Introduction to machine learning

Machine Learning is the study of algorithms that
• improve their performance P
• at some task T
• with experience E.
A well-defined learning task is expressed in terms of <P, T, E>.

Example machine learning tasks in real world:
Facial recognition = identifying persons by their image or video
Product Recommendation = While shopping on eCommerce websites like Amazon or Flipkart, the user is shown options like 'users who bought this product also bought'; 'based on your activity we recommend the following products' . This recommendation comes from machine learning

Chatbot = developed using natural language processing techniques , it can respond to queries.

Quality of output depends on the quality of the data on which the system is trained and how good the training data is representative of the real world data.

Types of Learning
1) Supervised learning

Supervised learning -
Given the set of input variables what is the best mathematical model to match the output - Mathematical model could be equation or set of rules or tree(s) .

If output variable is a real number, the Machine learning solution is called a regression model
If output variable is a label or category (like positive/negative, Yes/ No) , it is called as a classification model

Identifies the influence of each variable on the output - summarizes the large data into a ML model in such a way that prediction error should be minimal or zero(ideal case) . Getting 100% accuracy of prediction is not possible always.

Output value
Regression:
Given the inputs square feet, location of house, distance to bus stop , distance to metro train , number of rooms, house price of selected houses in an area find the house price given the inputs

Given the car info and its historical price, predict the price of a new car
Input variable:
Class = SUV, Luxury, Sedan
Manufacturer name = Maruti, Hyundai ,

Inside City Mileage =
Highway Mileage =
Free Service offered = Yes or No
Fuel Type = Petrol / Diesel

Input variables are called features, predictors, independent variables and the output variable is also called target or dependent variable or response variable.

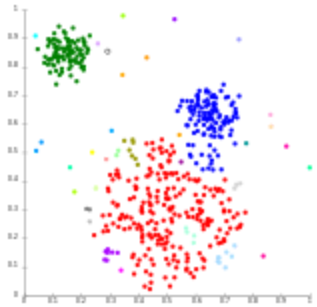
Output Variable: Predict Car Price

Credit scoring is a supervised learning - classification problem . Inputs are the details of the person like income, savings , age , industry and output is labeled like high-risk or low-risk . Credit scoring used by determine whether a person can be issued credit card or loan

For good introduction to Machine learning please refer this file:
This file contains very good introduction to machine learning
https://www.cmpe.boun.edu.tr/~ethem/i2ml3e/3e_v1-0/i2ml3e-chap1.pdf
There are other good pdf files here https://www.cmpe.boun.edu.tr/~ethem/i2ml3e/

Unsupervised learning

Given a set of points group them into different clusters.
Image source: https://en.wikipedia.org/wiki/Cluster_analysis
Different colours indicate different groups.



NOTE : There is no clear separation indicating which is the input variable and which is the output variable .

Uses of clustering:
Customer segmentation = group of customers based on the spending patterns as
- tech gadget savvy customer

- book loving customer
- fashion loving customer

Ecommerce websites group customers as above to send targeted ads (personalized or customized ads for a person).

Clustering is also used in finding anomalies or outliers - points which are completely different from the rest of the data. Anomaly detection finds a lot of uses in the financial industry like fraud transactions, unusual login activities.

Reinforcement Learning

Learning sequence of actions based on the current state based on rewards , positive and negative , and a set of learning policies . Learning is by trial and error method . Repeats many actions independently.
Example : Autonomous car driving

1. Car driving off the road - negative reward
2. Car hitting an object - negative reward
3. Car travelling on road continuously - positive reward
4. Learns Best action to take , depends on current state
5. NOTE: At each state, a safety criteria has to be honoured

Some of the algorithms used in the industry:

Generalized additive models (GAM)

The Regression Equation becomes:

$$f(x) = yi = \alpha + f_1(x_{i1}) + f_2(x_{i2}) + \ldots f_p(x_{ip}) + \epsilon_i$$

where the functions $f_1, f_2, f_3, \ldots f_p$ are different Nonlinear Functions on variables X_p.

In simpler words apply a transformation function like straight line, polynomial function or any other curve and do a weighted sum to get the output variable.

