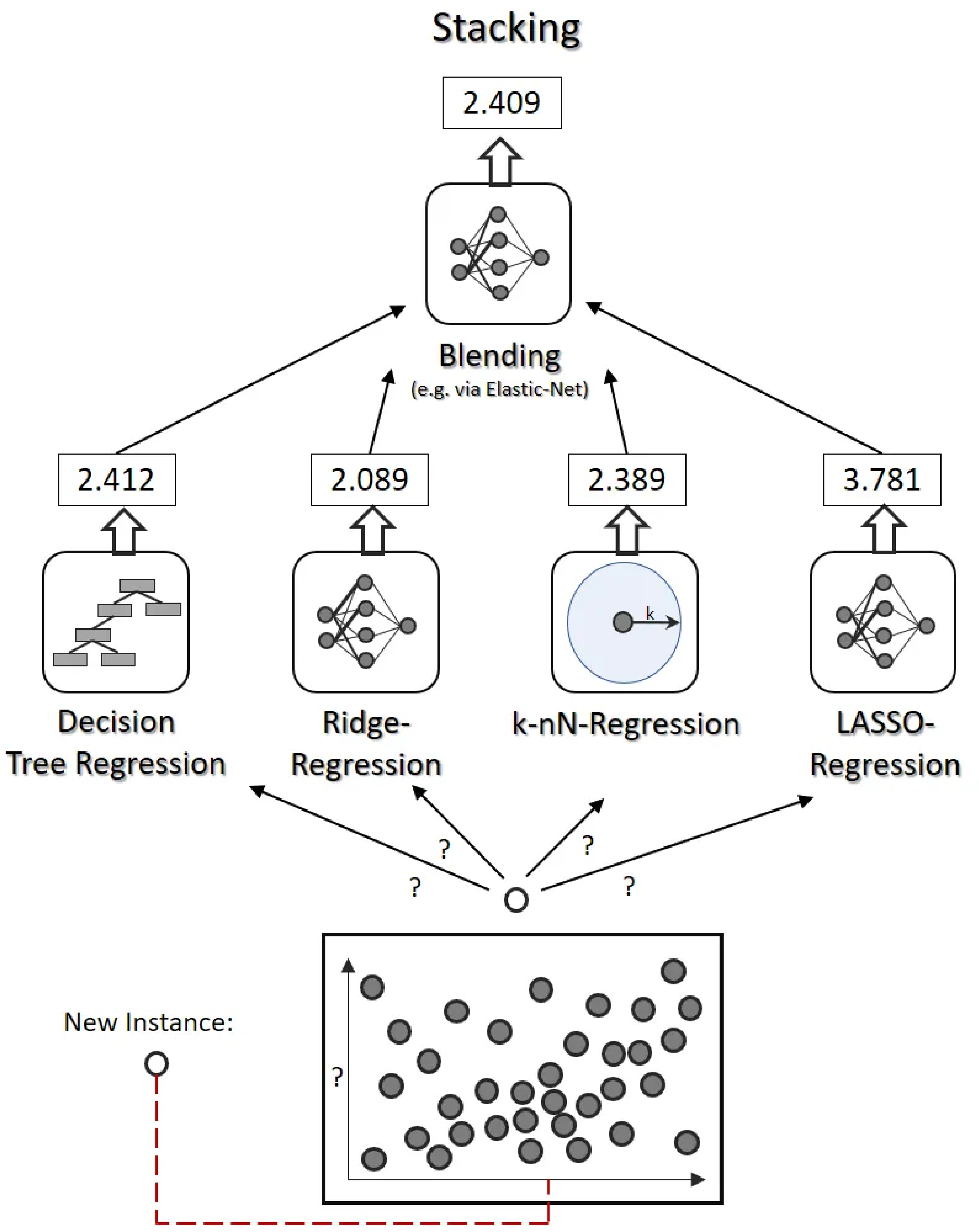


Source : https://data-science-blog.com/

Hey Folks...!!! Welcome Again this article series of Ensemble learning...!!! Today, I am talking about amazing concept called Stacking. So let us assume the scenario where you want to buy some product and you have taken so many review from people about that product and feedback etc. and Then after you bought that product based on all feedback you have received, This process is similar like Stacking.

> ***Stacking** is a ensemble learning method that combine multiple machine learning algorithms via meta learning, In which base level algorithms are trained based on a complete training data-set, them meta model is trained on the final outcomes of the all base level model as feature. We have deal with bagging and boosting method for handling bias and variance. Now we can learn stacking which is improve the your model prediction accuracy.*

Ensemble learning Series — (Happy Learning…!!!)

