iterations):
        # Step 3: Calculate the linear model output
        z = dot(X, w) + b

        # Step 4: Calculate the logistic function output (predicted probabilities)
        y_pred = sigmoid(z)
        # Step 5: Calculate the cost function
        cost = (-1/m) * sum(y*log(y_pred) + (1-y)*log(1-y_pred))

        # Step 6: Calculate the gradients
       dw = (1/m) * dot(X.T, (y_pred - y))
       db = (1/m) * sum(y_pred - y)

        # Step 7: Update weights and bias
        w = w - learning_rate * dw
         b = b - learning_rate * db

    return w, b
```

where,

x is the input feature matrix (size m x n)
y is the target variable (size m x 1)
learning rate is the step size used in gradient descent
num_iterations is the maximum number of iterations for gradient descent
w is the weight vector (size n x 1)
b is the bias scalar
sigmoid is the logistic function that maps any real value into the range [0,1]

2. Simple and multiple linear regression

This is the most popular machine learning modeling techniques that has been widely used. In this technique, the dependent variable is continuous, while the independent variable can be continuous or discrete, and the form of the regression line is linear. Linear regression creates a relationship between the dependent variable (Y) and one or more independent variable (X) using the best fit straight line also known as *regression line* (Su et al., 2012; Montgomery et al., 2021). In simple linear regression, the model is trained on the dataset of input features (x) and corresponding continuous labels (y) to predict the label for a new set of input features. The simple linear regression can be represented as:

$$y = a + bx + e$$

Similarly, in the multiple linear regression, the model is trained on the dataset of m input features (x1, x2, x3, …… xm) and corresponding continuous labels (y) to predict the label for a new set of input features. The multiple linear regression can be represented as:

$$y = a + b_1x_1 + b_2x_2 + b_3x_3 + \ldots\ldots + b_mx_m + e$$

where a is the intercept, b is slope of the line and e is the error term. This equation can be used to predict the value of the target variable based on the given predictor variable(s).

To train the model, a cost function is determined to measure how well the predicted outputs match the true labels. The cost function for multiple linear regression is the mean squared error (MSE):

$$MSE = \frac{1}{n} * \sum(y - y\_hat)^2$$

where n is the total number of training examples.

Simple and Multiple Regression Pseudo Code:

```
# Simple Linear Regression

# Initialize the intercept and coefficient
intercept = 0
coefficient = 0

# Set the learning rate and number of iterations
```

```
learning_rate = 0.01
num_iterations = 1000

# Calculate the mean of x and y
x_mean = mean(x)
y_mean = mean(y)

# Calculate the total number of training examples
n = length(x)

# Train the model using gradient descent
for i in 1:num_iterations:
    # Calculate the predicted output for each training example
    y_pred = intercept + coefficient * x

    # Calculate the error between the predicted output and the true label
    error = y_pred - y

    # Calculate the partial derivative of the cost function with respect to the intercept
    d_intercept = (2/n) * sum(error)

    # Calculate the partial derivative of the cost function with respect to the coefficient
    d_coefficient = (2/n) * sum(error * x)

   # Update the intercept and coefficient using gradient descent
    intercept = intercept - learning_rate * d_intercept
    coefficient = coefficient - learning_rate * d_coefficient
```

```
# Multiple Linear Regression

# Initialize the intercept and coefficients
intercept = 0
coefficients = array(0, dim = length(x[1]))

# Set the learning rate and number of iterations
learning_rate = 0.01
num_iterations = 1000

# Calculate the mean of x and y
x_mean = apply(x, 2, mean)
y_mean = mean(y)

# Calculate the total number of training examples
n = length(x)

# Train the model using gradient descent
```

```
for i in 1:num_iterations:
    # Calculate the predicted output for each training example
    y_pred = intercept + sum(coefficients * x, axis = 1)

    # Calculate the error between the predicted output and the true label
    error = y_pred - y

    # Calculate the partial derivative of the cost function with respect to the intercept
    d_intercept = (2/n) * sum(error)

    # Calculate the partial derivative of the cost function with respect to each coefficient
    d_coefficients = (2/n) * t(x) %*% error

    # Update the intercept and coefficients using gradient descent
    intercept = intercept - learning_rate * d_intercept
    coefficients = coefficients - learning_rate * d_coefficients
```

3. Decision Tree

A decision tree is a tree-based supervised learning algorithm used for both classification and regression tasks. In this algorithm, the input data is partitioned recursively into subsets based on the values of input features, using a tree structure to represent a sequence of decisions and their possible consequences. The tree structure of a decision tree consists of internal nodes and leaf nodes. Internal nodes correspond to features, and each edge leading from an internal node represents a possible value of that feature (Fig 5). Leaf nodes represent the predicted output value for the input data that has reached that node. To build a decision tree, the algorithm selects the feature that best splits the input data into subsets with the most homogeneous target variable (in case of classification, the subsets with the most homogenous class labels, and in regression, the subsets with the least variance in the target variable). The algorithm then recursively applies the same process to each subset, until a stopping criterion is reached, such as a maximum tree depth or a minimum number of samples per leaf (Quinlan, 1986; Kotsiantis et al., 2013). The resulting decision tree can be used for prediction on new data by following the tree from the root to a leaf node, where the predicted output value is the value associated with the leaf node. Decision trees are intuitive and easy to interpret, and they can be used for feature selection and data exploration.

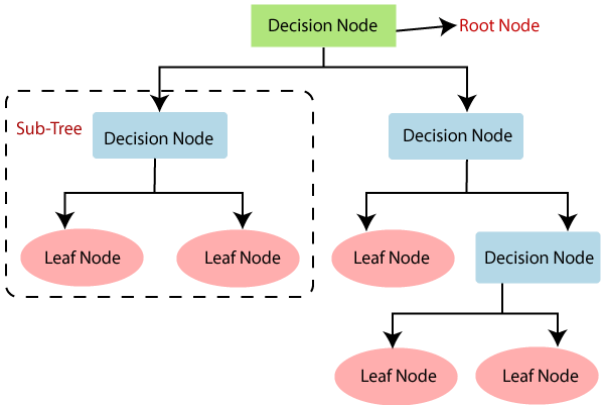


Fig 5. Decision tree structure

Decision Tree Pseudo Code:

```
function decision_tree(data):
    if stopping_condition(data):
        return create_leaf(data)  # create a leaf node

    split = find_best_split(data)  # find the best split point

    left_child_data = subset_data(data, split)  # subset data for left child
    right_child_data = subset_data(data, ~split)  # subset data for right child

    left_child = decision_tree(left_child_data)  # recursively build left child
    right_child = decision_tree(right_child_data)  # recursively build right child