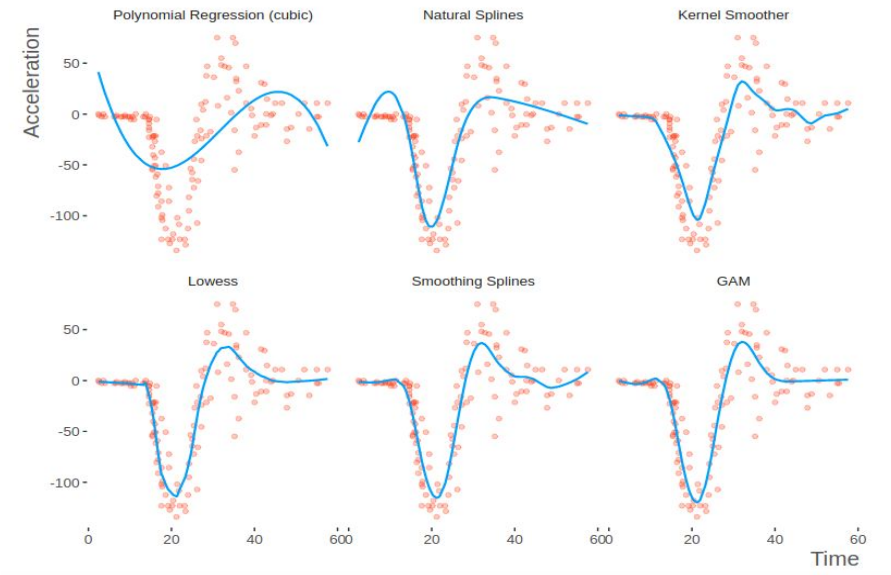


Image source: https://en.wikipedia.org/wiki/Generalized_additive_model

Decision Tree

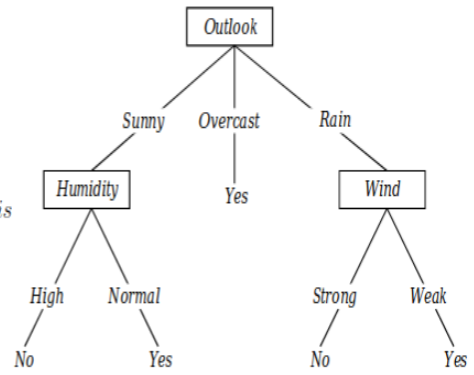
Given the data:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

predict the value of PlayTennis for

$$\langle Outlook = sunny,\ Temp = cool,\ Humidity = high,\ Wind = strong \rangle$$

