**Phase4 :Development part2**

**Registrar of Company:**

Registrars of Companies (ROC) appointed under Section 609 of the Companies Act covering the various States and Union Territories are vested with the primary duty of registering companies and LLPs floated in the respective states and the Union Territories and ensuring that such companies and LLPs comply with statutory requirements under the Act. These offices function as registry of records, relating to the companies registered with them, which are available for inspection by members of public on payment of the prescribed fee. The Central Government exercises

**The different feature engineering techniques examples:**

In this blog, we will look at the following feature engineering techniques and understand their implementations:

* Scaling.
* Normalization.
* Standardization.
* One hot encoding.
* Ordinal Encoding.
* Bucketing/Binning.
* Bag of words.
* Derived Features.

**Feature based modeling techniques:**

Feature-based modeling is the traditional and predominant method of creating 3D models in CAD/CAM software, which requires defining the geometry and topology of the model by adding and subtracting features such as sketches, extrusions, fillets, holes, and patterns.

**What is feature engineering?**

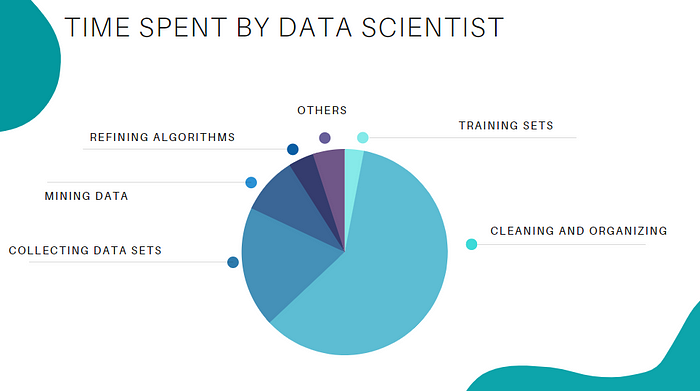
Feature engineering is a machine learning technique that leverages data to create new variables that aren’t in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of **simplifying and speeding up data transformations** while also **enhancing model accuracy**. Feature engineering is required when working with machine learning models. Regardless of the data or architecture, a terrible feature will have a direct impact on your model.

**Feature engineering consists of various process :**

* **Feature Creation**: Creating features involves creating new variables which will be most helpful for our model. This can be adding or removing some features. As we saw above, the cost per sq. ft column was a feature creation.
* **Transformations**: Feature transformation is simply a function that transforms features from one representation to another. The goal here is to plot and visualise data, if something is not adding up with the new features we can reduce the number of features used, speed up training, or increase the accuracy of a certain model.
* **Feature Extraction**: Feature extraction is the process of extracting features from a data set to identify useful information. Without distorting the original relationships or significant information, this compresses the amount of data into manageable quantities for algorithms to process.
* **Exploratory Data Analysis :**Exploratory data analysis (EDA) is a powerful and simple tool that can be used to improve your understanding of your data, by exploring its properties. The technique is often applied when the goal is to create new hypotheses or find patterns in the data. It’s often used on large amounts of qualitative or quantitative data that haven’t been analyzed before.
* **Benchmark:**A Benchmark model is the end of this article. Now, let’s have a look at why we need feature engineering in machine learning. : A Benchmark Model is the most user-friendly, dependable, transparent, and interpretable model against which you can measure your own. It’s a good idea to run test datasets to see if your new machine learning model outperforms a recognised benchmark. These benchmarks are often used as measures for comparing the performance between different machine learning models like neural networks and support vector machines, linear and non-linear classifiers, or different approaches like bagging and boosting. To learn more about feature engineering steps and process, check the links provided at

# **Importance Of Feature Engineering:**

Feature Engineering is a very important step in machine learning. Feature engineering refers to the process of designing artificial features into an algorithm. These artificial features are then used by that algorithm in order to improve its performance, or in other words reap better results. Data scientists spend most of their time with data, and it becomes important to make models accurate.



When feature engineering activities are done correctly, the resulting dataset is optimal and contains all of the important factors that affect the business problem. As a result of these datasets, the most accurate predictive models and the most useful insights are produced.

# **Feature Engineering Techniques for Machine Learning:**

Lets see a few feature engineering best techniques that you can use. Some of the techniques listed may work better with certain algorithms or datasets, while others may be useful in all situations.

**1.Imputation**

When it comes to preparing your data for machine learning, missing values are one of the most typical issues. Human errors, data flow interruptions, privacy concerns, and other factors could all contribute to missing values. Missing values have an impact on the performance of machine learning models for whatever cause. The main goal of imputation is to handle these missing values. :

* **Numerical Imputation**: To figure out what numbers should be assigned to people currently in the population, we usually use data from completed surveys or censuses. These data sets can include information about how many people eat different types of food, whether they live in a city or country with a cold climate, and how much they earn every year. That is why numerical imputation is used to fill gaps in surveys or censuses when certain pieces of information are missing.

*#Filling all missing values with 0*

*data = data.fillna(0)*

* **Categorical Imputation:**When dealing with categorical columns, replacing missing values with the highest value in the column is a smart solution. However, if you believe the values in the column are evenly distributed and there is no dominating value, imputing a category like “Other” would be a better choice, as your imputation is more likely to converge to a random selection in this scenario.

