INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 12, issue 12 (2025)

# METHODS FOR DETERMINING COMMODITY CODES OF GOODS AND VEHICLES ACCORDING TO THE COMMODITY NOMENCLATURE OF FOREIGN ECONOMIC ACTIVITY

**Khamroyev Ulugbek Rustamovich**
Associate Professor Customs Institute of the Customs Committee PhD in Technical Sciences
E-mail: uzbekiston@inbox.ru

**Abstract.** The article examines traditional and modern methods for determining commodity codes of goods and vehicles according to the Commodity Nomenclature of Foreign Economic Activity (CN FEA). Alongside methods dependent and independent of human factors, an automated classification system based on artificial intelligence technologies – Word2Vec (PV-DBOW model) and logistic regression – is proposed. The model was trained on data from import declarations in the Uzbekistan customs system and demonstrated 85.7% accuracy in test trials. This approach serves to accelerate customs processes, reduce errors, and facilitate the work of specialists.

**Keywords:** CN FEA, commodity classification, customs control, artificial intelligence, Word2Vec, logistic regression, import declaration, HS code, Harmonized System.

As is known, in the mid-19th century, as a result of the development of industry, science, and technology, foreign trade operations also advanced. The influx of new types of goods into global markets caused a number of problems in the process of customs control over various goods. This necessitated the classification of goods for customs purposes, systematic organization of commodity nomenclature, and the assignment of commodity codes in countries engaged in foreign trade.

Maintaining the Commodity Nomenclature of Foreign Economic Activity and classifying goods constitute one of the key components of customs operations and the important tasks assigned to customs authorities.[1]

In customs control, the identification of characteristics of goods and vehicles is carried out using methods dependent and independent of human factors (Table 1).

**Table 1.**
**Methods for identifying characteristics of goods and vehicles in the customs control process**

| No. | Methods dependent on human factors | Methods independent of human factors |
|---|---|---|
| 1. | Sight | Inspection-examination complexes |
| 2. | Smell | Technical tools or software |
| 3. | Taste | Laboratory equipment |
| 4. | Touch | Technical tools |
| 5. | Sound | Technical tools or software |
| 6. | Perception based on information type | Technical tools or software |

[1] Mirsagatov M.A. Commodity Nomenclature of Foreign Economic Activity. Study Guide. – Tashkent: Customs Institute, 2023. – 185 p.

INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 12, issue 12 (2025)

Analysis of images from inspection-examination complexes based on digital technologies.
The Commodity Nomenclature of Foreign Economic Activity assigns a set of digital codes to goods and vehicles according to certain criteria and patterns.

**What is commodity classification?**

Commodity classification is performed in accordance with the basic rules for interpreting the Harmonized System. Classifying goods according to the Harmonized System means determining the code of the goods.

Correct classification and naming of goods according to the basic rules for interpreting the Commodity Nomenclature of Foreign Economic Activity (CN FEA) provide opportunities to prevent a number of crimes that may occur in economic relations and to protect consumer interests.

**Methods for determining commodity codes**

1. According to chemical composition;
2. According to physical composition;
3. According to mathematical calculations;
4. According to technological methods (taking into account technical characteristics);
5. According to biological characteristics;
6. According to the software data of the goods;
7. Determined according to the description of the goods.

**The operating principle of the HS in determining CN FEA codes**

In accordance with the Resolution of the State Customs Committee of the Republic of Uzbekistan, registered on April 6, 2016, under No. 2773, "On Approval of the Instruction on the Procedure for Filling the Customs Cargo Declaration," the procedure for filling columns 31 and 33 of the Customs Cargo Declaration (CCD) is presented in Table 2.

