**Machine Learning Assignment 9**

**1.What is feature engineering, and how does it work? Explain the various aspects of feature engineering in depth.**

Feature engineering is the process that takes raw data and transforms it into features that can be used to create a predictive model using [machine learning](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML) or statistical modeling, such as [deep learning](https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network). The aim of feature engineering is to prepare an input data set that best fits the machine learning algorithm as well as to enhance the performance of machine learning models. Feature engineering [can help data scientists](https://www.techtarget.com/searchcio/tip/How-automating-feature-engineering-can-help-data-scientists) by accelerating the time it takes to extract variables from data, allowing for the extraction of more variables. Automating feature engineering will help organizations and data scientists create models with better accuracy.

**How feature engineering works**

The feature engineering process may look something like this:

* Devise features -- examine a lot of data, analyze feature engineering on other problems and figure out what to use from them.
* Define features -- involves two processes: feature extraction, which consists of defining and extracting a set of features that represent data that's important for the analysis; and feature construction, which entails transforming a particular set of input features to make a new set of more effective features that can be used for prediction. Depending on the problem, users can decide to use automatic feature extraction, manual feature construction or a combination of the two.
* Select features -- when users know something about the data, and they've defined the potential features, the next step is to choose the right features. This consists of two elements: feature selection, the process of selecting some subset of the features most relevant to a particular task; and feature scoring, an assessment of how useful a feature is for prediction.
* Evaluate models -- evaluate features by evaluating the accuracy of the model on unseen data using the selected features.

**Feature engineering techniques**

Feature engineering techniques include:

* Imputation -- a typical problem in machine learning is missing values in the data sets, which affects the way machine learning [algorithms](https://www.techtarget.com/whatis/definition/algorithm) Imputation is the process of replacing missing data with statistical estimates of the missing values, which produces a complete data set to use to train machine learning models.
* One-hot encoding -- a process by which categorical data is converted into a form that the machine learning algorithm understands so it can make better predictions.
* [Bag of words](https://www.techtarget.com/searchenterpriseai/definition/bag-of-words-model-BoW-model) -- a counting algorithm that calculates how many times a word is repeated in a document. It can be used to determine similarities and differences in documents for such applications as search and document classification.
* Automated feature engineering -- this technique pulls out useful and meaningful features using a framework that can be applied to any problem. Automated feature engineering enables data scientists to be more productive by allowing them to spend more time on other components of machine learning. This technique also allows [citizen data scientists](https://www.techtarget.com/searchbusinessanalytics/definition/citizen-data-scientist) to do feature engineering using a framework-based approach.
* Binning -- binning, or grouping data, is key to preparing numerical data for machine learning. This technique can be used to replace a column of numbers with categorical values representing specific ranges.
* N-grams -- help predict the next item in a sequence. In sentiment analysis, the n-gram model helps analyze the sentiment of the text or document.
* Feature crosses -- a way to combine two or more categorical features into one. This technique is particularly useful when certain features together denote a property better than they do by themselves.

There are some open source [Python](https://www.techtarget.com/whatis/definition/Python) libraries that support feature engineering techniques, including the Featuretools library for automatically creating features out of a set of related tables using deep feature synthesis, an algorithm that automatically generates features for relational data sets.

**2. What is feature selection, and how does it work? What is the aim of it? What are the various methods of function selection?**

Feature selection, one of the main components of [feature engineering](https://www.heavy.ai/technical-glossary/feature-engineering), is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model.

The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important, include:

* simpler models: simple models are easy to explain - a model that is too complex and unexplainable is not valuable
* shorter training times: a more precise subset of features decreases the amount of time needed to train a model
* variance reduction: increase the precision of the estimates that can be obtained for a given simulation
* avoid the curse of high dimensionality: dimensionally cursed phenomena states that, as dimensionality and the number of features increases, the volume of space increases so fast that the available data become limited - PCA feature selection may be used to reduce dimensionality

Feature selection techniques can be divided into two types: supervised and unsupervised. Supervised methods may be divided into three types: wrapper methods (forward, backward, and stepwise selection), filter methods (ANOVA, Pearson correlation, variance thresholding), and embedded methods (Lasso, Ridge, Decision Tree).