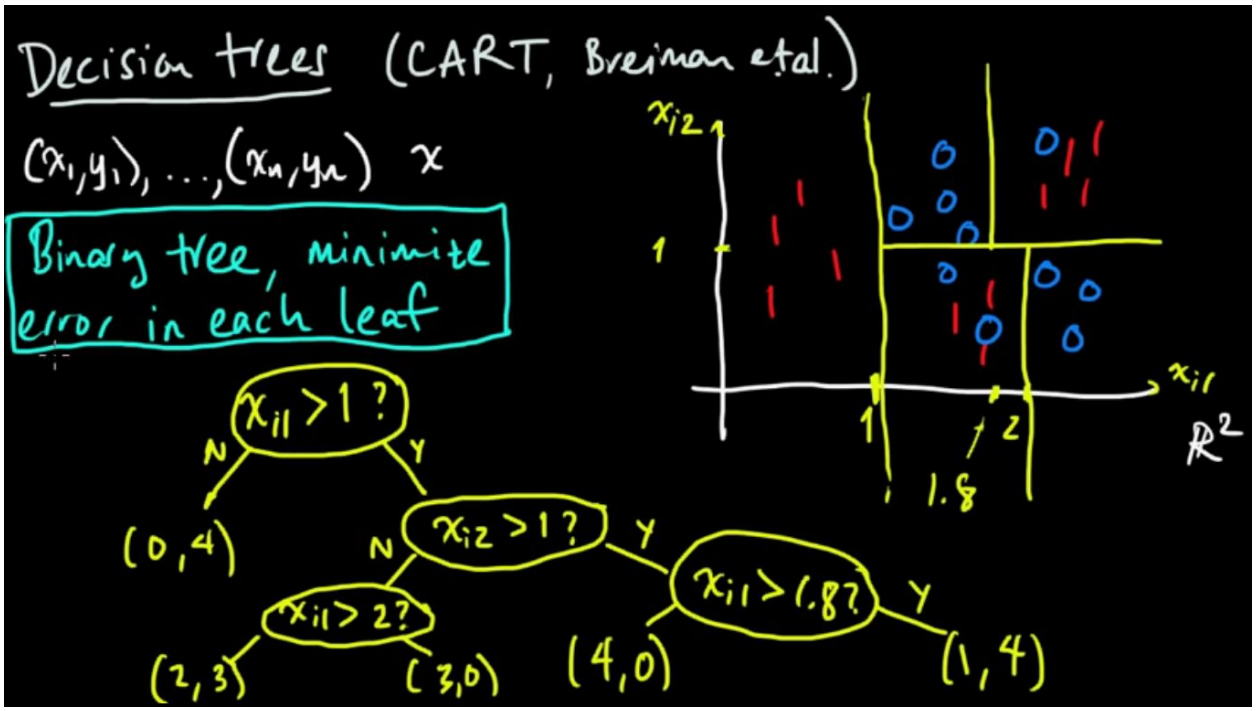
Example:
Decision Tree for $PlayTennis$



The above picture is from this link http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/mlbook/ch3.pdf Tom Mitchell's book on machine learning

Decision tree construction:

Cut the data at specific points (using < or > conditions) multiple times to build a tree



Picture is from the below video

https://www.youtube.com/watch?v=p17C9q2M00Q&list=PLScpSunNAuBY8E5O2s_61jTolHuYKp_c7

https://www.youtube.com/watch?v=zvUOpbgtW3c&list=PLScpSunNAuBY8E5O2s_61jTolHuYKp_c7&index=2

Boosting

Suppose that for a regression problem we built model M1 that predicted value is 9. Actual value is supposed to be 15 in the training data.
Residual is now = 15 - 9 = 6
For this residual , we built another model M2 to predict 6. Assume that the tree predicted 2 instead of 6.
Next residual = 6 -2 = 4
Again for this residual of 4, we built another model M3 to predict 4. Assume that the tree predicted 3 instead of 4.
Residual = 4 - 3 = 1

Assume that for this residual of 1 we built model M4 to predict 1. Assume that the tree predicted 1 . Residual 1 - 1 = 0

M1, M2, M3, M4 are applied one after another during prediction where each model tries to predict the residual from the previous tree. When M1, M2, M3, M4 are used one after another in sequence we will be able to get the prediction. This kind of model building is called boosting

https://www.linkedin.com/pulse/intuitive-ensemble-learning-guide-gradient-boosting-study-ahmed-gad/

Bagging:
To improve stability and accuracy we do not use one single decision tree.

Assuming we have data of 1000 rows.
We built 100 decision trees each with 300 rows , where these 300 rows are selected randomly from the 1000 rows with replacement . Note some rows may be repeated more than once since we are sampling the data points with replacement.