**Table 2.**

**Procedure for filling columns 31 and 33 of the Customs Cargo Declaration**

| **Column** | **Name** | **Filling procedure** |
|---|---|---|
| 31 | Goods description | The column provides a commercial description of the goods sufficient to reliably identify the goods and determine their code according to the CN FEA. This code is written without spaces (probel) and other separating characters. |
| 33 | CN FEA code | The column indicates the commodity code according to the Commodity Nomenclature of Foreign Economic Activity (CN FEA) based on the Harmonized System. The declarant must provide a code that allows unambiguous classification of the goods, taking into account subsidiary rules for classification. |

- Import declaration data include information such as the CN FEA code, report name, product name, model standards, ingredients, and others (Table 3).
- For each record, the Harmonized System (HS) code consists of 10–12 digits. Data such as the declared name, product name, model standard, and ingredients are stored in textual form.
- Textual data (unstructured data) comprise the HS code, report name, product description, model standard, and ingredients. These textual data are used as the dataset.

**Table 3.**

**Sample of import declaration data**

| **CN FEA code** | **Goods description** |
|---|---|
| ... | ... |
| 8450900000 | 1. Electronic control module. Used for washing machines; part number |

| | |
|---|---|
| | EBR8565645; quantity 1 pc. – weight: 1.22 kg<br>2. part of carton box<br>8. 000<br>9. 000 |
| 841590009 | 1. Electronic module, control board. Used for air conditioners; part number EBR5765801; quantity 1 pc. – weight: 0.145 kg<br>2. part of carton box<br>8. 000<br>9. 000 |

The dataset consisting of the CN FEA code and description columns is divided into a training dataset and a test dataset.
In this classification practice, the training dataset (training_data.csv) and the test dataset (test_data.csv) are provided to evaluate the model's performance.
The ratio of the training dataset to the test dataset is typically 9:1.
train_data.csv represents the "train" dataset used to train the model.
test_data.csv represents the "test" dataset used to evaluate the model's performance.
In the classification practice, the Word2Vec model is applied. The embedding is performed similarly to Doc2Vec, relative to the description text. The CN FEA code is treated as a single word (paragraph token) to generate its embedding (Figure 1).

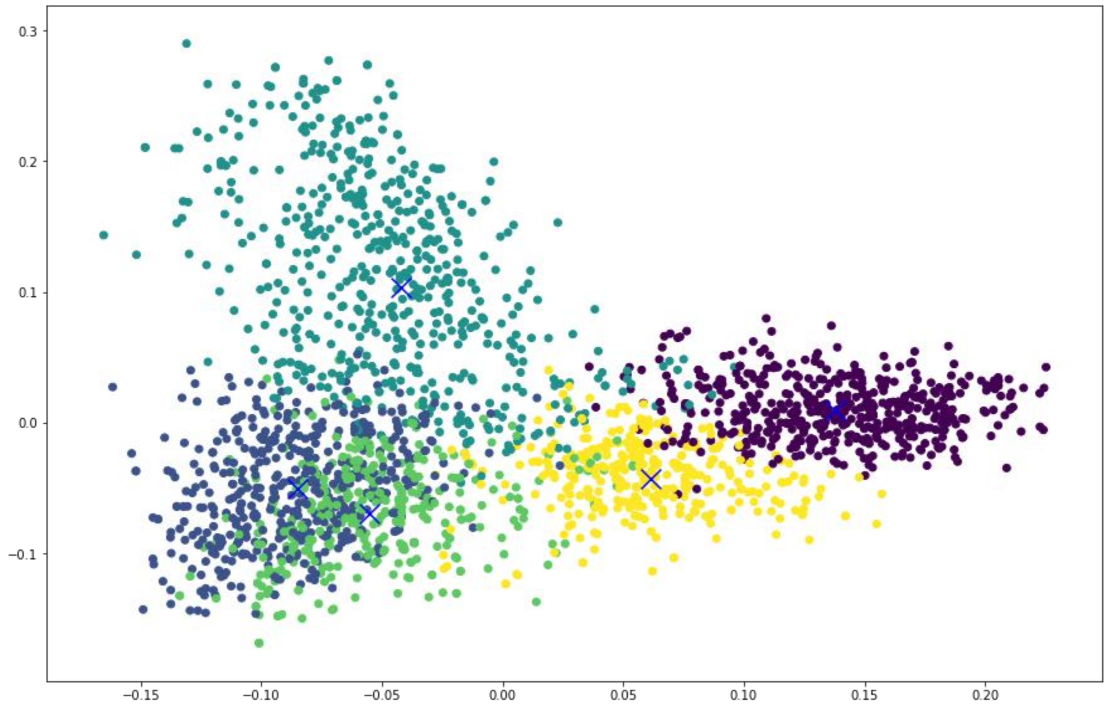
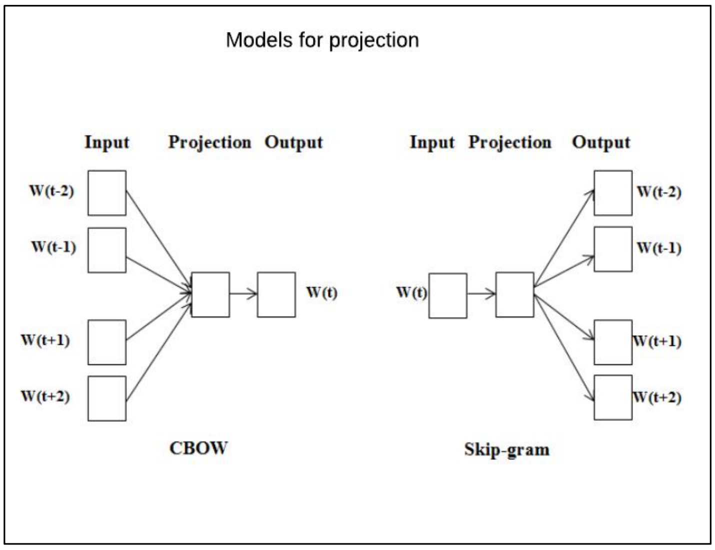


