Features Engineering

The algorithms we used are very standard for Kagglers. We spent most of our efforts in feature engineering.

By: Fares Hasan

What is features engineering?

The process of creating and deriving new features using domain knowledge to enhance the machine learning algorithm performance. If done right it will maximize the productivity and robustness of the learning.

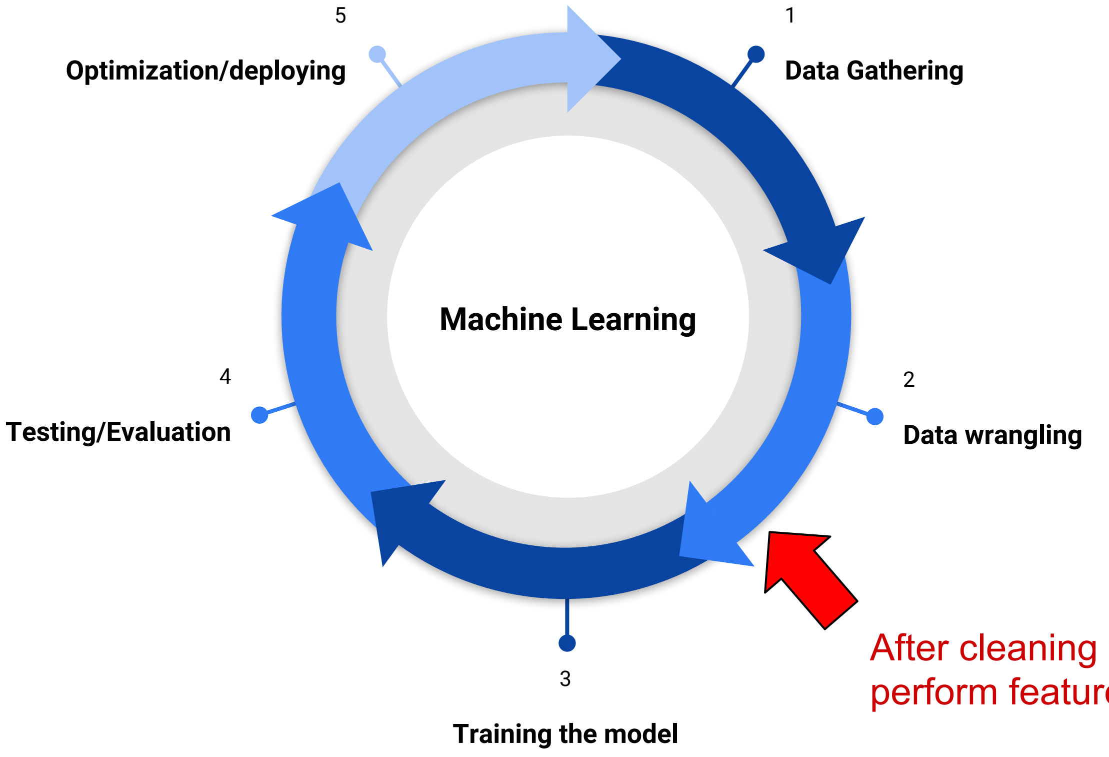
Features engineering is a work of art.

What can you engineer?

- Images will have known methods and techniques inspired by computer vision and statistics. You can experiment only to know what could work for you.
- Text have features drawn from Natural Language Processing/Understanding (NLP/NLU). Where the representation of the texts will form features.
- Sound/Signals(brain & heart) will have absolutely different way of modeling and deriving features.