## Wrapper methods

Wrapper methods are used to train a model using a subset of features. According to the conclusions drawn from the previous model, we decide whether to include or exclude certain features from the subgroup. The problem is essentially reduced to a search problem and usually has a high computational cost.

For example, standard wrapper methods are forward feature selection, backward feature elimination, recursive feature elimination, etc.

* **Forward Selection**  
  In bold selection, start with no feature in the model and iterate forward. On each subsequent repetition, add the feature that improves the model the most until the addition of a new variable does not improve its performance.
* **Backward Elimination**  
  Start with all the features during backward elimination and remove the least significant feature at every iteration, which improves the model's performance. This process is repeated until no improvement is observed.
* **Recursive Feature Elimination**  
  The algorithm aims to find the best performing subset of features using greedy optimization. Each iteration creates a new model and keeps aside the best or worst performing features. The next model is constructed using the remaining features until all the features have been exhausted. The features are then ranked according to their elimination order.

With wrapper methods, it's best to use the Boruta package, which identifies the importance of a feature by creating shadow features.

## Filter methods

Filter methods are generally used as preprocessing steps, and their selection is independent of any machine learning algorithm. Features are instead selected based on their scores in various statistical tests to determine their correlation with the outcome variable, which is subjective.

* **Pearson’s Correlation**  
  It is used to quantify linear dependence between two continuous variables, X and Y. Its value ranges from -1 to 1.
* **LDA**  
  A linear discriminant analysis determines a linear combination of features that characterizes or differentiates between a categorical variable’s two or more classes (or levels).
* **ANOVA**  
  ANOVA stands for analysis of variance. Its operation is similar to LDA, except it is based on one or more categorical independent features and one continuous dependent feature. The test determines if the means of several groups are equal or not.
* **Chi-Square**  
  This is a statistical test used to determine whether there is a correlation between categorical features based on their frequency distributions.

It is important to remember that filter methods do not remove multicollinearity. Before training models for your data, you must consider the multicollinearity of features.

## Embedded methods

Embedded methods combine the best features of filtering and wrapping by implementing algorithms with built-in methods for selecting features.

For example, RIDGE and LASSO regression both have inbuilt penalization functions that can reduce overfitting.

* The L1 regularization of Lasso regression adds a penalty equal to the absolute value of the coefficients' magnitude.
* Ridge regression performs L2 regularization, which imposes a penalty equal to the square of the coefficients' magnitude.

**3. Describe the function selection filter and wrapper approaches. State the pros and cons of each approach?**

* Filter methods: Filter methods select features based on statistics rather than feature selection cross-validation performance. A selected metric is applied to identify irrelevant attributes and perform recursive feature selection. Filter methods are either univariate, in which an ordered ranking list of features is established to inform the final selection of feature subset; or multivariate, which evaluates the relevance of the features as a whole, identifying redundant and irrelevant features.
* Wrapper methods: Wrapper feature selection methods consider the selection of a set of features as a search problem, whereby their quality is assessed with the preparation, evaluation, and comparison of a combination of features to other combinations of features. This method facilitates the detection of possible interactions amongst variables. Wrapper methods focus on feature subsets that will help improve the quality of the results of the clustering algorithm used for the selection. Popular examples include Boruta feature selection and Forward feature selection.
* Filter methods measure the relevance of features by their correlation with dependent variable while wrapper methods measure the usefulness of a subset of feature by actually training a model on it.
* Filter methods are much faster compared to wrapper methods as they do not involve training the models. On the other hand, wrapper methods are computationally very expensive as well.
* Filter methods use statistical methods for evaluation of a subset of features while wrapper methods use cross validation.
* Filter methods might fail to find the best subset of features in many occasions but wrapper methods can always provide the best subset of features.
* Using the subset of features from the wrapper methods make the model more prone to overfitting as compared to using subset of features from the filter methods.

