Machine Learning and Auto-Evaluation

In very simple terms, Machine Learning is about training or teaching computers to take decisions or actions without explicitly programming them. For example, whenever you read a tweet or movie review, you can figure out if the views expressed are positive or negative, but, can you teach a computer to determine the sentiment of that piece of text. This has many real life applications. Whenever Donald Trump makes a speech, Twitter goes crazy and there are both positive and negative sentiments all over the site. Trump's campaign management then determines what is the overall sentiment among the populace.

Another example, Baidu predicted that Germany would take the World Cup 2014 before even the match was played. http://www.marketing-interactive.com/germany-will-win-world-cup-2014-baidu-predicts/

Weather Problem

Lets take this small dataset of favourable weather conditions of playing some game. The problem here is to forecast whether one will be able to play the game in the given weather conditions.

Outlook	Temperature	Humidity	Windy	Play
IsSunny	Hot	High	False	No
Rainy	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes

Definitions

Feature/Attribute: Outlook, Temperature, Humidity and Windy w.r.t the above dataset are called features or attributes as they characterize or play a role in determine the favorable conditions.

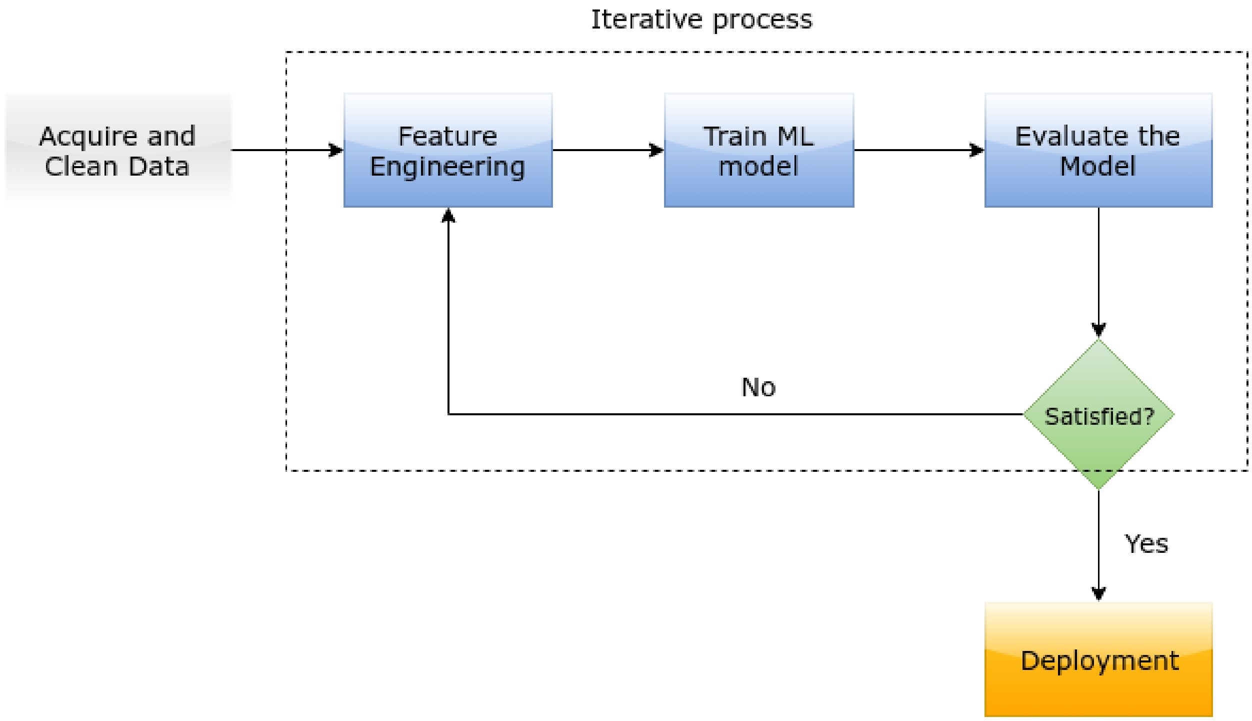
Outcome/Target: The problem is to predict what the outcome or target would be i.e whether you would be able to play the game or not.