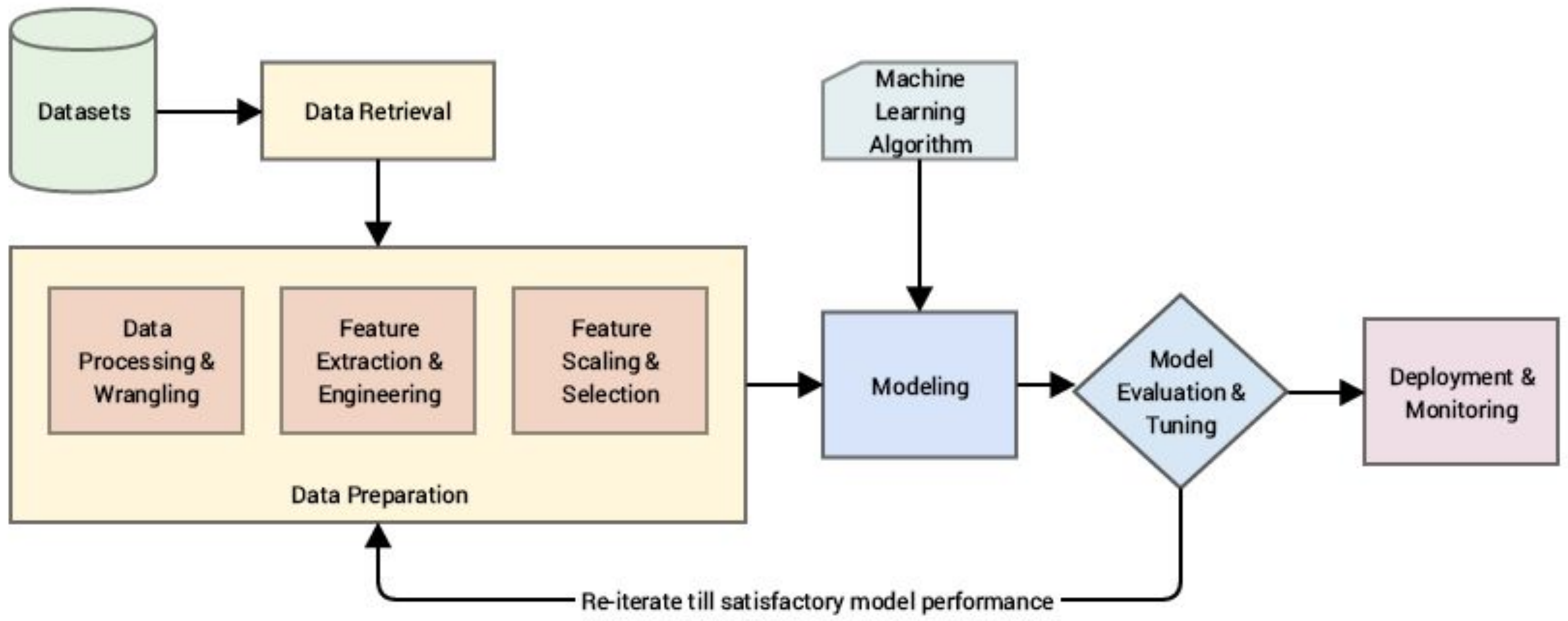
Use your artistic sense and math knowledge to program your masterpiece features.

Where in the ML workflow?



After cleaning your data you can perform features engineering.

Typical Machine Learning Pipeline


Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance

A standard machine learning pipeline (source: Practical Machine Learning with Python, Apress/Springer)

Do we have to ?

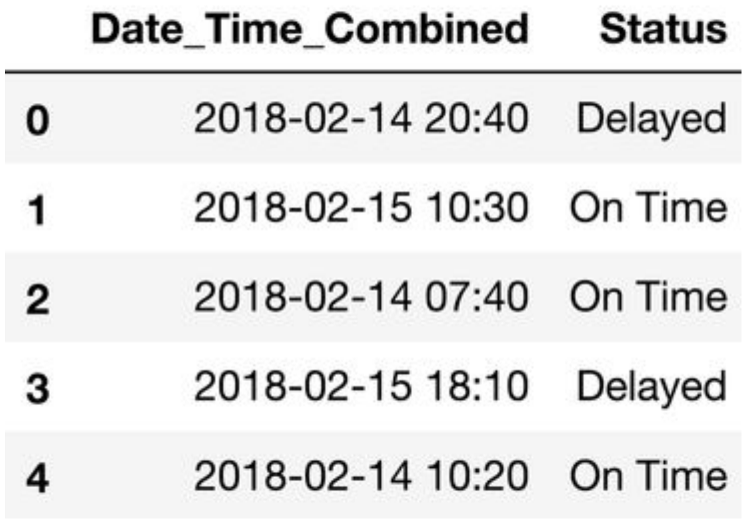
In every case and machine learning problem the initial features are sufficient to experiment and test the model performance. But quite often the performance needs improvement and or the model performance hindered by variance or bias.

These problems can have multiple causes and sometimes the lack of meaningful features in the dataset or the features being insufficient for the model to learn. Therefore comes the need to engineer new features that can experimentally reduce the problem.