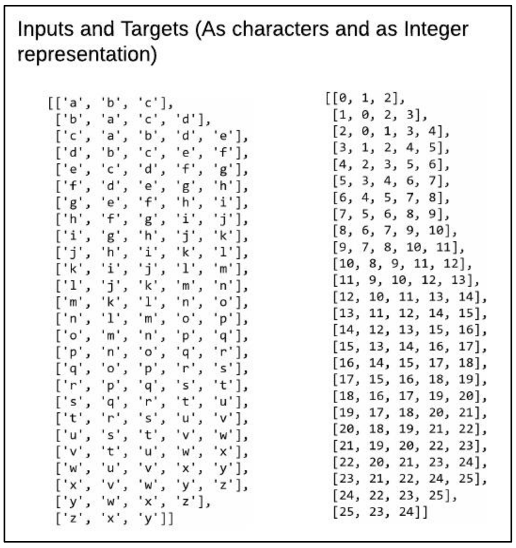
**Figure 1. Treating the CN FEA code as a single word (paragraph token) to generate its embedding**

Word2Vec has two models: DM (Distributed Memory) and DBOW (Distributed Bag of Words).
In this classification practice, we use the PV-DBOW method.
The DM (Distributed Memory) approach performs embedding by predicting the next words using paragraph vectors along with previous words.

INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 12, issue 12 (2025)

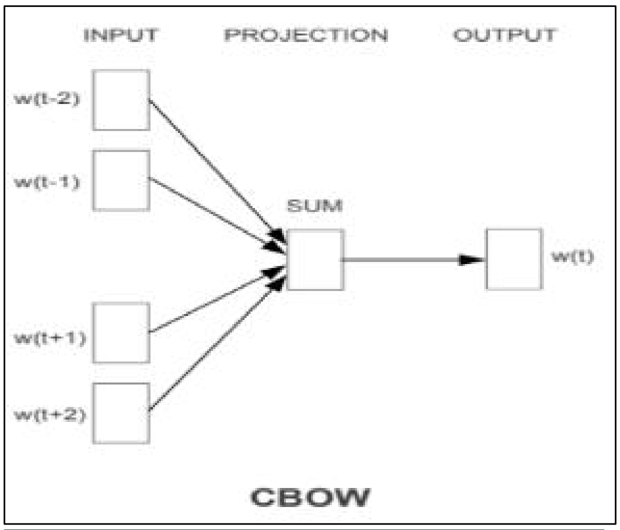
The PV-DBOW (Paragraph Vector – Distributed Bag of Words) method performs embedding by randomly predicting words in the paragraph using only the paragraph identifier. The paragraph vector serves as the input, while the output is a randomly selected word from the paragraph.