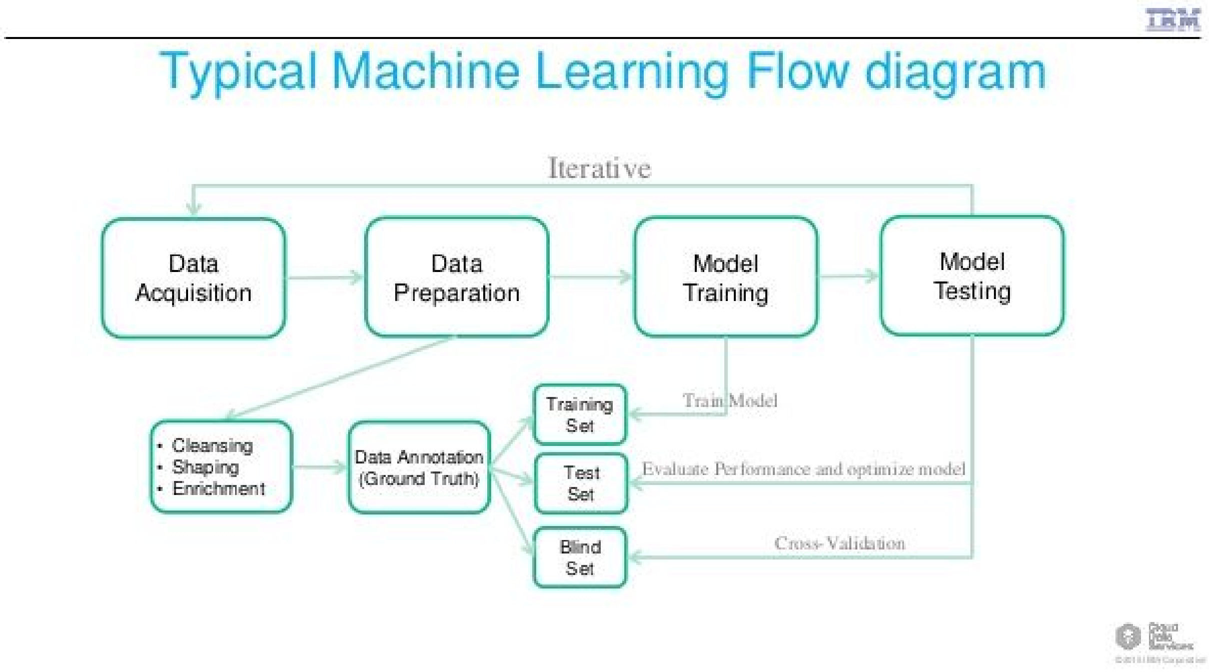
Vector: An ordered collection of features or attributes that correspond to a row in the dataset.
Eg. **(Sunny, Hot, High, False)** corresponding to the first row is a vector with the target as **No**.

ML Model: The algorithm or process or artifact that is generated as result of the learning process. There are lots of different types of ML models, e.g. Decision Trees, SVM, Naive Baiyes all capturing different aspects of the data.

Error Metric/ Evaluation Metric: Whenever we train an ML model, we need to determine how correct or how well the model performs i.e how accurate the predictions are. There are again different types of evaluation metrics according to the type of ML problem.

Typical Learning Process





Typical Machine Learning Flow diagram

Supporting ML Problems on HackerEarth

Our ML platform supports the machine learning flow as depicted in the above diagrams. We divide a given data set into training and test dataset. Training dataset is the data on which users have to train their models. Once the models are trained, users are expected to do the prediction on the test dataset and submit the prediction file. Test data that we give out to the users doesn’t have the target variable.

For e.g, for the following dataset of 10 rows, we can divide the dataset into train dataset with 60% of the rows and test dataset as the remaining 40% of the rows.

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Rainy	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Sunny	Mild	High	False	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	True	Yes

Train Dataset (train.csv)

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Rainy	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Overcast	Hot	Normal	False	Yes

Test Dataset (test.csv)

Id	Outlook	Temperature	Humidity	Windy
1	Sunny	Mild	Normal	True
2	Sunny	Mild	High	False
3	Overcast	Cool	Normal	True
4	Rainy	Mild	High	True

Observe that in the test dataset, we hold out the target variable.

Now, users train their models on train.csv, do the prediction on test.csv and submit a prediction file similar to the following file.

User Prediction File (user_prediction.csv)

Id	Play
1	Yes
2	Yes
3	No
4	No

On our server, we have the correct prediction file which we had separated out from the training set.

Correct Prediction File(correct_prediction.csv)

Id	Play
1	Yes
2	No
3	Yes
4	Yes

Evaluation Metric

Once a user submits the prediction file, we give a score or evaluate the model performance only on 50% of the test dataset according to an evaluation metric which we define for the given problem. For the weather problem, we can define one of the evaluation metric as:

Score = No of correct predictions / Total Rows

For the submitted file, there is only 1 correct prediction for id=1 for the first 50% of the test dataset(id=1, 2), so, the score according to the above defined formula for the first two rows would be:-

$$\textbf{Score}_{\textbf{online}} = \tfrac{1}{2} = 0.5$$

Once the contest is over, we re-evaluate the file and give a score on the complete dataset. Now the score would be:

$$\textbf{Score}_{\textbf{offline}} = \tfrac{1}{4} = 0.25$$

The point to observe here is that the score after the offline evaluation may go down or up. This is due to the overfitting i.e users may try to maximize their score during the online phase and their models may not be generic enough to perform well on data outside of what was used in the training set. The practice of doing the evaluation on 50% of the test dataset during the online phase is to discourage overfitting by the users.