**Q:** **You are working on an NLP model. So, you are dealing with words and sentences, not numbers. Your problem is to categorize these words and make sense of them. Your manager told you that you have to use embeddings.**

**Which of the following techniques are not related to embeddings? Explain the other terms related to embedding.**

**A. Count Vector**

**B. TF-IDF Vector**

**C. Co-Variance Matrix**

**Ans:** Covariance matrix is not an embedding technique. Covariance matrices are square matrices with the covariance between each pair of elements. It measures how much the change of one with respect to another is related.

**Other two are embedding techniques:**

**Count Vector**

* It is one of the simplest ways of doing text vectorization.
* It creates a document term matrix, which is a set of dummy variables that indicates if a particular word appears in the document.
* Count vectorizer will fit and learn the word vocabulary and try to create a document term matrix in which the individual cells denote the frequency of that word in a particular document, which is also known as term frequency, and the columns are dedicated to each word in the corpus.

**Example:**

**Document-1:** He is a smart boy. She is also smart.

**Document-2:** Chirag is a smart person.

The dictionary created contains the list of unique tokens(words) present in the corpus

**Unique Words:** [‘He’, ’She’, ’smart’, ’boy’, ’Chirag’, ’person’] Here, D=2, N=6

So, the count matrix M of size 2 X 6 will be represented as –

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **He** | **She** | **smart** | **boy** | **Chirag** | **person** |
| **D1** | **1** | **1** | **2** | **1** | **0** | **0** |
| **D2** | **0** | **0** | **1** | **0** | **1** | **1** |

**TF-IDF**

Term frequency denotes the frequency of a word in a document. For a specified word, it is defined as the ratio of the number of times a word appears in a document to the total number of words in the document.



Inverse document frequency looks at how common (or uncommon) a word is amongst the corpus. It measures the importance of the word in the corpus.

Document Frequency tells us about the proportion of documents that contain a certain word. IDF is the reciprocal of the Document Frequency.

The intuition behind using IDF is that the more common a word is across all documents, the lesser its importance is for the current document. A logarithm is taken to dampen the effect of (normalize) IDF in the final calculation.

The final TF-IDF score comes out to be:

**Q: You are a junior Data Scientist and are working on a deep neural network model to optimize the level of customer satisfaction for after-sales services with the goal of creating greater client loyalty. You are struggling with your model (learning rates, hidden layers and nodes selection) for optimizing processing and to let it converge in the fastest way. What is this problem called in ML language? Explain**

**Answer: Hyperparameter Tuning**

* **Hyperparameters** are configuration variables that influence the training process itself: Learning rate, hidden layers number, number of epochs, regularization, batch size are all examples of hyperparameters.
* Explanation

**Q: Imagine, you are working with a microblogging website and you want to develop a machine learning algorithm which predicts the number of views on the articles.**

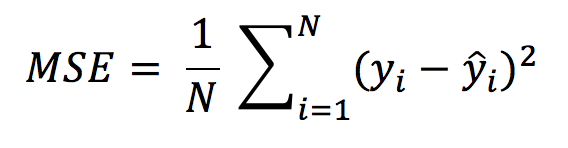
**Your analysis is based on features like author name, number of articles written by the same author in past and a few other features. Which of the following evaluation metric would you choose in that case? Explain the metric used.**

1. **Mean Square Error**
2. **Accuracy**
3. **F1 Score**

**Solution: Mean Square Error**

The number of views of articles is the continuous target variable which fall under the regression problem. So, mean squared error will be used as an evaluation metrics.

Mean Square Error is an absolute measure of the goodness for the fit.



* MSE is calculated by the sum of square of prediction error which is real output minus predicted output and then divide by the number of data points.
* It gives you an absolute number on how much your predicted results deviate from the actual number.
* You cannot interpret many insights from one single result but it gives you a real number to compare against other model results and help you select the best regression model.

Q: **You are working with categorical feature(s) and you have not looked at the distribution of the categorical variable in the test data. You want to apply one hot encoding (OHE) on the categorical feature(s). What challenges you may face if you have applied OHE on a categorical variable of train dataset?**

* All categories of categorical variable are not present in the test dataset.
* Frequency distribution of categories is different in train as compared to the test dataset.