*#Max fill function for categorical columns*

*data[‘column\_name’].fillna(data[‘column\_name’].value\_counts().idxmax(), inplace=True)*

**2.Handling Outliers**

Outlier handling is a technique for removing outliers from a dataset. This method can be used on a variety of scales to produce a more accurate data representation. This has an impact on the model’s performance. Depending on the model, the effect could be large or minimal; for example, linear regression is particularly susceptible to outliers. This procedure should be completed prior to model training. The various methods of handling outliers include:

1. **Removal**: Outlier-containing entries are deleted from the distribution. However, if there are outliers across numerous variables, this strategy may result in a big chunk of the datasheet being missed.
2. **Replacing values**: Alternatively, the outliers could be handled as missing values and replaced with suitable imputation.
3. **Capping**: Using an arbitrary value or a value from a variable distribution to replace the maximum and minimum values.
4. **Discretization :**Discretization is the process of converting continuous variables, models, and functions into discrete ones. This is accomplished by constructing a series of continuous intervals (or bins) that span the range of our desired variable/model/function.

**3.Log Transform**

Log Transform is the most used technique among data scientists. It’s mostly used to turn a skewed distribution into a normal or less-skewed distribution. We take the log of the values in a column and utilise those values as the column in this transform. It is used to handle confusing data, and the data becomes more approximative to normal applications.

*//Log Example*

*df[log\_price] = np.log(df[‘Price’])*

**4.One-hot encoding**

A one-hot encoding is a type of encoding in which an element of a finite set is represented by the index in that set, where only one element has its index set to “1” and all other elements are assigned indices within the range [0, n-1]. In contrast to binary encoding schemes, where each bit can represent 2 values (i.e. 0 and 1), this scheme assigns a unique value for each possible case.

**5.Scaling**

Feature scaling is one of the most pervasive and difficult problems in machine learning, yet it’s one of the most important things to get right. In order to train a predictive model, we need data with a known set of features that needs to be scaled up or down as appropriate. This blog post will explain how feature scaling works and why it’s important as well as some tips for getting started with feature scaling.

After a scaling operation, the continuous features become similar in terms of range. Although this step isn’t required for many algorithms, it’s still a good idea to do so. Distance-based algorithms like k-NN and k-Means, on the other hand, require scaled continuous features as model input. There are two common ways for scaling :

**Normalization**: All values are scaled in a specified range between 0 and 1 via normalisation (or min-max normalisation). This modification has no influence on the feature’s distribution, however it does exacerbate the effects of outliers due to lower standard deviations. As a result, it is advised that outliers be dealt with prior to normalisation.

**Standardization**: Standardization (also known as z-score normalisation) is the process of scaling values while accounting for standard deviation. If the standard deviation of features differs, the range of those features will likewise differ. The effect of outliers in the characteristics is reduced as a result. To arrive at a distribution with a 0 mean and 1 variance, all the data points are subtracted by their mean and the result divided by the distribution’s variance.

# **FeatureTools**

Featuretools is a framework to perform automated feature engineering. It excels at transforming temporal and relational datasets into feature matrices for machine learning. Featuretools integrates with the machine learning pipeline-building tools you already have. In a fraction of the time it would take to do it manually, you can load in pandas dataframes and automatically construct significant features.

**FeatureTools Summary**

* Easy to get started, good documentation and community support
* It helps you construct meaningful features for machine learning and predictive modelling by combining your raw data with what you know about your data.
* It provides APIs to verify that only legitimate data is utilised for calculations, preventing label leakage in your feature vectors.
* Featuretools includes a low-level function library that may be layered to generate features.
* Its AutoML library(EvalML) helps you build, optimize, and evaluate machine learning pipelines.
* Good at handling relational databases.

**FeatureTools Summary**

* Easy to get started, good documentation and community support
* It helps you construct meaningful features for machine learning and predictive modelling by combining your raw data with what you know about your data.
* It provides APIs to verify that only legitimate data is utilised for calculations, preventing label leakage in your feature vectors.
* Featuretools includes a low-level function library that may be layered to generate features.
* Its AutoML library(EvalML) helps you build, optimize, and evaluate machine learning pipelines.
* Good at handling relational databases.

**AutoFeat Summary**

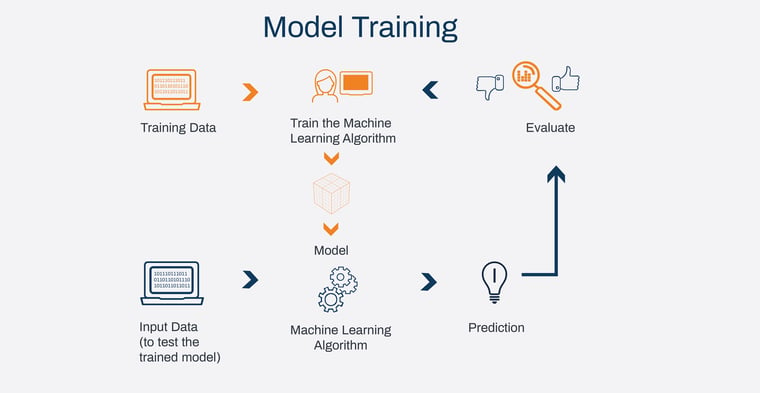
* AutoFeat can easily handle categorical features with One hot encoding.
* The AutoFeatRegressor and AutoFeatClassifier models in this package have a similar interface to scikit-learn models
* General purpose automated feature engineering which is Not good at handling relational data.
* It is useful in logistical data