**4.i. Describe the overall feature selection process.**

**ii. Explain the key underlying principle of feature extraction using an example. What are the most widely used function extraction algorithms?**

Feature Selection Models

Feature selection models are of two types:

1. Supervised Models: Supervised feature selection refers to the method which uses the output label class for feature selection. They use the target variables to identify the variables which can increase the efficiency of the model
2. Unsupervised Models: Unsupervised feature selection refers to the method which does not need the output label class for feature selection. We use them for unlabelled data.

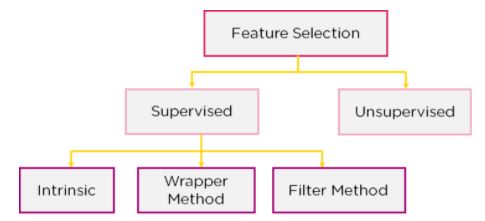


Figure 4: Feature Selection Models

We can further divide the supervised models into three :

1. Filter Method: In this method, features are dropped based on their relation to the output, or how they are correlating to the output. We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly. Eg: Information Gain, [Chi-Square Test](https://www.simplilearn.com/tutorials/statistics-tutorial/chi-square-test), Fisher’s Score, etc.

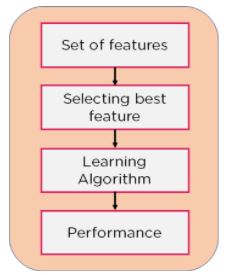


Figure 5: Filter Method flowchart

2. Wrapper Method: We split our data into subsets and train a model using this. Based on the output of the model, we add and subtract features and train the model again. It forms the subsets using a greedy approach and evaluates the accuracy of all the possible combinations of features. Eg: Forward Selection, Backwards Elimination, etc.

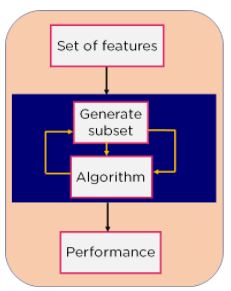


                                             Figure 6: Wrapper Method Flowchart

3. Intrinsic Method: This method combines the qualities of both the Filter and Wrapper method to create the best subset.

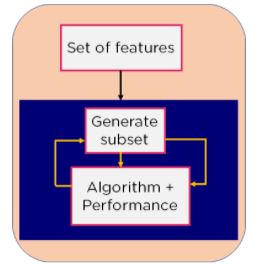


                                                Figure 7: Intrinsic Model Flowchart

This method takes care of the machine training iterative process while maintaining the computation cost to be minimum. Eg: Lasso and Ridge Regression.

**5. Describe the feature engineering process in the sense of a text categorization issue.**

In the context of text categorization, feature engineering is the process of extracting relevant information from the text data and transforming it into a form that can be used as input for a machine learning model. The specific steps of the process may vary depending on the specific task and dataset, but a general outline of the process is as follows:

Data preprocessing: This step involves cleaning and preparing the text data for feature extraction. This may include tasks such as removing special characters, lowercasing all text, removing stop words, and stemming or lemmatizing the words.

Feature extraction: This is the core step of feature engineering, where relevant information is extracted from the text data. Common techniques include bag of words, n-grams, and term frequency-inverse document frequency (TF-IDF).

Feature selection: After extracting the features, it is important to select the most relevant and informative ones for the task at hand. This can be done using techniques such as mutual information, chi-squared test, and LASSO.

Feature transformation: The final step is to transform the extracted features into a format that can be used as input for the machine learning model. This may include vectorization techniques such as one-hot encoding or word embeddings.