    return create_decision_node(split, left_child, right_child)  # create decision node
```

where,

data is the input data to build a decision tree

stopping_condition is a function that checks if a node should be a leaf node, based on some criterion (e.g. maximum depth, minimum number of samples per leaf)

create_leaf is a function that creates a leaf node containing a prediction value for the data

find_best_split is a function that finds the best split point for the data based on some criterion (e.g. information gain, Gini impurity)

subset_data is a function that subsets the data based on a split point

create_decision_node is a function that creates a decision node containing the split point and references to its left and right child nodes.

4. Random Forest

Random Forest is an ensemble learning algorithm used for classification, regression, and other tasks. It creates multiple decision trees based on random subsets of the training data and features and aggregates their predictions to make a final prediction (Fig 6). Random Forest has several advantages over individual decision trees, such as improved accuracy and robustness to overfitting (Breiman, 2001; Belgiu and Drăguţ, 2016). It can also handle high-dimensional data and noisy or incomplete datasets. However, it can be slower and require more computational resources than simpler models.

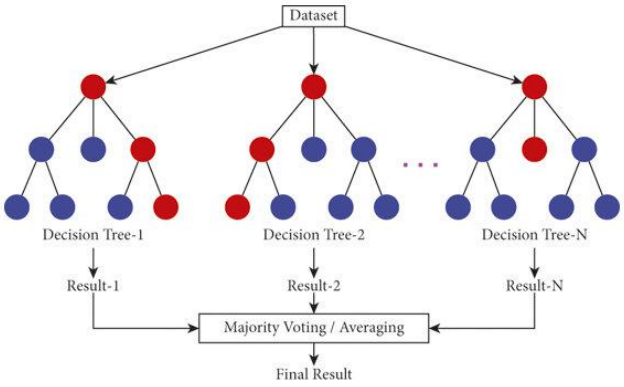


Fig 6. Random forest structure considering multiple trees.

Random Forest Pseudo Code:

Input:
- A set of training examples D = {(x1, y1), (x2, y2), ..., (xn, yn)}, where xi is a feature vector and yi is the corresponding class label.
- The number of decision trees T to create.
- The maximum depth or number of nodes for each decision tree.
- The number of features to consider at each split point in each tree.

Training:
For t = 1 to T:
- Randomly select a subset of the training data (with replacement) to create a new "bootstrap" sample.
- Randomly select a subset of the features of size K to consider at each split point in the decision tree.
- Build a decision tree using the bootstrap sample and the selected features, by recursively splitting nodes based on the feature values that maximize information gain or another criterion, until reaching a maximum depth or number of nodes.
- Add the decision tree to the forest.

Return the forest of T decision trees.

Classification:
For a new input example x:

- Pass it through each decision tree in the forest.
- Aggregate the resulting predictions (e.g., by majority voting for classification or averaging for regression).

Return the final prediction.

Output:

- The predicted class label or value for the new input example x.

5. Naïve Bayes

Naïve Bayes is a probabilistic machine learning algorithm used for classification and prediction. It is based on Bayes' theorem, which describes the probability of an event based on prior knowledge and evidence.

$$P(y|x) = \frac{P(x|y) * P(y)}{P(x)}$$

The Naïve Bayes algorithm assumes that the features in the input data are independent of each other, given the class label. This is a simplifying assumption that allows for efficient computation and often works well in practice, especially for text classification and spam filtering. Naïve Bayes is a simple yet powerful algorithm that can work well on many types of datasets, especially when the independence assumption holds or when there are many features (Frank et al., 2000; Murphy, 2006). However, it may suffer from the "zero probability problem" when a feature is not present in the training data for a given class, and it may not perform as well as more complex models on some datasets.

Naïve Bayes Pseudo Code:

- Input: A set of training examples D = {(x1, y1), (x2, y2), ..., (xn, yn)}, where xi is a feature vector and yi is the corresponding class label.
- Preprocessing: Convert the feature vectors xi into discrete or categorical values, if necessary.
- Training: Calculate the prior probability P(y) for each class label y by counting the number of training examples in each class and dividing by the total number of examples. Calculate the conditional probability P(xi|y) for each feature xi and class label y by counting the number of examples with xi and y and dividing by the total number of examples with y.
- Prediction: Given a new input example x, calculate the posterior probability P(y|x) for each class label y by applying Bayes' theorem in equation 6, where P(x|y) is the product of the conditional probabilities P(xi|y) for each feature xi in x, and P(x) is a normalization constant that ensures that the probabilities sum to 1.
- Classification: Assign the class label y with the highest posterior probability as the predicted class for the new example x.
- Evaluation: Measure the accuracy, precision, recall, F1 score, or other metrics on a separate validation or test set to assess the performance of the Naïve Bayes model.

6. K-nearest neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple and intuitive machine learning algorithm used for both classification and regression tasks. It is a non-parametric and instance-based learning algorithm, which means that it doesn't make any assumptions about the underlying data distribution and the model is not built using a training set of data. Instead, KNN classifies new data points by looking at the k closest labeled data points in the training set and using their labels to predict the class of the new data point (Bezdek et al., 1986; Zhang, 2016). Below is the step-by-step framework of KNN algorithm:

- Collect and preprocess data: The first step is to collect the dataset and preprocess it by cleaning the data, handling missing values, and transforming the data if needed.
- Choose the number of neighbors (k): The user needs to choose the number of neighbors (k) to consider when classifying a new data point. Typically, the value of k is an odd number to avoid ties in the class assignments.
- Calculate distances: For each data point in the test set, calculate the Euclidean distances between the test point and all data points in the training set.
- Select k-nearest neighbors: Identify the k-nearest neighbors of the test data point based on the calculated distances. These are the k training data points that are closest to the test data point.
- Assign class labels: Based on the k-nearest neighbors, assign a class label to the test data point. In classification, this can be done by majority vote, where the class label that occurs most frequently among the k neighbors is assigned to the test point. In regression, the output is the average of the k nearest neighbors' labels.
- Evaluate the model: Once the model has been trained and tested, evaluate its performance using metrics such as accuracy, precision, recall, and F1 score.

KNN Pseudo Code:

```
Function KNN (train_data, test_data, K):
  For each test point in test_data:
    1. Calculate the distance between the test point and each point in train_data.
    2. Sort the distances in ascending order.
    3. Select the top K data points with the smallest distances.
    4. Determine the class of the test point based on the most common class among the K
       neighbors.
    5. Return the class of the test point.
```

7. Support vector machine (SVM)

Support Vector Machine (SVM) is a popular machine learning algorithm used for classification and regression tasks. In high or infinite-dimensional space, a support vector machine constructs a

hyperplane or a set of hyperplanes that best separates the data into different classes. The goal of the SVM algorithm is to find the decision boundary that maximizes the margin between the two classes. The margin is defined as the distance between the decision boundary and the closest data points from each class. SVM selects the decision boundary that maximizes this distance, which results in a better generalization performance on new, unseen data (Hearst et al., 1998; Pisner and Schnyer, 2020). Intuitively, the greater the margin, the lower the classifier's generalization error. SVM is computationally efficient, and it can be used with different kernel functions, such as, linear, polynomial, radial basis function (RBF), etc. to transform the data into a higher-dimensional feature space.

SVM Pseudo Code:

```
Input: Training set {(x1, y1), (x2, y2), ..., (xn, yn)}
       Regularization parameter C
       Kernel function K(x_i, x_j)
       Tolerance for convergence tol

Output: Support vectors, alpha values and bias term b

Step 1: Initialize alpha and b to zero
Step 2: Repeat until convergence or maximum iterations:
       a. For each pair of alpha values (alpha_i, alpha_j):
          i. Compute the error E_i and E_j between predicted class and true class
          ii. Check if (alpha_i, alpha_j) violate KKT conditions
          iii. If the conditions are violated, update alpha_i and alpha_j values
       b. Compute the bias term b
Step 3: Compute the decision boundary using support vectors
Step 4: Make predictions on new, unseen data
```

8. K-means clustering

K-means is a clustering algorithm used to partition data into K clusters based on their similarity. It is an unsupervised machine learning algorithm that is widely used in various applications, including image segmentation, customer segmentation, and anomaly detection. It does not require any labeled data for training, instead it tries to find patterns or structure in the data by grouping similar data points together (Hartigan and Wong, 1979; Likas et al., 2003; Kodinariya and Makwana, 2013).

The first step of the algorithm is to randomly initialize K cluster centers or centroids. These centroids are randomly selected from the data points or using some other heuristic method. Each data point is then assigned to the nearest centroid based on some similarity measure, such as Euclidean distance. The distance between a data point and a centroid can be calculated as the square of the Euclidean distance or some other distance metric. After assigning all data points to their nearest centroids, the centroids are recalculated as the mean of all data points assigned to that

centroid. In other words, the centroid is shifted towards the center of its assigned data points. The previous two steps of assigning each data point to the nearest centroid and updating centroid is repeated until the centroids no longer change or the maximum number of iterations is reached. The algorithm terminates when there is no change in the centroids or when the maximum number of iterations is reached. Once the algorithm has converged, the final output is the K clusters, with each cluster containing a set of data points that are similar to each other (Fig 7.)

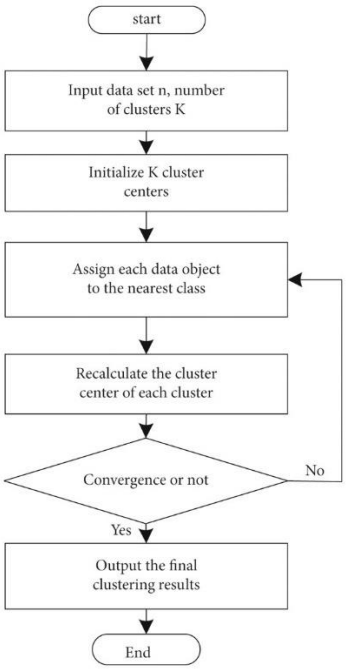


Fig 7. K-means clustering flowchart

K-means clustering Pseudo Code:

```
Algorithm kMeans(data, k, max_iterations)
    Initialize k random centroids
    for i in range(max_iterations):
        Create k empty clusters
        for each data point in data:
            Calculate the distance between the data point and each centroid
            Assign the data point to the cluster with the closest centroid
        for each cluster:
            Calculate the mean of all data points in the cluster
            Update the centroid of the cluster with the mean
    return the final clusters and their centroids
```

9. Feature selection

Feature selection is the process of selecting a subset of relevant features or variables from a larger set of features to improve the performance of a model. It is an essential step in machine learning

and data analysis, as it reduces the number of input features and focuses on the most important ones that can provide meaningful insights and improve the accuracy of the model. The objective of feature selection is to eliminate irrelevant, redundant, or noisy features, which can cause overfitting, reduce model interpretability, and increase computational complexity. By reducing the dimensionality of the feature space, feature selection can also improve model performance and speed up the training and testing process (Li et al., 2017).

There are different types of feature selection techniques, including filter methods, wrapper methods, and embedded methods. Filter methods use statistical or correlation-based measures to rank the features and select the top ones, while wrapper methods evaluate the performance of the model with different subsets of features and select the best one based on a validation set. Embedded methods incorporate feature selection as part of the model training process, such as regularization or decision tree-based algorithm. The choice of feature selection technique depends on the type and size of the dataset, the complexity of the model, and the specific problem being addressed. Effective feature selection can lead to more accurate, interpretable, and efficient models, and is an essential component of many data analysis and machine learning workflows (Chandrashekar and Sahin, 2014).

10. Feature extraction

Feature extraction is the process of transforming raw data into a set of meaningful and informative features that can be used for machine learning and data analysis tasks. It involves selecting or designing a set of features that capture the essential characteristics or patterns of the data, and transforming the raw input into a vector or matrix representation that can be used by the model. The aim of feature extraction is to reduce the dimensionality of the input data, enhance the signal-to-noise ratio, and improve the performance and interpretability of the model (Levine, 1969; Lee, 1993). It is especially important when dealing with high-dimensional or complex data, such as images, text, or sensor data, where the raw input may contain a large number of irrelevant or redundant features. For instance, principal component analysis (PCA) is often used as a dimensionality-reduction technique to extract a lower-dimensional space creating new brand components from the existing features in a dataset (Khalid et al., 2014).

11. Principal component analysis (PCA)

Principal Component Analysis (PCA) is a well-known unsupervised learning approach in the field of machine learning and data science. This technique that can be used to transform a high-dimensional dataset into a lower-dimensional space while preserving the most important patterns or structure in the data. In particular, PCA is a mathematical technique that identifies a new set of orthogonal (uncorrelated) variables, called principal components, which capture the maximum amount of variance in the data (Pearson, 1901; Hotelling, 1933). Thus, PCA can be used as a feature extraction technique that reduces the dimensionality of the datasets to build an effective machine learning model. Technically, PCA identifies the completely transformed with the highest eigenvalues of a covariance matrix and then uses those to project the data into a new subspace of equal or fewer dimensions (Abdi and Williams, 2010; Vidal et al., 2016).

PCA Pseudo Code:

Inputs:
- X: n x p matrix of input data

Output:
- Y: n x k matrix of transformed data
- V: p x k matrix of eigenvectors
- S: k x k matrix of eigenvalues

1. Center the input matrix X by subtracting the mean of each column.
2. Compute the covariance matrix C = X^T X / (n-1).
3. Compute the eigenvectors and eigenvalues of the covariance matrix using a method such as singular value decomposition (SVD).
4. Sort the eigenvectors in descending order by their corresponding eigenvalues.
5. Select the top k eigenvectors to use as the new basis for the transformed data, where k is the desired number of dimensions.
6. Compute the transformed data Y = X V, where V is the matrix of selected eigenvectors.
7. Compute the eigenvalues of the selected eigenvectors to obtain the variance explained by each component.
8. Return the matrix Y as the transformed data, and the matrices V and S as the eigenvectors and eigenvalues, respectively.

Data and Methodology

The present analysis first identifies the major sectors of the United States (U.S) industry, which experience simultaneous exports, imports and reexports. The major product groups include mineral fuels, base metals, pharmaceuticals and organic chemicals, plastics products, vehicles and auto-components, aircraft, machinery and electrical equipment, textile and garments, articles of wood and paper, precious metals, optical instruments, animal and vegetable products. The selected sectors collectively account for more than 85 per cent of U.S. economy's aggregate export and import flows (Agarwal and Chakraborty, 2017; Aggarwal and Chakraborty; 2020b). The trade data (export, import and reexport) for the purpose of the analysis is drawn in Harmonized System (HS) classification at HS 2-digit level (i.e., tariff headings) for the period 2002-2021 from Trade Map database, maintained by International Trade Centre (ITC, undated).

Second, it is observed that at the composite level, U.S. exports and imports has witnessed an increasing trend over the past two decades. United nations trade flows have registered fairly strong growth over 2002-2008, accompanied by rising commodity prices. Post financial crises in 2008; trade fell steeply before rebounding strongly during 2010-2011 (WTO, 2015; Aggarwal, 2020). After witnessing moderate growth over 2012-2014, an economic downturn is observed in 2015-2016, and thereafter a strong rebound is marked in 2017-2018 (UNCTAD, 2019; Aggarwal, 2020). Following the COVID-19 crises, a major slowdown impacted the U.S economy drastically in 2020 (UNCTAD, 2022), finally a remarkable progress in U.S. trade with world economy has been observed in 2021. Fig. 8 reports the year-wise trend of U.S. trade during the period 2002-2021. Exports has witnessed high growth in its value from US $693 billion to US$1753 billion, imports

have been marked with ever higher growth in its value from US $1200 billion to US $2937 billion, whereas reexports has increased from US $63 billion to US $274 billion respectively in 2002 and 2021.

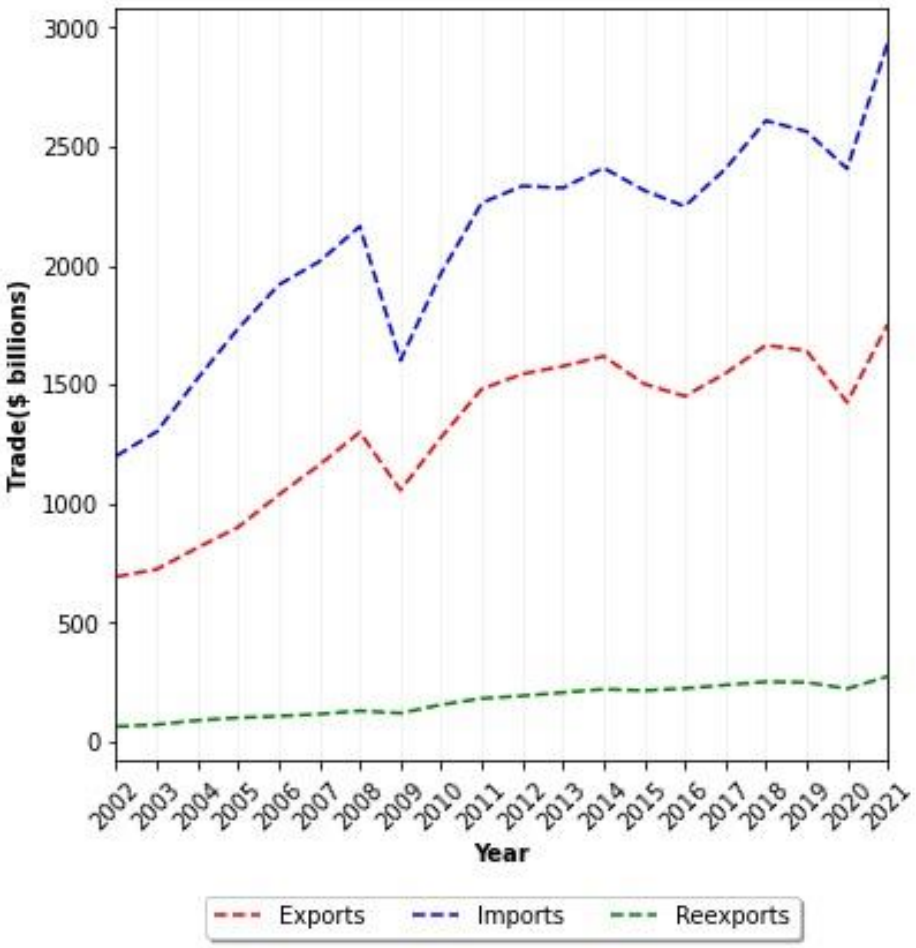


Fig 8. United States trade with ROW (2002-2021)

Source: Author's own construction from Trade Map (ITC, undated) data

Third, the evolving shares of the thirteen selected product categories, in Unites States export and import basket are reported in Table 2. The corresponding HS two-digit codes under each product category is reported in the second column of the table. For observing the temporal perspective, their average shares are compared over the periods 2002-2006, 2007-2011, 2012-2016 and 2017-2021, respectively. It is observed that in the export basket, the proportional shares of these product groups are 88 percent across the study (2002-2021) with minimal fluctuations. The corresponding numbers on the import front is 85 percent with similar trend throughout the period (2002-2021). Interestingly, among the selected sectors, it has been noted that the import share of mineral fuels, vehicles and auto-components, machinery and electrical equipment exceeds the share of exports in the total trade basket. This can be attributed to U.S. economy's increasing reliance on Persian Gulf countries, such as, Saudi Arabia for mineral fuels and oils consumption; Germany and Korea for imports demands of automobiles and vehicles (e.g., BMW, Mercedes); and China, Mexico and