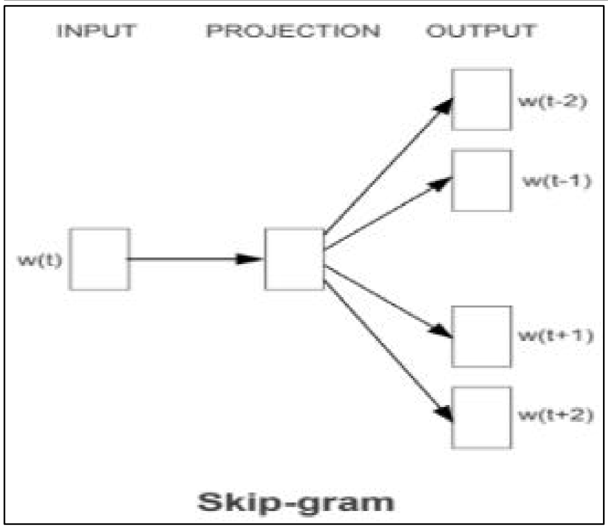



Inputs and Targets (As characters and as Integer representation)

```
[['a', 'b', 'c'],
 ['b', 'a', 'c', 'd'],
 ['c', 'a', 'b', 'd', 'e'],
 ['d', 'b', 'c', 'e', 'f'],
 ['e', 'c', 'd', 'f', 'g'],
 ['f', 'd', 'e', 'g', 'h'],
 ['g', 'e', 'f', 'h', 'i'],
 ['h', 'f', 'g', 'i', 'j'],
 ['i', 'g', 'h', 'j', 'k'],
 ['j', 'h', 'i', 'k', 'l'],
 ['k', 'i', 'j', 'l', 'm'],
 ['l', 'j', 'k', 'm', 'n'],
 ['m', 'k', 'l', 'n', 'o'],
 ['n', 'l', 'm', 'o', 'p'],
 ['o', 'm', 'n', 'p', 'q'],
 ['p', 'n', 'o', 'q', 'r'],
 ['q', 'o', 'p', 'r', 's'],
 ['r', 'p', 'q', 's', 't'],
 ['s', 'q', 'r', 't', 'u'],
 ['t', 'r', 's', 'u', 'v'],
 ['u', 's', 't', 'v', 'w'],
 ['v', 't', 'u', 'w', 'x'],
 ['w', 'u', 'v', 'x', 'y'],
 ['x', 'v', 'w', 'y', 'z'],
 ['y', 'w', 'x', 'z'],
 ['z', 'x', 'y']]
```

```
[[0, 1, 2],
 [1, 0, 2, 3],
 [2, 0, 1, 3, 4],
 [3, 1, 2, 4, 5],
 [4, 2, 3, 5, 6],
 [5, 3, 4, 6, 7],
 [6, 4, 5, 7, 8],
 [7, 5, 6, 8, 9],
 [8, 6, 7, 9, 10],
 [9, 7, 8, 10, 11],
 [10, 8, 9, 11, 12],
 [11, 9, 10, 12, 13],
 [12, 10, 11, 13, 14],
 [13, 11, 12, 14, 15],
 [14, 12, 13, 15, 16],
 [15, 13, 14, 16, 17],
 [16, 14, 15, 17, 18],
 [17, 15, 16, 18, 19],
 [18, 16, 17, 19, 20],
 [19, 17, 18, 20, 21],
 [20, 18, 19, 21, 22],
 [21, 19, 20, 22, 23],
 [22, 20, 21, 23, 24],
 [23, 21, 22, 24, 25],
 [24, 22, 23, 25],
 [25, 23, 24]]
```





**Logistic Regression Model**

Logistic regression employs an equation similar to that of linear regression for its representation. The input data values (x) related to the goods description are linearly combined using weights or coefficient values to predict the output value of the CN FEA code (y).

