EXP 9

1. **Data preprocessing:**

 Data preprocessing is the process of transforming raw data into an understandable format.The quality of the data should be checked before applying machine learning or data mining algorithms.

Drop the fields you think have no use for the modeling or are closely related to other attributes. Dimensionality reduction is one of the very important aspects of Data Preprocessing.

**Major Tasks in Data Preprocessing:**

1. Data cleaning
2. Data integration
3. Data reduction
4. Data transformation

* Data cleaning is the process to remove incorrect data, incomplete data and inaccurate data from the datasets, and it also replaces the missing values. Standard values like “Not Available” or “NA” can be used to replace the missing values.
* The process of combining multiple sources into a single dataset. The Data integration process is one of the main components in data management.
* This process helps in the reduction of the volume of the data which makes the analysis easier yet produces the same or almost the same result. This reduction also helps to reduce storage space. There are some of the techniques in data reduction are Dimensionality reduction, Numerosity reduction, Data compression.
* The change made in the format or the structure of the data is called data transformation. This step can be simple or complex based on the requirements. There are some methods in data transformation i.e smoothing,aggregation,discretization,normalization.

1. **tokenization:**

Tokenization is a way of separating a piece of text into smaller units called tokens.

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, perhaps at the same time throwing away certain characters, such as punctuation.

Here, tokens can be either words, characters, or subwords.

Hence, tokenization can be broadly classified into 3 types – word, character, and subword (n-gram characters) tokenization.

The most common way of forming tokens is based on space.

For ex : machine learning lab

The number of tokens are 3.

Similarly, tokens can be either characters or subwords. For example, let us consider “smarter”:

1. Character tokens: s-m-a-r-t-e-r
2. Subword tokens: smart-er

We have

>word tokeninzation : . It splits a piece of text into individual words based on a certain delimiter. Depending upon delimiters, different word-level tokens are formed. Word2Vec and GloVe comes under word tokenization

>subword tokenization: Subword Tokenization splits the piece of text into subwords (or n-gram characters). For example, words like lower can be segmented as low-er, smartest as smart-est, and so on. Transformed based models – the SOTA in NLP – rely on Subword Tokenization algorithms.

>character tokenization: Character Tokenization splits apiece of text into a set of characters. Character Tokenizers handles OOV words coherently by preserving the information of the word. It breaks down the OOV(out of vocabulary) word into characters and represents the word in terms of these character. It also limits the size of the vocabulary

Also, Byte Pair Encoding (BPE) is a widely used tokenization method among transformer-based models. BPE addresses the issues of Word and Character Tokenizers:

* BPE tackles OOV effectively. It segments OOV as subwords and represents the word in terms of these subwords
* The length of input and output sentences after BPE are shorter compared to character tokenization

BPE is a word segmentation algorithm that merges the most frequently occurring character or character sequences iteratively.

3>**word embedding** :

Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network.

In order to map text data into a form that an algorithm can process, we transformed the data into numbers using one-hot encoding. What we basically do here, we take our vocabulary of say N size and map this vocabulary into a set of N vectors of 1s and 0s, each of size N. Every vector represents a word. The*ith*-word has a 1 in the vector at the *ith* position and rest of vector is 0. For example, if our vocabulary consists of say four words, “Computer,” “Machine,” “Learning,” “Language,” then we will represent this set of text using 4 vectors of size 4 – [1000], [0100], [0010] and [0001]. 5 years back, this was pretty much how we processed text data before applying any algorithm.

One of the most popular algorithms in the word embedding space has been *Word2Vec*.

It proposes two different architectures: the Skip-gram model and the Continuous-Bag-Of-Words model (CBOW).

n both methods, the user sets a *window size*that defines how many surrounding context words to consider in the modeling. In general, more context means better predictions but also places more computational stress on the model.

.> In the Skip-gram method, the model takes the target word for the embedding being learned and seeks to predict the surrounding context words from it.

.> . In the CBOW model, the model takes the context surrounding the target word and seeks to predict the target word from it.

In practice, the Skip-gram model is often favored because it can be trained accurately with smaller datasets and more accurately represent uncommon/rare words. The main advantage of the CBOW model is that it is less expensive to train, which can be an important determinant depending on your particular set of constraints.

One of the primary uses for word embeddings is for determining *similarity*, either in meaning or in usage. This is usually done by computing a metric such as the *cosine similarity*(ie. a normalized dot product), between two vectors. If the embedding model was trained properly, then similar words such as *storm*and *gale*will show a high cosine similarity, as measured on a scale from 0 to 1.

Other algos are GloVe,ELMO,BERT.

EXP 12

* 1. **Model deployment:** It means to integrate a machine learning model and integrate it into an existing production environment, where it can take in an input and return an output. The purpose of deploying your model is so that you can make the predictions from a trained ML model available to others, whether that be users, management, or other systems.