Canada for supplying them the machinery and electrical equipment's to assemble electronic products in the U.S. market (e.g., iPhone parts and components), which is also exported back to the rest of the world. Thus, international production fragmentation and vertical disintegration of manufacturing stages by transnational corporations have improved industrial capabilities, thereby creating scope for vertical specialization in finished goods (Aggarwal and Chakraborty, 2021). The regionalization drive over the last two decades and growing cross-border intra-firm trade have produced interesting dynamics within global value chains (GVCs) which are crucial in supplying cost-competitive finished products to the global market (WTO, 2011; Mudambi and Venzin, 2010; Lanz and Miroudot, 2011).

Table 2. Importance of the selected commodities in U.S. trade basket

Sector	HS Codes	Average Export Share (%)				Average Import Share (%)			
		2002-2006	2007-2011	2012-2016	2017-2021	2002-2006	2007-2011	2012-2016	2017-2021
Animal, Vegetable Products and Prepared Foodstuffs	01-24	8.06	9.55	10.04	10.02	5.03	5.46	6.19	6.87
Mineral Fuels	27	2.46	5.98	8.28	11.40	14.40	19.60	13.21	7.80
Chemicals	28-29	4.17	4.10	3.50	3.14	3.21	3.31	2.90	2.61
Pharmaceuticals	30	2.27	2.95	2.84	3.45	2.06	2.89	3.27	4.87
Plastics	39	4.09	4.05	3.93	4.08	1.87	1.82	2.08	2.48
Wood and Paper Articles	44-49	3.26	2.88	2.53	2.33	3.13	2.05	1.82	1.95
Textiles	50-63	2.59	1.83	1.66	1.59	5.98	5.01	4.88	4.64
Precious stones	71	2.35	3.98	4.21	4.05	2.20	2.52	2.74	2.96
Base Metals	72-83	4.58	5.41	4.84	4.27	5.17	5.38	5.23	5.34
Machinery and Electrical Equipment	84-85	32.61	27.02	24.48	23.51	25.78	25.46	27.44	28.91
Vehicles	87	9.08	8.12	8.51	7.76	12.79	9.37	11.46	11.25
Aircrafts	88	6.41	6.41	7.96	7.19	1.14	1.03	1.33	1.19
Optical Instruments	90	6.04	5.69	5.44	5.48	2.87	2.92	3.24	3.66
Total		87.97	87.98	88.21	88.27	85.64	86.83	85.78	84.52

Source: Author's estimation

Fourth, the input data is standardized (z-score normalized) using 'scikit-learn' Python library before applying a machine learning algorithm. To standardize the variables, the mean of each variable is subtracted from each observation and then divided by the standard deviation of that variable (Annexure 1). This centers the variables around zero and scales them to have unit variance. The standardized variables can be represented as:

$$Z = \frac{(X - mean(X))}{std\ dev(X)}$$

where X is the original variable, mean(X) is the mean of X, and std dev(X) is the standard deviation of X. The resulting scores have a mean of 0 and a standard deviation of 1.

Finally, K-means clustering, a popular unsupervised machine learning algorithm discussed in the previous section has been deployed in the current analysis for determining the accurate clusters (i.e., industries with similar structure of exports, imports, and investment) for the period 2002, 2011 and 2021. K-means clustering is a distance-based algorithm, which calculates the distance between data points to determine cluster membership. After standardization, the K-means algorithm is applied to the standardized data, and the resulting clusters are then interpreted based

on the original data. Standardization ensures that each variable contributes equally to the distance calculation, and the resulting clusters are not biased towards variables with larger scales or variances.

The first step of the algorithm is to randomly initialize K cluster centers or centroids. Therefore, four clusters have been identified given the structure of the industries in the United States, namely, LL, HL, LH and HH. The interpretation of these clusters is as follows:

- LL represents the product groups with low exports, imports and reexports, classified as low investment industry.
- HL illustrates the product groups with medium exports, imports, reexports, and standardized export to standardized import ratio > 1, classified as category 1 medium investment industry.
- LH illustrates the product groups with high imports, medium exports and reexports, and standardized export to standardized import ratio < 1, classified as category 2 medium investment industry.
- HH represents the product groups with high exports, imports and reexports, classified as high investment industry.

Next, each data point is then assigned to the nearest centroid. The Euclidean distance between a data point and a centroid is calculated. The previous two steps of data assignment and centroid update is repeated until the centroids converges. This algorithm has been executed in python using 'scikit-learn' library. The python program for K-means algorithm is reported in Annexure 2. The final clustering results are discussed in the next section.

Results

In the K-means clustering algorithm, the iteration process continue until the result of the last cluster value is equal to the previous cluster value. This is noted when the value of the centroid converges for each variable used in the analysis. In the current study, the K-means algorithm is executed for the period 2002, 2011 and 2021 to segment similar industries in the past two decades for the selected period. The value of a final cluster centroid for each representative cluster identified earlier is reported in Table 3.

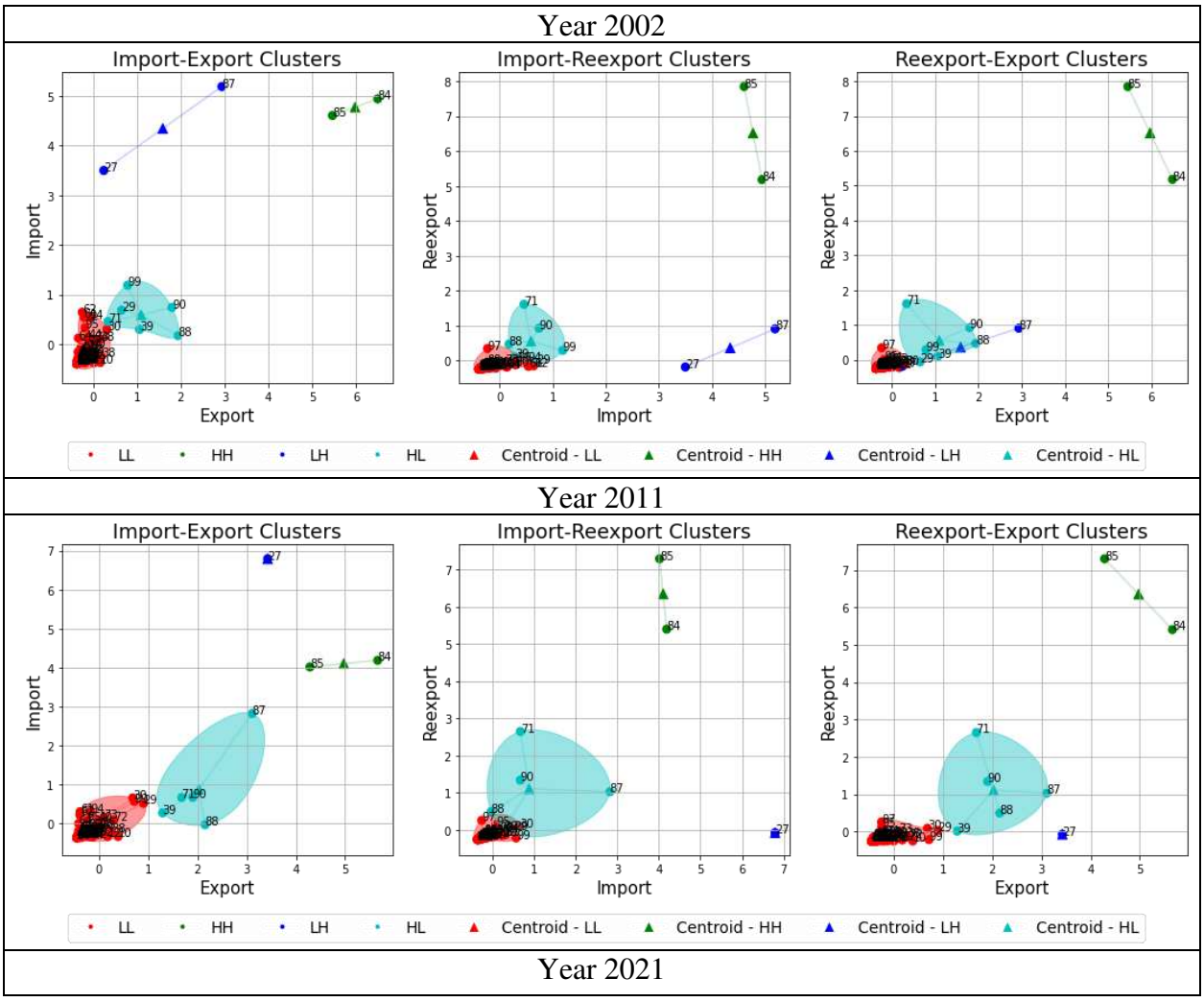
Table 3. Calculation of the cluster centroid in the last iteration of K-means clustering algorithm

Year	2002		
Clusters	Exports	Imports	Reexports
LL	-0.249	-0.249	-0.196
HL	1.092	0.589	0.554
LH	1.588	4.341	0.362
HH	5.976	4.770	6.517
Year	2011		
Clusters	Exports	Imports	Reexports
LL	-0.264	-0.218	-0.203
HL	2.028	0.886	1.103
LH	3.425	6.791	-0.077

HH	4.973	4.103	6.351
Year	2021		
Clusters	Exports	Imports	Reexports
LL	-0.291	-0.258	-0.215
HL	1.488	0.760	0.939
LH	3.672	3.716	0.304
HH	4.524	5.246	6.235

Source: Author's estimation

K-means cluster diagram typically illustrates the data points in a scatter plot with different markers representing the assigned clusters for each point. The cluster centroids (Δ) for each representative clusters are also indicated in the plot. The pair-wise representation of k-means clusters for U.S. exports, imports and reexports in 2002, 2011 and 2021 is reported in Fig. 9.



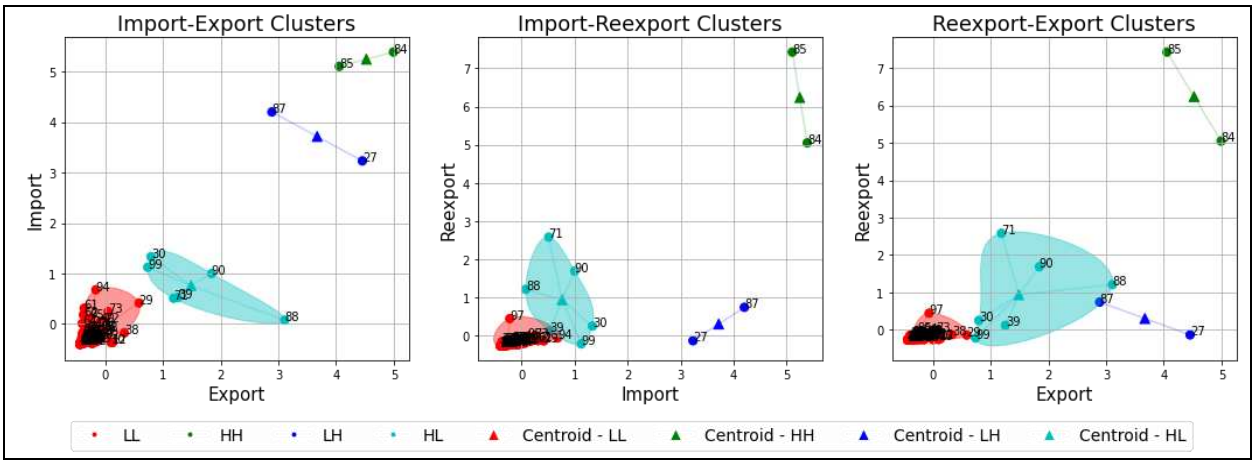


Fig 9. Pair-wise representation of K-means clusters for U.S. exports, imports and reexports

Source: Author's construction

The empirical analysis shows the result of K-means clustering algorithm illustrated through the formation of clusters where similar sectors are grouped under one cluster (Fig. 9). Several conclusions emerge from the empirical results.

Firstly, it is observed that the structure of select sectors, such as, animal and vegetable products, inorganic chemicals, wood and paper articles, textiles and base metals has been similar across the past two decades. These sectors are characterized with low exports, imports and reexports in the select time-period (2002, 2011 and 2021). For less developed industries to advance in the technology ladder, rising skill in the sector needs to be complemented with innovation orientation and necessary investment for boosting the firms to graduate to a higher quality plane (Aggarwal and Chakraborty, 2020a, 2020c). This will further lead to rise in domestic production, exports and reexports. Lower imports increases the trade balance on one hand and reduces the scope of technology transfer essential for efficient production mechanism on the other. For these reasons, it is difficult for these sectors to move upward in the technology ladder and, therefore, are classified as low investment industries (red-colored clusters in Fig 9). This poses a potential threat for the sectors lying on a lower quality plane, many of which are also characterized by worsening industrial efficiency.

Secondly, machinery and electrical equipment sector has been marked with high exports, imports and reexports in the selected time-period. This sector signifies that product differentiation in machinery and electrical parts and components, complemented with external economies of scale due to high-capital intensive nature of these industries, may lead to efficient production mechanism for an expanding export market and increase in trade growth with the rest of world (Aggarwal and Chakraborty, 2019). Thus, machinery and electrical equipment sector (HS 84-85) is classified as high investment industries (dark-green colored clusters in Fig 9).