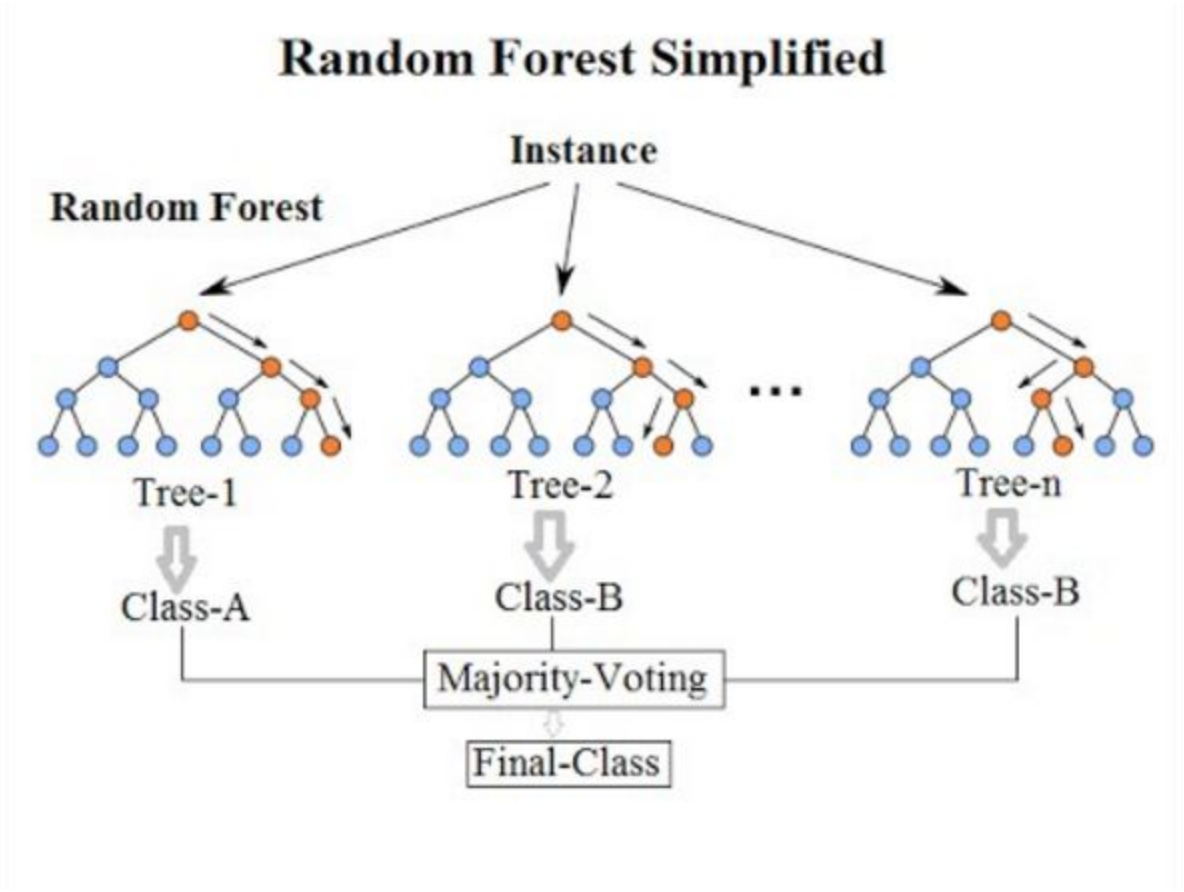
To predict, the prediction output of each decision tree is averaged (for regression problem) or maximum voting is used to find output class (for classification problem).

Maximum voting means if the majority of the trees give one class as the output in classification problem then that class is the predicted output.
Bagging or Boosting can be used with any machine learning models not only with decision trees. Bagging with decision trees is called random forest.

Source of image: https://en.wikipedia.org/wiki/Random_forest

In case of a regression problem (predict real number), instead of class-A, class-B .. as output we will be getting var1, var2 , var3… varn . As output where each value is a real number. Final

predicted output of random forest is average of var1, var2, var3 …varn



Support Vector Machine(SVM)

SVM when used for classification divides the positive class and negative class using a hyperplane. Note the input data should be linearly separable . That is, some hyperplane exists to divide the input data into positive and negative classes. If the data is not linearly separable, transformation functions are applied on the data to make it linearly separable.

