Vector Search with OpenSearch

Step 1 – Upload model and copy the returned task ID

```
POST /_plugins/_ml/models/_upload
{
  "name": "huggingface/sentence-transformers/all-mpnet-base-v2",
  "version": "1.0.1",
  "model_format": "TORCH_SCRIPT"
}
```

Step 2 – Use task ID to get model ID

```
GET /_plugins/_ml/tasks/<your-task-id>
```

Step 3 – Load the model with the model ID

```
POST /_plugins/_ml/models/<your-model-id>/_load
```

Step 3 – Create an ingest pipeline with model ID

```
PUT /_ingest/pipeline/nlp-ingest-pipeline
{
  "description": "A text embedding pipeline",
  "processors": [
    {
      "text_embedding": {
        "model_id": "<your-model-id>",
        "field_map": {
          "passage_text": "passage_embedding"
        }
      }
    }
  ]
}
```

Step 4 – Create a search pipeline with model ID

```
PUT /_search/pipeline/default_model_pipeline
{
  "request_processors": [
    {
      "neural_query_enricher": {
        "default_model_id": "<your-model-id>"
      }
    }
  ]
}
```

Step 5 – Create an index that supports vector search

```
PUT /my-nlp-index
{
  "settings": {
    "index.knn": true,
    "default_pipeline": "nlp-ingest-pipeline",
    "index.search.default_pipeline" : "default_model_pipeline"
  },
  "mappings": {
    "properties": {
      "passage_embedding": {
        "type": "knn_vector",
        "dimension": 768,
        "method": {
          "engine": "lucene",
          "space_type": "l2",
          "name": "hnsw",
          "parameters": {}
        }
      },
      "passage_text": {
        "type": "text"
      }
    }
  }
}
```

Step 6 – Index some data

```
PUT /my-nlp-index/_doc/1
{
  "passage_text": "Hello world"
}

PUT /my-nlp-index/_doc/2
{
  "passage_text": "Hi planet"
}

PUT /my-nlp-index/_doc/3
{
  "passage_text": "Dirty Dancing"
}

PUT /my-nlp-index/_doc/4
{
  "passage_text": "How to catch a salmon"
}

PUT /my-nlp-index/_doc/5
{
  "passage_text": "Happy Feet"
}
```

Step 7 – Check the vector field

```
GET /my-nlp-index/_search
```

Step 8 – Perform a neural query

```
GET /my-nlp-index/_search
{
  "_source": {
    "excludes": [
      "passage_embedding"
    ]
  },
  "query": {
    "neural": {
      "passage_embedding": {
        "query_text": "hello world",
        "k": 100
      }
    }
  }
}
```