Thirdly, plastics, precious stones, aircrafts, and optical instruments has been marked with medium-type exports, imports and reexports in the selected time-period. Further, it has been noted that the standardized export to standardized import ratio is generally > 1 in the select sectors. Such industries focus on export promotion through industrial restructuring process which enhances their

efficiency and promotes both exports and imports for long-term growth of the economy. Therefore, these industries have been classified as medium investment industries with high potential of growth.

Fourthly, mineral fuels have been marked with high imports, medium exports and reexports in the selected time-period. It has been further noted that the standardized export to standardized import ratio is generally < 1 in this sector. This can be attributed to United States increased reliance on Persian Gulf countries for mineral fuels and oils consumption. This has propelled the growth of U.S. energy sector on one hand and had worsened the trade deficit on the other. Such investments on the first place are necessary for boosting the firms to graduate to a higher quality plane in the long run. However, such indirect impact of growth on other sectors through rising import bills is not sustainable. So, after a certain threshold time-period and investment, efficient industries are capable of undertaking R&D expenditures to periodically go for product innovation and mass production of previously imported product-groups, to reduce its dependence on imports (Aggarwal and Chakraborty, 2022). Thus, mineral fuels (HS-27) are classified as medium investment industry with increased reliance on imports.

Lastly, some sectors have demonstrated the interesting dynamics and changed their cluster membership over the past two decades. First, pharmaceuticals sector was categorized as a low-investment industry in 2002 and 2011. However, it has been categorized as a medium investment type industry in 2021, with a high potential of long-term growth. This can be attributed to the COVID-19 pandemic that positively impacted this market by the end of 2020 with increased sale of drugs and vaccines, R&D investment, and pipeline portfolio of the major players in the market. The pipeline portfolio for infectious diseases and vaccines witnessed a significant rise driven by COVID-19 related to treatments and immunization programs. Second, organic chemicals were categorized as a medium investment industry in 2002 has now been degraded to low investment type industry. Third, vehicles and automobiles sectors were classified as medium investment industry with increased reliance on imports in 2002. This sector witnessed a growth in 2011 and changed the cluster membership to medium investment industry with high potential of growth. However, the sector has again went back to its former position of increased reliance on imports under medium investment industry in 2021. This can be explained from the fact that United States have increased their imports of automobiles parts and components from foreign nations (e.g., Germany, Korea, etc.) in the recent times, adding financial pressure on the trade account deficit. The cluster membership across the product categories for the period 2002, 2011 and 2021 is reported in Table 4.

Table 4. Cluster membership across select product categories.

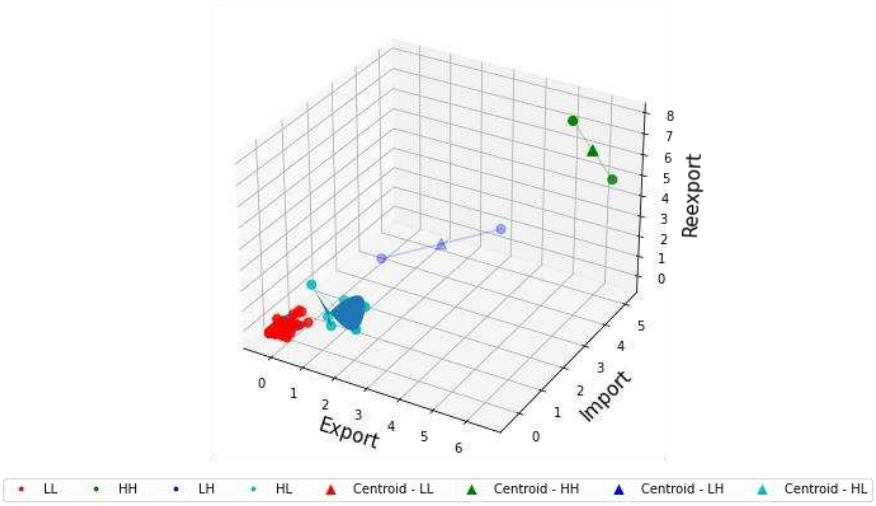
Sector	**HS Codes**	**Cluster Membership**		
		2002	**2011**	**2021**
Animal, Vegetable Products and Prepared Foodstuffs	01-24	LL	LL	LL
Mineral Fuels	27	LH	LH	LH
Inorganic chemicals	28	LL	LL	LL
Organic chemicals	29	HL	LL	LL

Pharmaceuticals	30	LL	LL	HL
Plastics	39	HL	HL	HL
Wood and Paper Articles	44-49	LL	LL	LL
Textiles	50-63	LL	LL	LL
Precious stones	71	HL	HL	HL
Base Metals	72-83	LL	LL	LL
Machinery and Electrical Equipment	84-85	HH	HH	HH
Vehicles	87	LH	HL	LH
Aircrafts	88	HL	HL	HL
Optical Instruments	90	HL	HL	HL

Source: Author's estimation

Once the initial clusters have been obtained from the K-means clustering algorithm, the convex hull of each cluster can also be derived. The convex hull may be defined either as the intersection of all convex sets containing a given subset of a Euclidean space, or equivalently as the set of all convex combinations of points in the subset. The 3D representation of the convex hulls representing each of the four clusters for the select time-period (2002, 2011 and 2021) is reported in Fig 10.

Year 2002



Year 2011

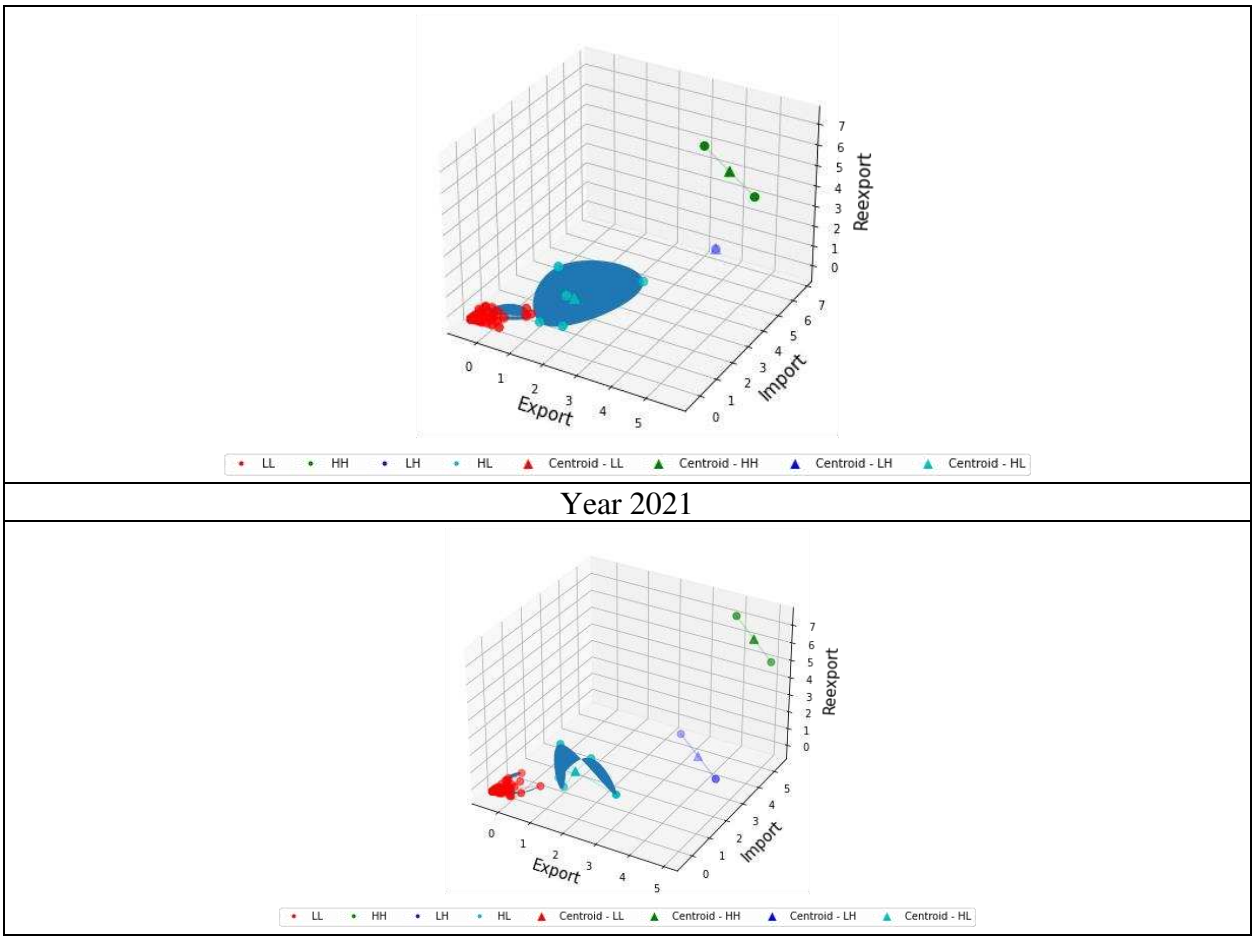


Fig 10. 3D representation of K-means clusters for U.S. exports, imports and reexports

Source: Author's construction

Conclusion

The results of the research carried out state that the K-means clustering method is not limited to a particular domain, rather it can be applied and implemented to international trade and economic data also that may impact the government policy-making process at the global level.

Several conclusions emerge from the empirical results. First, animal and vegetable products, inorganic chemicals, wood and paper articles, textiles and base metals have been categorized as low investment industries. Second, machinery and electrical equipment sector is classified as high investment industries. Third, plastics, precious stones, aircrafts, and optical instruments have been categorized as medium investment industries with significant potential of growth. Fourth, mineral fuels are classified as medium investment industry with increased reliance on imports. The industries listed in the above four categories have not changed their cluster membership and have been associated with the same cluster throughout the analysis period (2002-2021). Therefore, K-means clustering algorithm has successfully formed clusters based on data-points and grouped similar industries into one cluster.

Interestingly, some sectors have changed their cluster membership which are as follows. First, pharmaceuticals sector was initially categorized as a low-investment industry has now been uplifted to medium investment type industry with a high potential of growth. Second, organic chemicals which once categorized as a medium investment industry in 2002 has now been downgraded to low investment type industry. Third, vehicles and automobiles sector were uplifted to medium investment industry with high potential of growth in 2011. However, the growth did not last longer and the sector went back to its former position of increased reliance on imports under medium investment industry in 2021.

Among all the sectors of U.S. economy analyzed in this research, it has been noted that machinery and electrical equipment sector is the most advanced and developed sector. This sector is capital intensive in nature and has already graduated to a higher quality plane. It is the only sector where reexports are also higher, thus, contributing significantly to industrial growth rate of the U.S. economy. Among the other sectors, aircrafts and optical instruments have shown the consistent growth rate in the past two decades. These sectors have not only remained in their clusters of medium type investment with high potential to growth but have also witnessed the improvement in the ratio of standardized exports to standardized imports over the years, signifying latest development in the aircrafts and optical instruments industries that boosted these industries to graduate to a higher quality plane. The policymakers, therefore, need to facilitate skill-augmentation and productivity rise as well as technology transfer through the appropriate policy measures and other initiatives especially in the sectors characterized by low and medium investment. Only then the necessary export impetus, upstream domestic value chain integration and rise in domestic competitiveness can be generated in the low investment industries.

Future Scope

Future research may focus on introduction of more trade-specific variables in the empirical model for explaining similarity of industries in major manufacturing product groups. Also, the analysis

can be extended to identify the similar structure of industries for other advanced economies as well.

Appendix

Annexure 1.