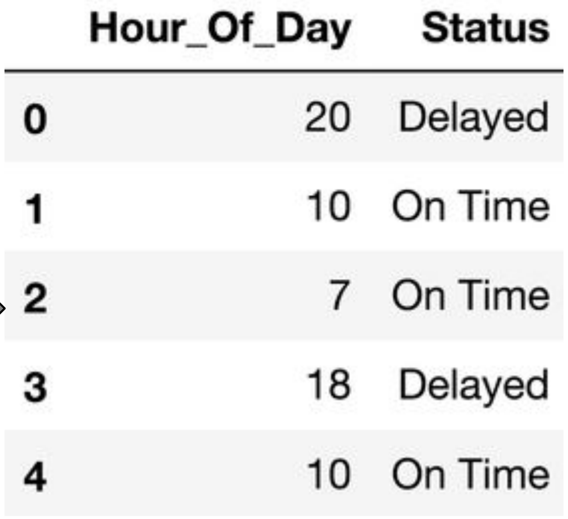
Prof. Andrew Ng

“Coming up with features is difficult, time-consuming, requires expert knowledge. ‘Applied machine learning’ is basically feature engineering.”

Example



	Date_Time_Combined	Status
0	2018-02-14 20:40	Delayed
1	2018-02-15 10:30	On Time
2	2018-02-14 07:40	On Time
3	2018-02-15 18:10	Delayed
4	2018-02-14 10:20	On Time



	Hour_Of_Day	Status
0	20	Delayed
1	10	On Time
2	7	On Time
3	18	Delayed
4	10	On Time

Steps

- Brainstorm features.
- Create features.
- Check how the model performs with the new features (train a model and evaluate).
- Try new features and see which combination performs well.

Tips: always make sure you test if the new features satisfy the assumptions of the algorithm.

Features Engineering Techniques

- Percentiles/ bucketing.
- Parts of time (the hour of the day).
- Flagging holidays.
- Serializing categorical values into numeric values.
- Reading images as RGB pixels and convert into grayscale.

There advanced techniques that employ math and statistics.

Feature Selection

Simply put selecting certain features and eliminate others is considered part of the features engineering via selection process.

It the process of selecting relevant subset of features to our prediction problem.

Feature selection is different from dimensionality reduction. Even though both methods reduce the dimensions of the features but dimensionality reduction creates new combination of features whereas feature selection it only include and exclude features in the present dataset.

What problem feature selection solve?

It helps in better learning performance for the model with less number of features. It helps in identify and remove irrelevant and redundant features that don’t contribute to a better performance for the model. In general it helps in:

- Reduce overfitting.
- Improve accuracy.
- Reduce training.

Fewer attributes is desirable because it reduces the complexity for the model.

Source

Feature selection methods:

Filtering method: scoring the features based on a statistical measure and rank the features based on the scores. Some of statistical tests for scoring are pearson's correlation, chi-square, or entropy (information gain).

Wrapper method: treats the selection process as a search problem. Several Combinations of the features will be produced and the predictive model will be trained on each combination and then compare between the results from each set. An example of this method is the Recursive Feature Elimination algorithm (RFE).

Embedded methods: learning which feature contribute more to the accuracy of the model while the model is being created. Regularization methods are an example (LASSO, Elastic Net and Ridge Regression.)

Model Evaluation & Performance Analysis

How to evaluate our model?

What measure to use?

What is the objective state?

How to improve the performance of the model?

The desirable state to achieve: Good Generalization.

We consider the model to perform well if and only if it can generalize well.

Two states the model could fall into:

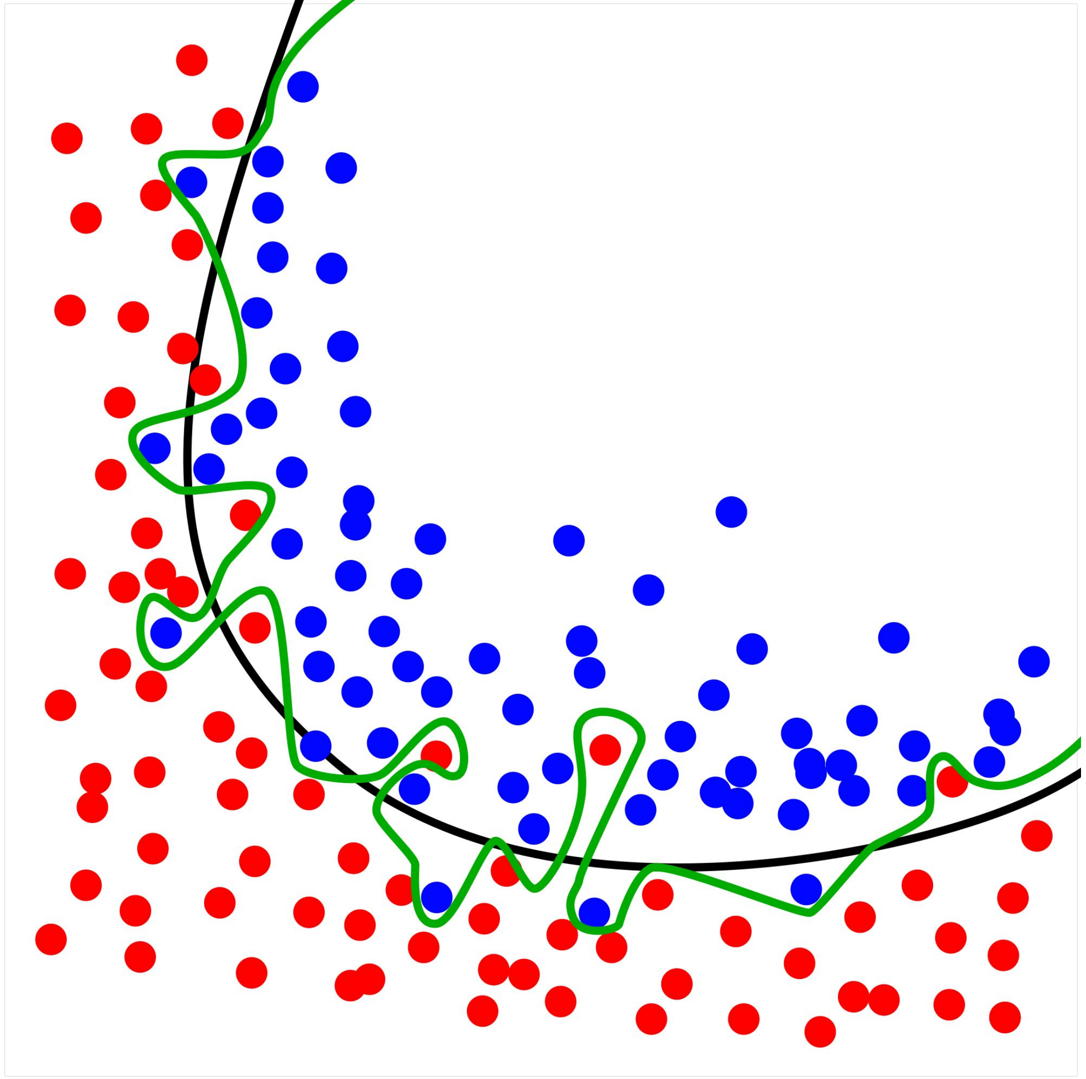
1. Overfitting.
2. Underfitting.

Both are considered ae error that result in an unsatisfactory performance and predictions.

Overfitting

&

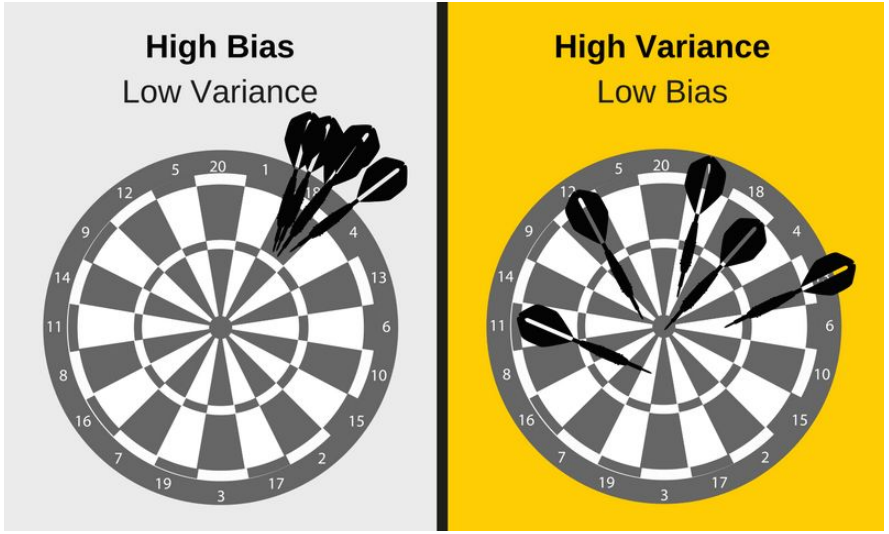
Underfitting



Both errors are the result of the tradeoffs between variance and bias.

Overfitting is when the model learns all the details and noise about your data to the extent that it performs 100% accurate in your training data, but when you test on a new data the model performs poorly.

Underfitting is when the model is not able to perform well in the dataset. The poor performance in both training and testing sets is an indicator of the underfitting error.



How to prevent this?

1. Cross validation.
2. Using Ensemble or bagging methods.
3. Add more data.
4. Reduce features (via feature selection or PCA).
5. Start with simple method.
6. Early stopping.
7. Regularization.