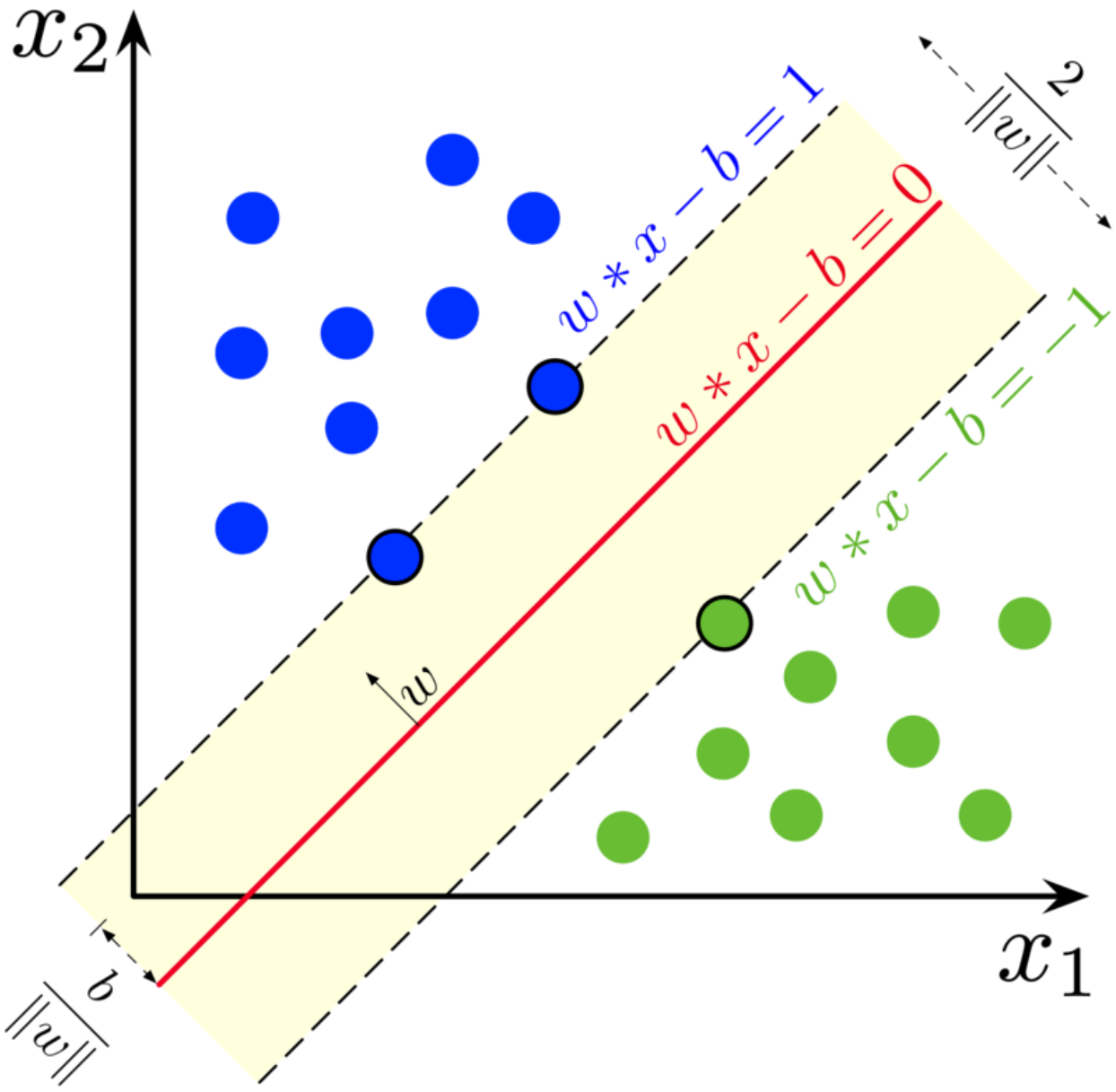


Image source: https://en.wikipedia.org/wiki/Support_vector_machine

K - Nearest neighbours

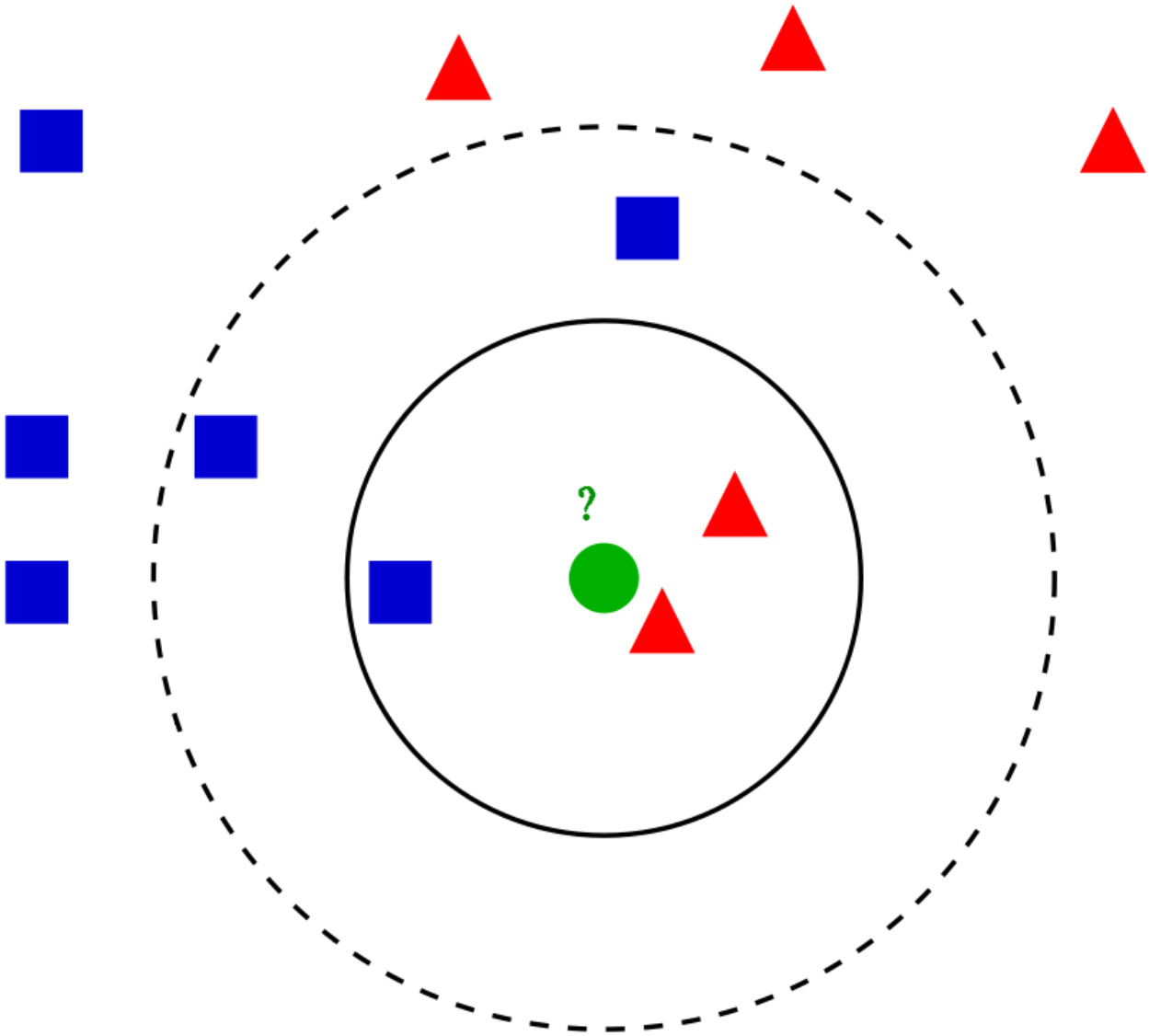


Image source: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

In the above diagram which class does the green point belong to . Blue or Red class ? We find 3-Nearest neighbours to it. One neighbour is Blue and two are Red. Majority of them are Red . So we say the green point should belong to the Red (majority) class. This is a K-NN model with K=3. The K-NN algorithm can be used for regression as well. Find K nearest points and take the average of the output variable for those points. This will be the predicted output for one test input. Usually k to 3 to 5 is common.

What is the right value of K. Use the value of K that gives good test accuracy. Usually machine learning data is split into 70:30 and 80:20. We do not build models on the full data. We build ML models on 70 or 80 percent of data and evaluate the model on the remaining data called test data. Test data should not be

part of model building. If it is part of model building it is called a data leakage problem. Data leakage should be avoided to get a better model. How to choose the right value of K ? Vary K from 1 to some random value of K , say 30 and then compute prediction errors for multiple value of K. Choose K with minimal error. For detailed steps, please check the end of this blog
https://www.ritchieng.com/machine-learning-k-nearest-neighbors-knn/

Class imbalance

In the credit scoring example if 90% of data points are low-risk and 10% are high-risk then there is a class imbalance. One class is present in more examples than the others. The data point of the majority class will highly influence the final model. This will lead to inaccurate predictions.

Multiple techniques like SMOTE , Adasyn , Random Oversampling , Random Undersampling should be used. Essence of these techniques will be to make low-risk and high-risk data points approximately in similar percentage (50%) either by throwing few points of majority class or selecting more points from the minority class with repetition.