```
from sklearn import preprocessing, decomposition
data_scaled = sklearn.preprocessing.scale(df1.values)
df1['scaled1'] = data_scaled[:,0]
df1['scaled2'] = data_scaled[:,1]
df1['scaled3'] = data_scaled[:,2]
```

Source: Author's construction

Annexure 2.

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4,random_state=0).fit(data_scaled)
print(kmeans.labels_)
df1['predicted'] = kmeans.labels_

targets = [0,1,2,3]
colors = ['r', 'g', 'b','c']
df1['c'] = df1.predicted.map({0:colors[0], 1:colors[1], 2:colors[2], 3:colors[3]})
df1['category'] = df1.predicted.map({0:'LL', 1:'HH', 2:'LH', 3:'HL'})
label_dict = {0:'LL', 1:'HH', 2:'LH', 3:'HL'}
centroids  = kmeans.cluster_centers_
cen_x = [i[0] for i in centroids]
cen_y = [i[1] for i in centroids]
cen_z = [i[2] for i in centroids]
## add to df
df1['cen_x'] = df1.predicted.map({0:cen_x[0], 1:cen_x[1], 2:cen_x[2], 3:cen_x[3]})
df1['cen_y'] = df1.predicted.map({0:cen_y[0], 1:cen_y[1], 2:cen_y[2], 3:cen_y[3]})
df1['cen_z'] = df1.predicted.map({0:cen_z[0], 1:cen_z[1], 2:cen_z[2], 3:cen_z[3]})
```