Model building : After feature engineering, we can use machine learning algorithms to train models on the features and predict the class of unseen texts.Bottom of Form

**6. What makes cosine similarity a good metric for text categorization? A document-term matrix has two rows with values of (2, 3, 2, 0, 2, 3, 3, 0, 1) and (2, 1, 0, 0, 3, 2, 1, 3, 1). Find the resemblance in cosine.**

Cosine similarity is a good metric for text categorization because it takes into account the angle between two vectors, rather than the Euclidean distance. This is particularly useful in the context of text data, where the length of the documents can vary greatly.

To find the cosine similarity between the two documents represented by the document-term matrix, you would first need to convert the rows into vectors. Let's call the first vector A and the second vector B. To find the cosine similarity, you would use the following formula:

cos(theta) = (A . B) / (||A|| \* ||B||)

where A . B is the dot product of the two vectors, and ||A|| and ||B|| are the magnitudes of the vectors.

In this case, the dot product of the vectors is (22) + (31) + (20) + (00) + (23) + (32) + (31) + (03) + (1\*1) = 20

The magnitude of the first vector is sqrt((22) + (33) + (22) + (00) + (22) + (33) + (33) + (00) + (1\*1)) = sqrt(60)

The magnitude of the second vector is sqrt((22) + (11) + (00) + (00) + (33) + (22) + (11) + (33) + (1\*1)) = sqrt(24)

So the cosine similarity between the two vectors is (20) / (sqrt(60) \* sqrt(24)) = 20/ (sqrt(1440)) = 20/ (sqrt(6024)) = 20/ (sqrt(60212)) = 20/ (sqrt(1440)) = 20/ (122sqrt(5)) = 20/ (24sqrt(5)) = 5/sqrt(5) = 1

Therefore the cosine similarity is 1, meaning the two vectors point in the same direction, indicating the two documents are similar

**7.i. What is the formula for calculating Hamming distance? Between 10001011 and 11001111, calculate the Hamming gap.**

The formula for calculating Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In other words, it is the number of substitutions required to change one string into the other.

To calculate the Hamming distance between 10001011 and 11001111:

Compare the first bit, they are different so we add 1 to the distance.

Compare the second bit, they are the same so we add 0 to the distance.

Compare the third bit, they are different so we add 1 to the distance.

Compare the fourth bit, they are the same so we add 0 to the distance.

Compare the fifth bit, they are different so we add 1 to the distance.

Compare the sixth bit, they are the same so we add 0 to the distance.

Compare the seventh bit, they are different so we add 1 to the distance.

Compare the eighth bit, they are different so we add 1 to the distance.

So the Hamming distance between 10001011 and 11001111 is 1+0+1+0+1+0+1+1 = 5. The Hamming gap is 5.

**ii. Compare the Jaccard index and similarity matching coefficient of two features with values (1, 1, 0, 0, 1, 0, 1, 1) and (1, 1, 0, 0, 0, 1, 1, 1), respectively (1, 0, 0, 1, 1, 0, 0, 1).**

The Jaccard index is a measure of similarity between two sets, defined as the size of the intersection divided by the size of the union of the sets. In this case, the intersection of the two sets is {1,1,0,0}, and the union is {1,1,0,0,1,0,1,1}. Therefore, the Jaccard index for these two sets is 4/8 = 0.5.

The similarity matching coefficient (SMC) is another measure of similarity between two sets, defined as the size of the intersection divided by the minimum of the sizes of the sets. In this case, the intersection of the two sets is still {1,1,0,0}, and the minimum size of the sets is 8. Therefore, the SMC for these two sets is 4/8 = 0.5.

In this case, both Jaccard index and similarity matching coefficient are the same, which is 0.5.

**8. State what is meant by “high-dimensional data set”? Could you offer a few real-life examples?**

**What are the difficulties in using machine learning techniques on a data set with many dimensions?What can be done about it?**

A high-dimensional data set is one that has a large number of features or variables. Real-life examples of high-dimensional data sets could include:

A dataset of images, where each image has many pixels, and each pixel is a feature.