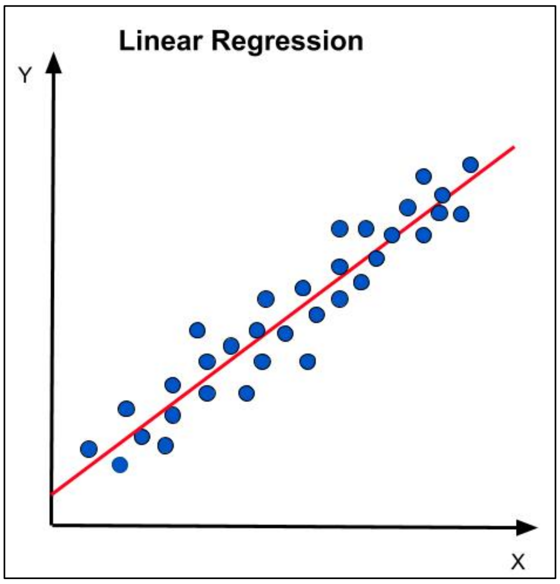
The primary difference from linear regression is that the modeled output value is not a continuous numerical value but a binary value (0 or 1). An example of the logistic regression equation is provided below:

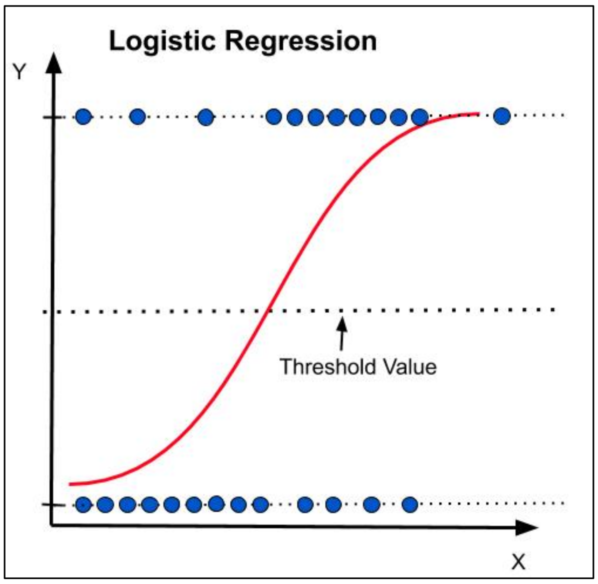
$$y=\frac{e^{(b_0+b_1x)}}{1+e^{(b_0+b_1x)}}$$

Here, y represents the predicted output, $b_0$ is the bias or intercept term, and $b_1$ is the coefficient for a single input value (x). For each column in the input data, there is a corresponding b coefficient (a constant real value) that must be learned from the training data. The actual form of the model that is stored in memory or in a file consists of these coefficients in the equation (beta values or b).

**Logistic Regression vs. Linear Regression:**

| **Aspect** | **Linear Regression** | **Logistic Regression** |
|---|---|---|
| Output type | Continuous numerical value (e.g., price, weight) | Binary (0 or 1) or probabilistic (between 0 and 1) |
| Equation | $y=b_0+b_1x$ | $y=\frac{1}{1+e^{-(b_0+b_1x)}}$ (sigmoid function) |
| Purpose | Predict continuous values | Classification (predict class probabilities) |
| Output interpretation | Direct predicted value | Probability of belonging to class 1 |
| Loss function | Mean Squared Error (MSE) | Log loss (cross-entropy) |





**Predicting the CN FEA Code with Probability**

Using the train_data.csv file, all goods descriptions are learned through Word2Vec embedding features applied within a Logistic Regression model, resulting in the creation of a unified database.

With the trained model, embeddings are generated for descriptions in the test_data.csv file (test set).
The model calculates similarity against the vectors of trained CN FEA codes and outputs the single CN FEA code vector with the highest probability (Table 4).