Source: Author's construction

References

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4), 433-459.

Agarwal, A., Dawson, S., McKee, D., Eugster, P., Tancreti, M., & Sundaram, V. (2017, April). Detecting abnormalities in IoT program executions through control-flow-based features. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation* (pp. 339-340).

Aggarwal, S. (2016). Determinants of money demand for India in presence of structural break: An empirical analysis. *Business and Economic Horizons* (BEH), Prague Development Center (PRADEC), 12(4), 173-177.

Aggarwal, S. (2020). Determinants of Intra-Industry Trade and Labour Market Adjustment: A Sectoral Analysis for India (Doctoral dissertation, Indian Institute of Foreign Trade).

Aggarwal, S., & Chakraborty, D. (2017). Determinants of India's bilateral intra-industry trade over 2001–2015: Empirical results. *South Asia Economic Journal*, 18(2), 296–313.

Aggarwal, S. (2017). Smile curve and its linkages with global value chains. *Journal of Economic Bibliography*, 4(3).

Aggarwal, S., & Chakraborty, D. (2019). Which factors influence India's intra-industry trade? Empirical findings for select sectors. *Global Business Review*. Retrieved from https://journals.sagepub.com/doi/10.1177/0972150919868343 (Accessed on April 23, 2020).

Aggarwal, S., & Chakraborty, D. (2020a). Labour market adjustment and intra-industry trade: Empirical results from Indian manufacturing sectors. *Journal of South Asian Development*, 15(2), 238-269.

Aggarwal, S., & Chakraborty, D. (2020b). Determinants of vertical intra-industry trade: Empirical evidence from Indian manufacturing sectors. *Prajnan: Journal of Social and Management Sciences*, 49(3), 221-252.

Aggarwal, S., & Chakraborty, D. (2020c). Is there any relationship between Marginal Intra-Industry Trade and Employment Change? Evidence from Indian Industries. Working Paper, No. 20-44, Indian Institute of Foreign Trade, Delhi.

Aggarwal, S., Chakraborty, D., & Bhattacharyya, R. (2021). Determinants of Domestic Value Added in Exports: Empirical Evidence from India's Manufacturing Sectors. *Global Business Review*. https://doi.org/10.1177/09721509211050138.

Aggarwal, S., & Chakraborty, D. (2021). Which factors influence vertical intra-industry trade in India?: Empirical results from panel data analysis. Working Paper, No. 21-04, Indian Institute of Foreign Trade, Delhi.

Aggarwal, S., Chakraborty, D. (2022). Which Factors Influence India's Bilateral Intra-Industry Trade? Cross-Country Empirical Estimates. Working Papers 2260, Indian Institute of Foreign Trade, Delhi.

Aggarwal, S., Mondal, S., & Chakraborty, D. (2022). Efficiency Gain in Indian Manufacturing Sectors: Evidence from Domestic Value Addition in Exports. *Empirical Economics Letters*, 21(2): 69-83.

Aggarwal, S., Chakraborty, D., & Banik, N. (2023). Does Difference in Environmental Standard Influence India's Bilateral IIT Flows? Evidence from GMM Results. *Journal of Emerging Market Finance*, 22(1), 7–30. https://doi.org/10.1177/09726527221088412.

Alakus, T. B., & Turkoglu, I. (2020). Comparison of deep learning approaches to predict COVID-19 infection. *Chaos, Solitons & Fractals*, 140, 110120.

Athukorala, P. C., & Yamashita, N. (2006). Production fragmentation and trade integration: East Asia in a global context. *The North American Journal of Economics and Finance*, 17(3), 233-256.

Baldwin, R. (2013). Trade and industrialization after globalization's second unbundling: How building and joining a supply chain are different and why it matters. *In Globalization in an age of crisis: Multilateral economic cooperation in the twenty-first century* (pp. 165-212). University of Chicago Press.

Baraniuk, R. G. (2011). More is less: Signal processing and the data deluge. *Science*, 331(6018), 717-719.

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114, 24-31.

Bendre, M. R., & Thool, V. R. (2016). Analytics, challenges, and applications in big data environment: a survey. *Journal of Management Analytics*, 3(3), 206-239.

Bevan, A. (2015). The data deluge. *Antiquity*, 89(348), 1473-1484.

Beyer, M. A., & Laney, D. (2012). The importance of 'big data': a definition. *Stamford, CT: Gartner*, 2014-2018.

Bezdek, J. C., Chuah, S. K., & Leep, D. (1986). Generalized k-nearest neighbor rules. Fuzzy Sets and Systems, 18(3), 237-256.

Bishop, C. M. (2006). Pattern recognition and machine learning: springer New York.

Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.

Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big data & society*, 3(1), 2053951715622512.

Buşoniu, L., Babuška, R., & De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, 183-221.

Cao, L. (2017). Data science: a comprehensive overview. *ACM Computing Surveys (CSUR)*, 50(3), 1-42.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16-28.

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., & Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug discovery today*, 23(6), 1241-1250.

Chowdary, M. K., Nguyen, T. N., & Hemanth, D. J. (2021). Deep learning-based facial emotion recognition for human–computer interaction applications. *Neural Computing and Applications*, 1-18.

Cingolani, I., Iapadre, L., & Tajoli, L. (2018). International production networks and the world trade structure. *International Economics*, 153, 11-33.

Clement, J. C., Ponnusamy, V., Sriharipriya, K. C., & Nandakumar, R. (2021). A survey on mathematical, machine learning and deep learning models for COVID-19 transmission and diagnosis. *IEEE reviews in biomedical engineering*, 15, 325-340.

Cummins, N., Baird, A., & Schuller, B. W. (2018). Speech analysis for health: Current state-of-the-art and the increasing impact of deep learning. *Methods*, 151, 41-54.

Dai, X., Li, C. K., & Rad, A. B. (2005). An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 6(3), 285-293.