At a high-level, there are four main parts to an ML system:

>**Data Layer:** the data layer provides access to all of the data sources that the model will require.

>**Feature Layer**: the feature layer is responsible for generating feature data in a transparent, scalable, and usable manner.

>**Scoring Layer**: the scoring layer transforms features into predictions. Scikit-learn is most commonly used and is the industry standard for scoring.

>**Evaluation Layer**: the evaluation layer checks the equivalence of two models and can be used to monitor production models. I.e. it is used to monitor and compare how closely the training predictions match the predictions on live traffic.

* 1. **Testing/Evaluating:**  It is a method of assessing the correctness of models on test data. The test data consists of data points that have not been seen by the model before. Any given model has several limitations depending on the data distribution. None of them can be entirely accurate since they are just *estimations.* These limitations are popularly known by the name of ***bias*** and ***variance***.

A model with high bias will oversimplify by not paying much attention to the training points (e.g.: in Linear Regression, irrespective of data distribution, the model will always assume a linear relationship).

A model with high variance will restrict itself to the training data by not generalizing for test points that it hasn’t seen before (e.g.: Random Forest with max\_depth = None).

Types of model selection:

**Resampling methods**

Resampling methods, as the name suggests, are simple techniques of rearranging data samples to inspect if the model performs well on data samples that it has not been trained on

**Random Split**

Random Splits are used to randomly sample a percentage of data into training, testing, and preferably validation sets. The advantage of this method is that there is a good chance that the original population is well represented in all the three sets. In more formal terms, random splitting will prevent a biased sampling of data.

**Time-Based Split**

There are some types of data where random splits are not possible. For example, if we have to train a model for weather forecasting, we cannot randomly divide the data into training and testing sets. This will jumble up the seasonal pattern! Such data is often referred to by the term – Time Series.

In such cases, a time-wise split is used. The training set can have data for the last three years and 10 months of the present year. The last two months can be reserved for the testing or validation set.

**K-Fold Cross-Validation**

The cross-validation technique works byrandomly shuffling the dataset and then splitting it into k groups. Thereafter, on iterating over each group, the group needs to be considered as a test set while all other groups are clubbed together into the training set. The model is tested on the test group and the process continues for k groups.

Thus, by the end of the process, one has k different results on k different test groups. The best model can then be selected easily by choosing the one with the highest score.

**Stratified K-Fold**

The process for stratified K-Fold is similar to that of K-Fold cross-validation with one single point of difference – unlike in k-fold cross-validation, the values of the target variable is taken into consideration in stratified k-fold.

If for instance, the target variable is a categorical variable with 2 classes, then stratified k-fold ensures that each test fold gets an equal ratio of the two classes when compared to the training set.

This makes the model evaluation more accurate and the model training less biased.

**Bootstrap**

Bootstrap is one of the most powerful ways to obtain a stabilized model. It is close to the random splitting technique since it follows the concept of random sampling.

The first step is to select a sample size (which is usually equal to the size of the original dataset). Thereafter, a sample data point must be randomly selected from the original dataset and added to the bootstrap sample. After the addition, the sample needs to be put back into the original sample. This process needs to be repeated for N times, where N is the sample size.

**Probabilistic measures**

Probabilistic Measures do not just take into account the model performance but also the model complexity. Model complexity is the measure of the model’s ability to capture the variance in the data.

For example, a highly biased model like the linear regression algorithm is less complex and on the other hand, a neural network is very high on complexity.

**Akaike Information Criterion (AIC)**

It is common knowledge that every model is not completely accurate. There is always some information loss which can be measured using the KL information metric. Kulback-Liebler or KL divergence is the measure of the difference in the probability distribution of two variables.

1. **Understanding BLEU:**  BLEU (Bilingual Evaluation Understudy) is a measurement of the differences between an automatic translation and one or more human-created reference translations of the same source sentence.

The BLEU algorithm compares consecutive phrases of the automatic translation with the consecutive phrases it finds in the reference translation, and counts the number of matches, in a weighted fashion. These matches are position independent. A higher match degree indicates a higher degree of similarity with the reference translation, and higher score. Intelligibility and grammatical correctness are not taken into account.

BLEU’s strength is that it correlates well with human judgment by averaging out individual sentence judgment errors over a test corpus, rather than attempting to devise the exact human judgment for every sentence.

We look at the **adequacy, fluency, and fidelity of the translations** to know it’s effectiveness.

**Adequacy**is a measure to know if all the meaning was expressed from source language to the target language

**Fidelity** is the extent to which a **translation** accurately renders the meaning of the source text

**Fluency**measures how grammatically well-formed the sentences are along with ease of interpretation.

BLEU compares the n-gram of the candidate translation with n-gram of the reference translation to count the number of matches. These matches are independent of the positions where they occur.

The more the number of matches between candidate and reference translation, the better is the machine translation.