Evaluation of classification models.

Accuracy

Precision

Sensitivity/Recall

F1-Score

AUC (Area Under the ROC Curve)

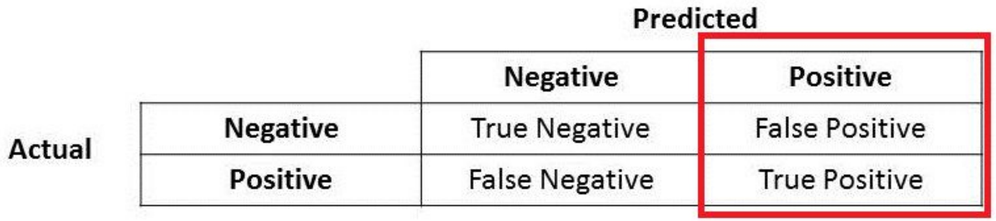
Precision

Spam Detection: you may lose few emails because you classify them as spam.

How precise your model is? Out of the predicted positive instances how many are actually positive.

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

$$= \frac{\textit{True Positive}}{\textit{Total Predicted Positive}}$$



		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

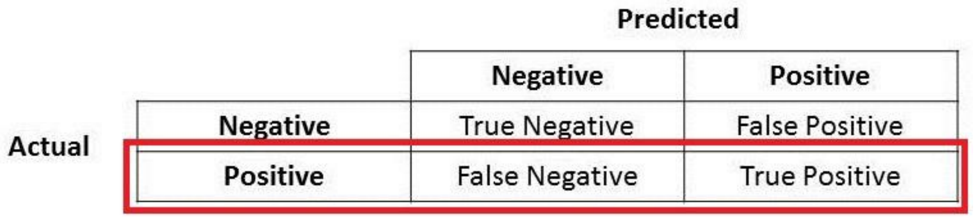
Recall
Sensitivity

Intrusion Detection: if the model misclassify an intrusion a harmful attach will cost the organization millions and sensitive data.

How many of the actual positives are classified as so.
How sensitive is the model in detecting the right class.

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}}$$

$$= \frac{\textit{True Positive}}{\textit{Total Actual Positive}}$$

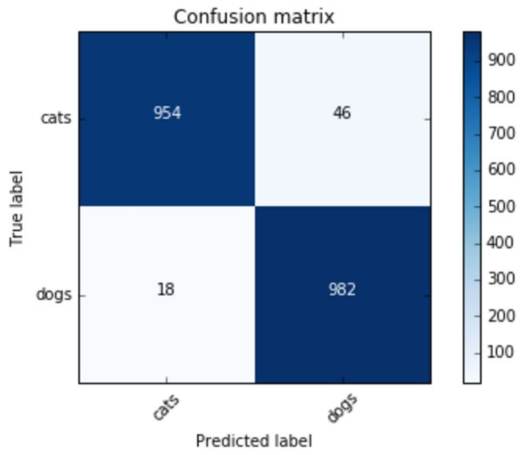


		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Tools in scikit learn

Confusion Matrix

```
[[954  46]
 [ 18 982]]
```



Classification Report

```
             precision    recall  f1-score   support

        0.0       0.86      0.90      0.88       543
        1.0       0.83      0.76      0.79       342

avg / total       0.85      0.85      0.85       885
```

Evaluation of Regression models.

Root Mean Squared Error.

Relative Squared Error.

Mean Absolute Error.

Regression Evaluation Metrics

RMSE: measure the error rate of a regression model. However, it can only be compared between models whose errors are measured in the same units.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}} \qquad RSE = \frac{\sum_{i=1}^{n}(p_i - a_i)^2}{\sum_{i=1}^{n}(\bar{a} - a_i)^2}$$

RSE: relative squared error (RSE) can be compared between models whose errors are measured in the different units.

MAE: has the same unit as the original data, and it can only be compared between models whose errors are measured in the same units.

Model Optimization Fine Tuning

Decision Trees.

Artificial Neural Networks.

Support Vector Machines.

Parameters

Entropy OR Gini Index

Number of branches allowed, levels.

Activation Functions

Number of layers

Regularization term

Error Functions

Kernel functions

Epsilon values

Grid Search Method

Read more

A search consists of:

- an estimator (regressor or classifier such as sklearn.svm.SVC());
- a parameter space;
- a method for searching or sampling candidates;
- a cross-validation scheme; and
- a score function.

```
param_grid = [
  {'C': [1, 10, 100, 1000], 'kernel': ['linear']},
  {'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001], 'kernel':
['rbf']},]
```