## Model training

Writing and running machine learning algorithms to produce an ML model is the central component of the ML workflow. A data science team typically uses the model engineering pipeline, which consists of several procedures, such as model testing, model evaluation and model packaging, to create the final model.

You can streamline these activities in several ways. For example, you can automate the machine learning model training process by building a pipeline, which makes it simpler to scale the solution to larger datasets and maintain and update the model over time.

It is important to understand how crucial and interconnected the stages of model training, evaluation and testing are in the machine-learning workflow. Model creation is followed by model training, assessment of the model’s performance on a different dataset and testing of the model on fresh or previously unexplored data. Since this process is iterative, it may be necessary to repeat model training several times before the model’s performance on the testing data is acceptable.



**The model training phase includes these steps/actions:**

* Step1:Depending on the data you need and your learning objectives, choose the appropriate algorithm.
* Step2:Choose the architecture, model variant or other parameters that will produce the best results.
* Step3:Set up, fine-tune and create the model parameter values that a machine learning algorithm eventually learns. The term “hyperparameter” refers to this regulatory mechanism. In order to get the best results, the model version is chosen with the aid of this hyperparameter adjustment. Examples include the number of layers, activation function and learning rate in neural networks, which are all controlled via fine-tuning.
* Step4:To demonstrate which hyperparameters and models are most effective for your use case, benchmark them.
* Step5:Determine whether the model has the necessary level of explainability.
* Step6Consider using an ensemble technique, which involves running multiple models concurrently, if applicable, or a more advanced technique if required by the business challenges or objective.

**Here are a few typical training approaches.**

* **Grid search:** Training the model with all conceivable combinations of hyperparameters while specifying a range of values for each one.
* **Random search:** Using a set of arbitrarily chosen hyperparameter values that fall within a predetermined range.
* **Bayesian optimization:** This method employs a probabilistic model to forecast how various hyperparameter values will perform and to pick the most promising ones for training.
* **Genetic algorithms:** To discover the ideal collection of hyperparameters, genetic algorithms evolve a population of potential hyperparameters over several generations.
* **Manual tuning:** Testing various hyperparameter values and evaluating the model’s performance.

**Model evaluation**

Model evaluation measures how well a trained machine learning model works to make sure it meets the original business objectives. The goal of model evaluation is to assess a model’s ability to predict outcomes correctly and to pinpoint areas for improvement.  To assess a model, a variety of methods can be applied, such as:

* **Holdout technique:** Data is divided into training and test sets. The model is developed on the training set, and the test set is used to assess the model’s success.
* **Bootstrapping:** In this method, the model undergoes training using a series of newly generated datasets. These datasets are crafted by resampling the original dataset, allowing the same data point to appear multiple times within a resampled dataset. By employing this method of replacement, the model's performance is evaluated based on the insights gathered from these resampled datasets.
* **Cross-validation:** In this method, the data is divided into different subgroups, each of which is used as a test set while the other subsets are used for training. This technique aids in lowering the danger of over-fitting which can lead to poor predictive performance.
* **Metrics:**Depending on the kind of issue being solved, different metrics can be used to assess how well a machine learning model is performing. Accuracy, precision, recall, F1 score, AUC-ROC and mean squared error are a few typical measures.
* **Visual inspection:** In some circumstances, the output of the model can be visually inspected to assess the success of the model. For instance, in image classification, the model’s success can be evaluated by comparing its predictions to the labels applied.

### **Model testing**

Model testing in machine learning is the process of assessing how well a trained model performs on a collection of data that it has never seen before. Model testing is done to determine how well a model generalizes to new, unforeseen data and to predict how well it will work in practice.

To prevent bias, the testing set should not be used in the training process and should be an accurate representation of the real-world data that the model will meet. To provide a statistically significant evaluation of the model’s performance, the testing set must be big enough. Decisions about the model’s suitability for deployment in the actual world are then made based on the outcomes of the model testing stage.

### **Model packaging**

Model packaging is the practice of combining a trained machine learning model with its related pre- and post-processing steps, configuration files and other essential resources into a unique package that can be readily distributed, deployed and used by others. The trained model can be shared and used in various contexts, such as cloud-based or on-premises systems, thanks to model packaging in the machine learning workflow. Depending on the unique requirements of the use case, a model can be packaged using a variety of techniques.

In this blog, we touched upon model training, evaluation and testing. The features of ML testing include the need to verify the quality of the data as well as the model and the need to iteratively tune the hyperparameters to achieve the best results. You can be certain of its performance if you follow all the steps outlined.