A dataset of customer data, where each customer has many different attributes such as age, income, location, purchase history, etc.

A dataset of genetic data, where each individual has many different genetic variations, each represented as a feature.

In all of these examples, the number of features is much larger than the number of data points, making the data high-dimensional. High-dimensional data can be challenging to work with, as traditional statistical methods and machine learning algorithms may not perform well in this setting.

There are several difficulties in using machine learning techniques on a data set with many dimensions, some of which include:

The "curse of dimensionality": As the number of dimensions increases, the amount of data required to accurately model the underlying relationships in the data also increases. This can make it difficult to find patterns in high-dimensional data.

**Overfitting:** With a large number of dimensions, it's easy for a model to fit to noise in the data rather than the underlying patterns. This can lead to poor generalization performance on new data.

**Computational challenges**: Many machine learning algorithms are computationally intensive, and the computational cost can increase exponentially with the number of dimensions.

There are several ways to address these difficulties, some of which include:

**Dimensionality reduction**: Techniques such as principal component analysis (PCA) and linear discriminant analysis (LDA) can be used to reduce the number of dimensions in a data set, while still retaining the most important information.

**Regularization:** Techniques such as L1 and L2 regularization can help to prevent overfitting by adding a penalty term to the cost function that discourages the model from fitting to noise in the data.

**Ensemble methods**: Bagging and boosting are ensemble methods that can help to reduce the overfitting caused by high-dimensional data by averaging the predictions of multiple models.

**Sampling:** It is possible to select a random subset of the data, which can reduce the computational complexity without losing much information.

**Neural network architectures**: Some architectures like autoencoder, deep belief network, and variational autoencoder are designed to handle high dimensional data, as they are able to learn compact representations of the data.

**9. Make a few quick notes on:**

**PCA is an acronym for Principal Component Analysis.**

PCA (Principal Component Analysis) is a mathematical technique for dimensionality reduction, not Personal Computer Analysis. It is a linear method that transforms the original high-dimensional data into a new set of linearly uncorrelated variables called principal components. These principal components are ordered so that the first component has the largest possible variance, and each subsequent component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. This can help to identify patterns in the data and make it easier to visualize and analyze. It is widely used in many fields such as image processing, natural language processing, genomics, and bioinformatics.

**2. Use of vectors**

Vectors are mathematical objects that can be used to represent data in machine learning. They are used to represent data in a high-dimensional space, where each dimension corresponds to a feature or variable. Vectors can be used to represent data in a number of ways in machine learning, such as:

**Feature representation**: Vectors can be used to represent data points as a set of features or variables. This can be used as input to a machine learning algorithm, which can then learn to make predictions based on the values of these features.

**Embedding:** Vectors can be used to represent data points in a low-dimensional space, known as an embedding. This can be used to reduce the dimensionality of the data, making it easier to visualize and analyze.

**Similarity:** Vectors can be used to calculate the similarity between data points based on their positions in the high-dimensional space. This can be used to cluster similar data points together or to find nearest neighbors.

**Linear algebra**: Vectors can be used to perform linear algebraic operations such as dot product, cross product, and matrix multiplication. These operations are widely used in machine learning to perform linear transformations, such as in PCA, LDA and neural network.

**Gradient Descent**: Vectors can be used to represent the parameters of a machine learning model as a vector, which makes it easy to perform gradient descent optimization.

In addition to these examples, vectors are used in many other ways in machine learning, and are a fundamental tool in the field.

**3. Embedded technique**

Embedding is a technique used in machine learning and natural language processing to represent data points, such as words or documents, in a low-dimensional space. The goal of embedding is to map the data points to a new space where the similarity between data points is preserved, making it easier to perform tasks such as text classification, language translation, and information retrieval.