**Table 4.**
**Sample results of predicting the CN FEA code with probability on the test set**

| No. | CN FEA code (actual) | Model predicted code | Accuracy |
|---|---|---|---|
| 1 | 8711609000 | 8711609000 | True |
| 2 | 8421290009 | 3926909709 | False |
| 3 | 8302420000 | 8302420000 | True |
| 4 | 7323930000 | 7323930000 | True |
| 5 | 3926909709 | 3926909709 | True |
| 6 | 3926400000 | 3926400000 | True |
| … | … | … | … |

**Testing the Model**

```
# Loading the 'word2vec-ruscorpora-300' library required for us into the program wv = api.load('word2vec-ruscorpora-300')
# Formulas (functions) for converting the given sentences into vector representation def sent_vec(sent):
vector_size = wv.vector_size
wv_res = np.zeros(vector_size)
ctr = 1
for w in [str(word.lemma_.lower().strip())+'_'+word.pos_ for word in nlp(sent)]:
if w in wv:
ctr += 1
wv_res += wv[w]
wv_res = wv_res / ctr
return wv_res
# Introducing a variable for various punctuations/symbols
punctuations = string.punctuation
# Calling the Russian language numbers (stop words/library, likely) library for our model
nlp = spacy.load("ru_core_news_sm")
# Introducing a function for tokenizing the given sentences
def spacy_tokenizer(sentence):
doc = nlp(sentence)
mytokens = [str(word.lemma_.lower().strip())+'_'+word.pos_ for word in doc]
mytokens = [word for word in mytokens if (word not in stop_words) and (word not in punctuations)]
return mytokens
# Converting the given commodity information into a vector representation
df_train['vec'] = df_train['REFINED_TEXT'].apply(sent_vec) df_train
```

| | **HSCODE** | **REFINED_TEXT** | **vec** |
|---|---|---|---|
| **0** | 2201101100 | "holy spring" carbonated natural water lemon p... | [-0.017241601573510304, - 0.005597955340312587,... |
| **1** | 2201101100 | "holy spring" carbonated natural water lemon p... | [-0.0073695062543265525, - 0.00437865536659956,... |
| **2** | 2201101100 | "svyatoy istochnik" non-carbonated natural drinking water... | [0.00471258592072197, 0.006382490423592654, -0... |
| **3** | 2201101100 | "svyatoy istochnik" non-carbonated natural drinking water... | [-0.008963648176921362, - 0.0013491364026611502... |
| **4** | 2201101100 | "svyatoy istochnik" non-carbonated natural drinking water... | [-0.0011369850004224905, 0.011143063344726605,... |
| **...** | ... | ... | ... |

```
10032 rows × 3 columns
# Introducing variables using the given dataset
# The main reason for this is to separate the test dataset using these variables.
X = df_train['vec'].to_list()
# Feature matrix (vectors of the goods' descriptions)
y = df_train['HSCODE'].to_list()
# Target variable (HS Codes)
z = df_train['REFINED_TEXT']
# Original text (for comparison/debugging)
X, y
# Separating the test dataset for the process
from sklearn.model_selection import train_test_split
# Splitting the data into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test, z_train, z_test = train_test_split (X, y, z, test_size=0.3)
# The sizes (dimensions) of the datasets ready for the process
len(X_train), len(X_test), len(y_train), len(y_test), len(z_train), len(z_test)
# Calling the Logistic Regression method
from sklearn.linear_model import Logistic Regression
classifier = Logistic Regression()
# Training the Datasets using the given model
classifier.fit (X_train, y_train)
# Predicting the results for the test dataset after the model is trained
from sklearn import metrics
predicted = classifier.predict(X_test)
# Comparing the results using the predicted outcomes
compare = pd.DataFrame(data=[y_test, X_test, z_test]).T
compare['predicted'] = pd.DataFrame(predicted)
```

