

Model Development Phase Template

Date	11 March 2025
Team ID	740052
Project Title	AI-Based Intelligent Insight Extractor
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report:

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training code:

```
[2] # passing the input data
doc=""Artificial neural networks are the brains behind some of the most sophisticated applications of artificial intelligence (AI). But that doesn't mean understanding the different t

In machine learning, each type of artificial neural network is tailored to perform certain sets of tasks. In order to explain these tasks and the best approaches to completing them, th

What's the difference between CNN and RNN?
The main difference between a CNN and an RNN is the ability to process temporal information – data that comes in sequences, such as a sentence. Recurrent neural networks are designed f

CNNs employ filters within convolutional layers to transform data (more on that later), whereas RNNs are predictive, reusing activation functions from other data points in the sequence

Once you look at the structure of both types of neural networks and understand what they are used for, the difference between CNN and RNN becomes more clear.""
```

Artificial neural networks are the brains behind some of the most sophisticated applications of artificial intelligence (AI). But that doesn't mean understanding the different types need

In machine learning, each type of artificial neural network is tailored to perform certain sets of tasks. In order to explain these tasks and the best approaches to completing them, thi

What's the difference between CNN and RNN?

The main difference between a CNN and an RNN is the ability to process temporal information – data that comes in sequences, such as a sentence. Recurrent neural networks are designed fo

CNNs employ filters within convolutional layers to transform data (more on that later), whereas RNNs are predictive, reusing activation functions from other data points in the sequence

Once you look at the structure of both types of neural networks and understand what they are used for, the difference between CNN and RNN becomes more clear.

Word Tokenization

```
[7] #Word tokenization is performed
tokens = [i.text for i in docs]
print(tokens)
```

```
['Artificial', 'neural', 'networks', 'are', 'the', 'brains', 'behind', 'some', 'of', 'the', 'most', 'sophisticated', 'applications', 'of', 'artificial', 'intelligence', '(', 'AI', ')',
```

```
word_frequencies = {} # Initialize word_frequencies as a dictionary
for word in docs:
    if word.text.lower() not in stopWords:
        if word.text.lower() not in punctuation:
            if word.text not in word_frequencies: # Check if the word is already a key in the dictionary
                word_frequencies[word.text]=1
            else:
                word_frequencies[word.text] +=1

print(word_frequencies)
```

```
{'Artificial': 1, 'neural': 10, 'networks': 9, 'brains': 1, 'sophisticated': 1, 'applications': 3, 'artificial': 4, 'intelligence': 1, 'AI': 2, 'mean': 1, 'understanding': 1, 'different
```

```
[13] from heapq import nlargest
```

```
[14] select_len = int(len(sent_tokens)*0.3)
select_len
```

```
3
```

```
[15] summary = nlargest(select_len, sentence_score, sentence_score.get)
summary
```

```
[In order to explain these tasks and the best approaches to completing them, this article will introduce two types of artificial neural networks: convolutional neural networks (CNNs) and recurrent neural networks (RNNs)., Recurrent neural networks are designed for this very purpose, while convolutional neural networks are incapable of effectively interpreting temporal information., When it comes to artificial neural networks – computing systems that mimic components of the brain – there are serious differences in type.]
```

```
[16] summary = [word.text for word in summary]
summary
```