Good read on imbalance data - click the link Learning from imbalanced data.

GridSearchCV example
We use GridSearchCV to adjust a model to give less prediction errors. GridSearch tries various combinations of parameters to construct the model and picks the one that gives better accuracy measure.

A sample usage of model building for a classification building using grid search

```
from sklearn.model_selection import GridSearchCV

#Create values to search over
cv_params = {'max_depth': [3,4,5], 'min_child_weight': [3,5], 'learning_rate':[0.1,0,2,0.3]}
ind_params = {'n_estimators': 50, 'seed':1, 'colsample_bytree': 1,
        'objective': 'binary:logistic'}
opt_XGBclassifier = GridSearchCV(xgb.XGBClassifier(**ind_params),
                    cv_params,
                     scoring = 'accuracy', cv = 5, n_jobs = -1, verbose=3)

# Grid Search tries to fit best model using the various parameters given in the option
opt_XGBclassifier.fit(X_train, y_train)  # fit the model
print(opt_XGBclassifier.grid_scores_)
```

One-hot encoding

One-hot encoding is where you represent each possible value for a category as a separate feature. The most straight-forward way to do this is with pandas (e.g. with the City feature again):

Assuming that the input data contains a column called City which contains 'Delhi','Mumbai' and 'Chennai' as possible values.

import pandas as pd

df = pd.read_csv('data.csv') # comment: load data from the file data.csv to data frame named df

df = pd.concat([df, pd.get_dummies(df["City"], prefix="City")

After executing the above lines, one can see additional columns by the name City_Delhi, City_Mumbai, City_Chennai. Only one of this column will contain a value of 1 indicating the respective city name.

If city is Chennai for a row , then for that row, City_Chennai = 1 ,City_Delhi =0 and City_Mumbai=0

Actual value = 100 predicted value = 102 absolute_error = 102 - 100 = 2 squared_error = 4
Actual value = 102 predicted value = 105 absolute_error = 105 - 102 = 3 squared_error = 9
Mean absolute error MAE =Average of absolute error = (2 + 3) / 2 = 2.5
Mean squared error MSE = Average of squared_error = (4 + 9) / 2 = 6.5
Root mean squared error RMSE = sqrt(MSE) = sqrt(MSE)
Mean absolute percentage error (MAPE) = absolute value (predicted value - actual value) / actual value * 100

Do not use MAPE if the output values are small .
For actual value is 0.3 and predicted value is 0.15 , MAPE = (0.3 - 0.15) / 0.3 = 50% Look at the percentage the error looks big . When you look at magnitude, error is tiny. Use one or more of RMSE , MAPE, MAE according to the regression problem.

Software coding

Installation of Anaconda in linux platform
Follow the steps 1 to 12 mentioned in https://docs.anaconda.com/anaconda/install/linux/
Starting from the section "For x86 systems.

1. In your browser, download the Anaconda installer for Linux "

(sample1) krishna@dev31:~/flaskfiles$ **conda install flask numpy pandas scikit-learn jupyter**

Type ‘y’ to proceed with installation

$ pip install shap
Type ‘y’ to proceed with installation

Type the following commands one after another
$conda create -n flask_env
$conda activate flask_env
$mkdir ~/flask_files
$cd ~/flask_files
$ git clone https://github.com/trkrishnan/techTalk.git

Working of a simple webservice

Start the sample webservice server by typing

$ cd techTalk

$ python sample_app.py
The last line of the output of the command will look like the one below:

 * Running on http://127.0.0.1:8888/ (Press CTRL+C to quit)

In another terminal, run this command

$ curl -X POST http://localhost:8888/sampleAPI
You should see a response: **Server received a request. Thank you**

The curl command sends a web service request to the sampleAPI method on the server. In the server the sampleAPI method is redirected to the method sampleResponse by means of the line
@app.route('/sampleAPI', methods = ['POST'])

Using webservice to build and predict model

We will run a server to predict house prices . First we make a webservice to build a model and save it in a file.

$ curl -X POST -H "Content-Type: application/json" http://localhost:5001/buildHousePriceModel

Please note depending on machine configuration building of model may take 30 seconds to 150 seconds for completion. The output looks like the one below:

```
{
  "model building": "success"
}
```

As the model building is successful , we now issue the webservice call to make predictions.