The OHE will fail to encode the categories which is present in test but not in train so it could be one of the main challenges while applying OHE. The challenge given in option B is also true you need to more careful while applying OHE if frequency distribution doesn’t same in train and test.

Q: **You’ve built a random forest model with 10000 trees. You got delighted after getting training error as 0.00. But, the validation error is 34.23. What is going on? Haven’t you trained your model perfectly?**

**Answer:** The model has overfitted. Training error 0.00 means the classifier has mimicked the training data patterns to an extent, that they are not available in the unseen data. Hence, when this classifier was run on unseen sample, it couldn’t find those patterns and returned prediction with higher error. In random forest, it happens when we use larger number of trees than necessary. Hence, to avoid these situations, we should tune number of trees using cross validation.

Q:  **We know that one hot encoding increasing the dimensionality of a data set. But, label encoding doesn’t. How?**

**Answer:** Using one hot encoding, the dimensionality (a.k.a features) in a data set get increased because it creates a new variable for each level present in categorical variables. For example: let’s say we have a variable ‘color’. The variable has 3 levels namely Red, Blue and Green. One hot encoding ‘color’ variable will generate three new variables as Color.Red, Color.Blue and Color.Green containing 0 and 1 value.

In label encoding, the levels of a categorical variables gets encoded as 0 and 1, so no new variable is created. Label encoding is majorly used for binary variables.

Q: **What do you understand by Type I vs Type II error ?**

**Answer:** Type I error is committed when the null hypothesis is true and we reject it, also known as a ‘False Positive’. Type II error is committed when the null hypothesis is false and we accept it, also known as ‘False Negative’.

In the context of confusion matrix, we can say Type I error occurs when we classify a value as positive (1) when it is actually negative (0). Type II error occurs when we classify a value as negative (0) when it is actually positive(1).

Q: **A linear regression model is generally evaluated using MSE or Adjusted R². How would you evaluate a logistic regression model?**

**Answer:** Since logistic regression is used to predict probabilities, we can use AUC-ROC curve along with confusion matrix to determine its performance. Explanation of AUC-ROC.

**Q: Explain dimensionality reduction, where it’s used, and its benefits?**

Dimensionality reduction is the process of reducing the number of feature variables under consideration by obtaining a set of principal variables which are basically the important features. Importance of a feature depends on how much the feature variable contributes to the information representation of the data and depends on which technique you decide to use. Deciding which technique to use comes down to trial-and-error and preference. It’s common to start with a linear technique and move to non-linear techniques when results suggest inadequate fit. Benefits of dimensionality reduction for a data set may be:

* Reduce the storage space needed.
* Speed up computation (for example in machine learning algorithms), less dimensions mean less computing, also less dimensions can allow usage of algorithms unfit for a large number of dimensions.
* Remove redundant features, for example no point in storing a terrain’s size in both sq meters and sq miles (maybe data gathering was flawed).
* Reducing a data’s dimension to 2D or 3D may allow us to plot and visualize it, maybe observe patterns, give us insights.
* Too many features or too complex a model can lead to overfitting.

**Q: Explain over- and under-fitting and how to combat them?**

**Q: Why is ReLU better and more often used than Sigmoid in Neural Networks?**

**Q: What is the curse of dimensionality? List some ways to deal with it?**

The curse of dimensionality is when the training data has a high feature count, but the dataset does not have enough samples for a model to learn correctly from so many features. For example, a training dataset of 100 samples with 100 features will be very hard to learn from because the model will find random relations between the features and the target. However, if we had a dataset of 100k samples with 100 features, the model could probably learn the correct relationships between the features and the target.

There are different options to fight the curse of dimensionality:

* **Feature selection.** Instead of using all the features, we can train on a smaller subset of features.
* **Dimensionality reduction.** There are many techniques that allow to reduce the dimensionality of the features. Principal component analysis (PCA) and using autoencoders are examples of dimensionality reduction techniques.
* **L1 regularization.** Because it produces sparse parameters, L1 helps to deal with high-dimensionality input.
* **Feature engineering.** It’s possible to create new features that sum up multiple existing features. For example, we can get statistics such as the mean or median.