Das, S., & Mandal, K. (2000). Modeling money demand in India: Testing weak, strong & super exogeneity. *Indian Economic Review*, 1-19.

Das, S., Dey, A., Pal, A., & Roy, N. (2015). Applications of artificial intelligence in machine learning: review and prospect. *International Journal of Computer Applications*, 115(9).

Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018). Unsupervised learning based on artificial neural network: A review. In 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 322-327). IEEE.

Duda, R. O., Hart, P. E. & Stork, D. G. (2001). Pattern Classification. Wiley, New Jersey.

Eagle, N., & Pentland, A. (2006). Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4), 255-268.

Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3), 381-396.

Frank, E., Trigg, L., Holmes, G., & Witten, I. H. (2000). Naive Bayes for regression. *Machine Learning*, 41, 5-25.

Grossman, G. M., & Rossi-Hansberg, E. (2008). Trading tasks: A simple theory of offshoring. *American Economic Review*, 98(5), 1978-1997.

Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques. Amsterdam: Elsevier.

Hanson, G. H., Mataloni Jr, R. J., & Slaughter, M. J. (2005). Vertical production networks in multinational firms. *Review of Economics and statistics*, 87(4), 664-678.

Harmon, S. A., Sanford, T. H., Xu, S., Turkbey, E. B., Roth, H., Xu, Z., ... & Turkbey, B. (2020). Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. Nature communications, 11(1), 4080.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics),* 28(1), 100-108.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications,* 13(4), 18-28.

Hegde, S., Shetty, S., Rai, S., & Dodderi, T. (2019). A survey on machine learning approaches for automatic detection of voice disorders. *Journal of Voice*, 33(6), 947-e11.

Hey, T., & Trefethen, A. (2003). The data deluge: An e-science perspective. *Grid computing: Making the global infrastructure a reality*, 72, 809-824.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6), 417.

Idrees, S. M., Alam, M. A., & Agarwal, P. (2019). A study of big data and its challenges. *International Journal of Information Technology*, 11, 841-846.

International Trade Centre (undated), "Trade Map", available at: http://www trademap org/Index aspx (accessed February 20, 2023).

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4(1), 237-285.

Kano, L., Tsang, E. W., & Yeung, H. W. C. (2020). Global value chains: A review of the multi-disciplinary literature. *Journal of international business studies*, 51, 577-622.

Keshavarzi Arshadi, A., Webb, J., Salem, M., Cruz, E., Calad-Thomson, S., Ghadirian, N., & Yuan, J. S. (2020). Artificial intelligence for COVID-19 drug discovery and vaccine development. *Frontiers in Artificial Intelligence*, 65.

Khadse, V., Mahalle, P. N., & Biraris, S. V. (2018, August). An empirical comparison of supervised machine learning algorithms for internet of things data. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-6). IEEE.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In 2014 *science and information conference* (pp. 372-378). IEEE.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). *Logistic regression* (p. 536). New York: Springer-Verlag.

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal*, 1(6), 90-95.

Koteluk, O., Wartecki, A., Mazurek, S., Kołodziejczak, I., & Mackiewicz, A. (2021). How do machines learn? artificial intelligence as a new era in medicine. *Journal of Personalized Medicine*, 11(1), 32.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, 39, 261-283.

Kowalski, P., Gonzalez, J. L., Ragoussis, A., & Ugarte, C. (2015). Participation of Developing Countries in Global Value Chains: Implications for Trade and Trade-Related Policies. *OECD Trade Policy Papers*, No. 179, OECD Publishing, Paris, https://doi.org/10.1787/5js33lfw0xxn-en.

Lade, P., Ghosh, R., & Srinivasan, S. (2017). Manufacturing analytics and industrial internet of things. *IEEE Intelligent Systems*, 32(3), 74-79.

Lalmuanawma, S., Hussain, J., & Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons & Fractals*, 139, 110059.

Lanz, R., & Miroudot, S. (2011). *Intra-firm trade: Patterns, determinants, and policy implications* (OECD Trade Policy Papers No. 114). Organization for Economic Cooperation and Development.

Le Glaz, A., Haralambous, Y., Kim-Dufor, D. H., Lenca, P., Billot, R., Ryan, T. C., & Lemey, C. (2021). Machine learning and natural language processing in mental health: systematic review. *Journal of Medical Internet Research*, 23(5), e15708.

Lee, C., & Landgrebe, D. A. (1993). Feature extraction based on decision boundaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4), 388-400.

Levine, M. D. (1969). Feature extraction: A survey. *Proceedings of the IEEE*, 57(8), 1391-1407.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6), 1-45.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision*, 128, 261-318.

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, 36(2), 451-461.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research*, 9, 381-386.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute.

Mirsadeghi, L., Haji Hosseini, R., Banaei-Moghaddam, A. M., & Kavousi, K. (2021). EARN: an ensemble machine learning algorithm to predict driver genes in metastatic breast cancer. *BMC Medical Genomics*, 14(1), 122.

Mohamadou, Y., Halidou, A., & Kapen, P. T. (2020). A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. *Applied Intelligence*, 50(11), 3913-3925.

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). Machine learning: algorithms and applications. CRC Press.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. John Wiley & Sons.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.

Mudambi, R., & Venzin, M. (2010). The strategic nexus of offshoring and outsourcing decisions. *Journal of Management Studies,* 47(8), 1510–1533.

Murphy, K. P. (2006). Naive bayes classifiers. *University of British Columbia*, 18(60), 1-8.

Nag, B., Chakraborty, D., & Aggarwal, S. (2021). India's Act East Policy: RCEP Negotiations and Beyond (No. 2101). Indian Institute of Foreign Trade, Delhi.

Nick, T. G., & Campbell, K. M. (2007). Logistic regression. *Topics in biostatistics*, 273-301.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11), 559-572.

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In *Machine learning* (pp. 101-121). Academic Press.

Pugliese, R., Regondi, S., & Marini, R. (2021). Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Science and Management*, 4, 19-29.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1, 81-106.

Ramachandran, M. (2004). Do broad money, output, and prices stand for a stable relationship in India? *Journal of Policy Modeling*, 26(8-9), 983-1001.

Ray, S. (2019). A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.

Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. *Journal of Big data*, 7(1), 1-29.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science,* 2(3), 160.

Sikora, R. (2015). A modified stacking ensemble machine learning algorithm using genetic algorithms. In *Handbook of research on organizational transformations through big data analytics* (pp. 43-53). IGi Global.

Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. In 2016 *3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1310-1315). IEEE.

Su, X., Yan, X., & Tsai, C. L. (2012). Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3), 275-294.

Tan, A. C., & Gilbert, D. (2003). Ensemble machine learning on gene expression data for cancer classification. Appl. Bioinf. 2 (3 Suppl. 1), S75-S83.

Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *expert systems with applications*, 36(10), 11994-12000.

United Nations Conference on Trade and Development (2019). World Investment Report, *Geneva: WTO*.

United Nations Conference on Trade and Development (2022). Trade and Development Report, *Geneva: WTO*.

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine learning*, 109(2), 373-440.

Vidal, R., Ma, Y., Sastry, S. S., Vidal, R., Ma, Y., & Sastry, S. S. (2016). *Principal component analysis* (pp. 25-62). Springer New York.

Wang, Y., Tetko, I. V., Hall, M. A., Frank, E., Facius, A., Mayer, K. F., & Mewes, H. W. (2005). Gene selection from microarray data for cancer classification—a machine learning approach. *Computational biology and chemistry*, 29(1), 37-46.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2005). Practical machine learning tools and techniques. *In Data Mining* (Vol. 2, No. 4).

World Trade Organisation (2011). *Trade patterns and global value chains in East Asia: From trade in goods to trade in tasks*. In collaboration with institute of developing economies (IDE) and Japan external trade organization (JETRO). WTO.

World Trade Organisation (2015). International Trade Statistics 2015, available at: www.wto.org/statistics (Accessed on September 5, 2022).

Yi, K. M. (2003). Can vertical specialization explain the growth of world trade? *Journal of political Economy*, 111(1), 52-102.

Zoabi, Y., Deri-Rozov, S., & Shomron, N. (2021). Machine learning-based prediction of COVID-19 diagnosis based on symptoms. *npj digital medicine*, 4(1), 3.

Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11).

Zhou, Z. H., & Zhou, Z. H. (2021). Semi-supervised learning. *Machine Learning*, 315-341.