We invoke the predict API by issuing a web service call like this

$ curl -X POST -H "Content-Type: application/json" -d '{"grade":9.0,"lat":37.45,"long":12.09,"sqft_living":1470.08,"waterfront":0.0,"yr_built":2008.0}' http://localhost:5001/predict_price

```
{
  "predict cost": 546634.691771444
}
```

On receiving the build model API the server calls the build model function in the predict.py file. The build model function reads the file from the file kc_house_data.csv. In real world applications data can be read from a database or data can be passed in the build model routine itself (just like how data is passed to the predictAPI). The built ML model can be saved in a file or in database or in a AWS S3 URL or in any cloud stage

Pycaret and H2o are briefly explained in their respective notebooks. H2o requires java to be installed and available in the system path.

To open the notebook , type

$ **pip install pycaret h2o shap**

$ **jupyter notebook pycaret_demo.ipynb**

$jupyter notebook h2o_demo.ipynb

To run a notebook , choose 'Run All' from the Cell menu. Pycaret prompts the user to type enter to proceed with the next steps while executing the setup command.

Owing to lack of time important topics like Feature Engineering ,Clustering, Feature selection , Model evaluation and production deployment are not covered . One needs to learn them as well to use them in industrial ML projects. Machine learning is a very vast field of study. Continuous learning and working on multiple projects helps in achieving proficiency in the subject.

Useful references

1. Free online book ISLR http://faculty.marshall.usc.edu/gareth-james/ISL/
   videos https://www.dataschool.io/15-hours-of-expert-machine-learning-videos/
2. Interpreting SHAP https://blog.datascienceheroes.com/how-to-interpret-shap-values-in-r/

3. Book (not free) Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron
4. Excellent self paced course for learning introduction to machine learning https://mlcourse.ai/
5. ML Videos of Prof Andreas Mueller (Follow any one of the below video tutorials)

https://www.youtube.com/watch?v=9rBc3rTsJsY&list=PL_pVmAaAnxIQGzQS2oI3OWEPT-dpmwTfA&index=5

6. ML Videos of Prof Saptarshi Goswami https://www.youtube.com/watch?v=rwx_1_IOj3g&list=PLTS7rWcD0Do3O1iRmH6-LydircHibNrOC
7. ML Videos of Prof Alexander ihler

https://www.youtube.com/watch?v=qPhMX0vb6D8&list=PLaXDtXvwY-oDvedS3f4HW0b4KxqpJ_imw

8. ML Videos of Prof Arti Ramesh (Very good theory coverage)

https://www.youtube.com/playlist?list=PLUZjIBGiCHFfRJwflq6NqU3CuiPhAhSfi

9. For those who love R Hands on machine learning using R online book https://bradleyboehmke.github.io/HOML/
10. Krish Naik's channel on machine learning using Python and R

https://www.youtube.com/watch?v=EqRsD3gqeCo&list=PLZoTAELRMXVOnN_g96ayzXX5i7RRO0QhL

11. Coursera course - choose Free Audit option to do the course for free https://www.coursera.org/learn/machine-learning-with-python
12. This is the page I used to learn about pycaret

https://www.analyticsvidhya.com/blog/2020/05/pycaret-machine-learning-model-seconds/

Please make use of the github link and data provided in the blog

13. 24 project ideas in data science

https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/

14. https://archive.ics.uci.edu/ml/index.php UCI machine learning repository for datasets
15. Kaggle.com provides datasets
16. For website interacting with flask server to do ML predictions, please check this article and the corresponding github link https://towardsdatascience.com/how-to-easily-deploy-machine-learning-models-using-flask-b95af8f https://github.com/abhinavsagar/Machine-Learning-Deployment-Tutorials
17. Deployment using flask https://www.youtube.com/watch?v=UbCWoMf80PY

Downloading the source files:
To get all files from the github repository
git clone https://github.com/trkrishnan/techTalk.git

For any query on this doc please connect at
trkrishnan

AT
GMAIL
(_____ . _____)
COM

(Email is intentionally split into multiple lines to trick email harvesting programs.
If you put all bottom 5 lines together into one email address you will get my email id)