There are several types of embedding techniques:

**Word embedding:** This is a technique used to map words to a low-dimensional space, where the similarity between words is preserved. Word2Vec and GloVe are popular methods for learning word embeddings.

**Sentence embedding**: This technique maps a sentence or a document to a low-dimensional space, where the similarity between sentences is preserved. Sentence embeddings such as Universal Sentence Encoder are commonly used in text classification and information retrieval tasks.

**Image embedding**: This technique maps an image to a low-dimensional space, where the similarity between images is preserved. This is useful for tasks such as image retrieval and image classification.

**Graph embedding**: This technique maps the nodes of a graph to a low-dimensional space, where the similarity between nodes is preserved. This is useful for tasks​

**10. Make a comparison between:**

**1. Sequential backward exclusion vs. sequential forward selection**

Sequential backward elimination and sequential forward selection are two different feature selection methods used to select a subset of features from a given set of features.

Sequential backward elimination (SBE) is a technique that starts with all the features and iteratively removes the feature that has the smallest effect on the performance of a model. This process is repeated until a satisfactory subset of features is obtained. The main advantage of SBE is that it can be used with a wide range of models, and it can be used to find the optimal subset of features. However, the main disadvantage of SBE is that it does not consider the interactions between features.

Sequential forward selection (SFS) is a technique that starts with an empty set of features and iteratively adds the feature that has the largest effect on the performance of a model. This process is repeated until a satisfactory subset of features is obtained. The main advantage of SFS is that it can be used to identify interactions between features. However, the main disadvantage of SFS is that it can be computationally expensive, as it requires fitting the model for each subset of features.

Both methods have their own advantages and disadvantages depending on the problem, the data and the model. While SBE is more widely used

**2.Function selection methods: filter vs. wrapper**

Feature selection methods can be broadly divided into two categories: filter methods and wrapper methods.

Filter methods are feature selection techniques that are independent of any specific machine learning algorithm. They use a statistical measure, such as correlation or mutual information, to evaluate the relevance of each feature and select a subset of features accordingly. These methods are typically fast and computationally efficient, but they may not always select the best subset of features for a specific machine learning algorithm. Examples of filter methods include mutual information, chi-squared test and ANOVA.

Wrapper methods, on the other hand, evaluate the performance of a specific machine learning algorithm with different subsets of features. The subset of features that results in the best performance is selected. These methods are generally more computationally expensive than filter methods, but they can select the best subset of features for a specific machine learning algorithm. Examples of wrapper methods include forward selection, backward elimination and recursive feature elimination.

In summary, filter methods use statistical measures to evaluate features independently of the model, while wrapper methods evaluate the features by training a specific model with different subsets of features. Filter methods are computationally efficient and can be used as a quick way to remove irrelevant features, while wrapper methods are computationally expensive but provide more accurate results.

**3.SMC vs. Jaccard coefficient**

SMC and Jaccard coefficient are two different measures used to quantify the similarity or dissimilarity between two sets of data.

SMC (Sørensen-Dice coefficient) is a similarity measure that compares the number of items that two sets have in common to the total number of items in both sets. It is defined as:

SMC = 2\*(|A∩B|) / (|A| + |B|)

Where A and B are the two sets being compared, and |A| and |B| are the number of elements in each set. SMC ranges between 0 and 1, where 1 represents an exact match and 0 represents no similarity.

Jaccard coefficient is another similarity measure used to quantify the similarity between two sets. It is defined as the size of the intersection of the two sets divided by the size of the union of the two sets. It is defined as:

Jaccard = |A∩B| / |A∪B|

Where A and B are the two sets being compared. Jaccard coefficient ranges between 0 and 1, where 1 represents an exact match and 0 represents no similarity.

Both SMC and Jaccard coefficient are commonly used in data mining and machine learning to compare sets of data, and to identify patterns and structures in the data. They are both commonly used for measuring the similarity between sets, and have similar properties, but they may lead to different results depending on the size of the sets and the nature of the data.