compare

| | **Actual HSCODE** | **Predicted HSCODE** | **Vector (partial)** | **Refined Goods Description (translated to English)** |
|---|---|---|---|---|
| 0 | 2201101100 | 2201101900 | [-0.0077168666903162375, 0.0068682209503921595, ...] | "Montella" non-carbonated mineral water, PET bottle... |
| 1 | 2202100000 | 2202100000 | [0.012580711394548416, -0.014026077202288434, ...] | Non-alcoholic malt drink with tropical fruits flavor... |
| 2 | 2202100000 | 2202100000 | [0.029732314869761467, -0.0233515050661351, -0...] | Non-alcoholic malt drink with peach flavor... |
| 3 | 2202100000 | 2202100000 | [0.005107819257924954, -0.015828499880929787, ...] | Mineral water with watermelon-strawberry flavor under the brand "..." |
| 4 | 2202100000 | 2202100000 | [0.021808310012732233, 0.02534417887883527, 0....] | Carbonated drink "Pepsi", total boxes for local market... |
| ... | ... | ... | ... | ... |

2508 rows × 4 columns

Assessing the model's effectiveness through the predicted results

| | **Actual HSCODE** | **Predicted HSCODE** | **Vector (partial)** | **Refined Goods Description (translated to English)** | **Correct Prediction** |
|---|---|---|---|---|---|
| 0 | 2201101100 | 2201101900 | [-0.0077168666903162375, 0.0068682209503921595, ...] | "Montella" non-carbonated mineral water, PET bottle... | False |
| 1 | 2202100000 | 2202100000 | [0.012580711394548416, -0.014026077202288434, ...] | Non-alcoholic malt drink with tropical fruits flavor... | True |
| 2 | 2202100000 | 2202100000 | [0.029732314869761467, -0.0233515050661351, -0...] | Non-alcoholic malt drink with peach flavor... | True |
| 3 | 2202100000 | 2202100000 | [0.005107819257924954, -0.015828499880929787, ...] | Mineral water with watermelon-strawberry | True |

| | | | | | |
|---|---|---|---|---|---|
| | | | | flavor under the brand "..." | |
| 4 | 2202100000 | 2202100000 | [0.021808310012732233, 0.02534417887883527, 0....] | Carbonated drink "Pepsi" for local market, in boxes... | True |
| ... | ... | ... | ... | ... | ... |

2508 rows × 5 columns

Output of Model Performance Evaluation

```
print("Total evaluation data count: {}".format(len(compare)))
print("Count included in top 1: {} / {}".format(len(compare[compare['INCLUDED'] == True]), len(compare)))
print("Inclusion rate: {}%".format(round(len(compare[compare['INCLUDED'] == True]) / len(compare) * 100, 2)))
Total evaluation data count: 12428
Count included in top 1: 10577 / 12428
Inclusion rate: 85,7%
```

**Conclusion**

We have trained an Artificial Intelligence (AI) model using historical import declaration data.
Additionally, we have implemented an algorithm that recommends the single CN FEA code with the highest similarity by inputting import declaration data into the AI system.
When evaluating the model's performance on unseen new data, the result achieved an accuracy of 85.7%.

**Expected outcomes of the model**

By applying this artificial intelligence approach to the previous methods of determining CN FEA codes, we can identify the correct CN FEA codes more quickly and accurately.
The positive impacts of this approach can be observed from the perspectives of both the importer and customs specialists as follows:
First, it reduces the time spent by importers in finding the correct CN FEA code.
As a result, this approach enhances the efficiency of foreign economic activity through the automation of customs control and helps prevent errors.

**References.**

1. Customs Code of the Republic of Uzbekistan. – Tashkent, 2023.
2. Resolution No. 2773 of the State Customs Committee of the Republic of Uzbekistan dated April 6, 2016 “On the Procedure for Filling the Customs Cargo Declaration”.
3. World Customs Organization. Harmonized Commodity Description and Coding System. – Brussels, 2022.
4. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J. Distributed Representations of Words and Phrases and their Compositionality // Advances in Neural Information Processing Systems. – 2013.
5. Jurafsky D., Martin J.H. Speech and Language Processing. – Pearson Education, 2021.

6. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. – O'Reilly Media, 2022.
7. Bishop C.M. Pattern Recognition and Machine Learning. – Springer, 2019.