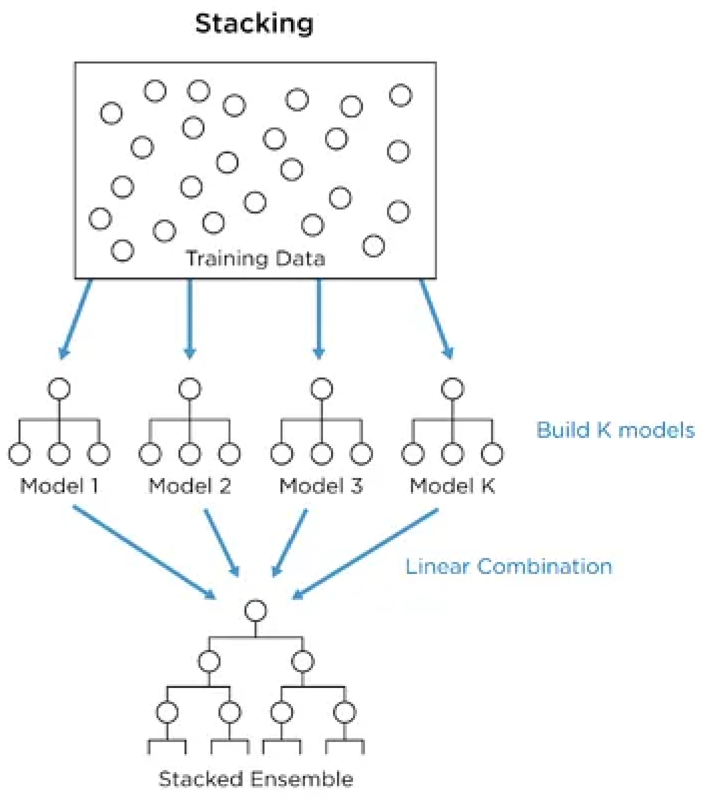
1. **Ensemble Learning Relation With Bias and variance**
2. **Ensemble Learning- The heart of Machine learning**
3. **Bagging — Ensemble meta Algorithm for *Reducing variance***
4. **Boosting- Ensemble meta Algorithm for *Reducing bias***
5. **Stacking -Ensemble meta Algorithms for *improve predictions***
6. **Ensemble learning impact on Deep learning**

Stacking — Method that Improve your prediction Result

In the previous Article, we got an idea of how multiple models of the same kind can help us improve the accuracy of our classification. What if we combine models of two different kinds? Can it help us? ***The answer is yes !!!*** In some cases, it is very helpful to use different kinds of prediction models to get higher prediction accuracy. Well, the next question is: how?

- **In Boosting,** We have trained single decision tree which is known as weak learner or **In Bagging,** It can help us only to make a partial prediction to reduce the bias error from our model for that we need to increase the number of classifier in our ensemble frameworks. In same way, if the data-set is complex, a single solution might be give you the proper prediction rate, for this kind of scenario we need to combine multiple and different classifier, where the output of one classifier become the input of another.

- Here Interesting part in the **Stacking process** come into the picture is that the new model is trained to combine prediction from previously trained models on the same data-set.

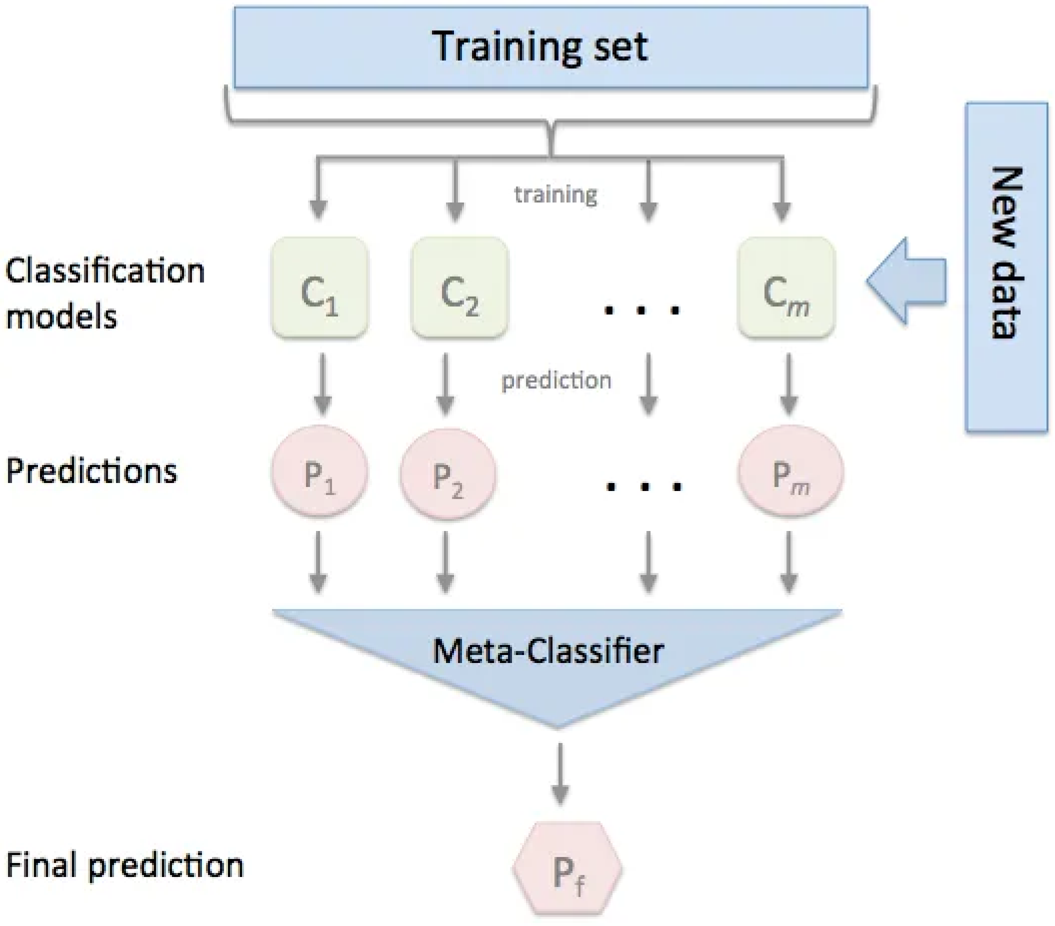


Source : Image.google.com

What is Stacking?

- **Stacking** is an extension of the voting classifier or voting regressor by a higher level (**blending level**), which learns the best aggregation of the individual results. At the top of stacking is (at least) another classifier or regressor.
- **Stacking** is particularly useful when the results of the individual algorithms can be very different, which is almost always the case with regression — since continuous values instead of a few classes. Stacking algorithms can even span several layers, which makes their training much more difficult.

How it works?



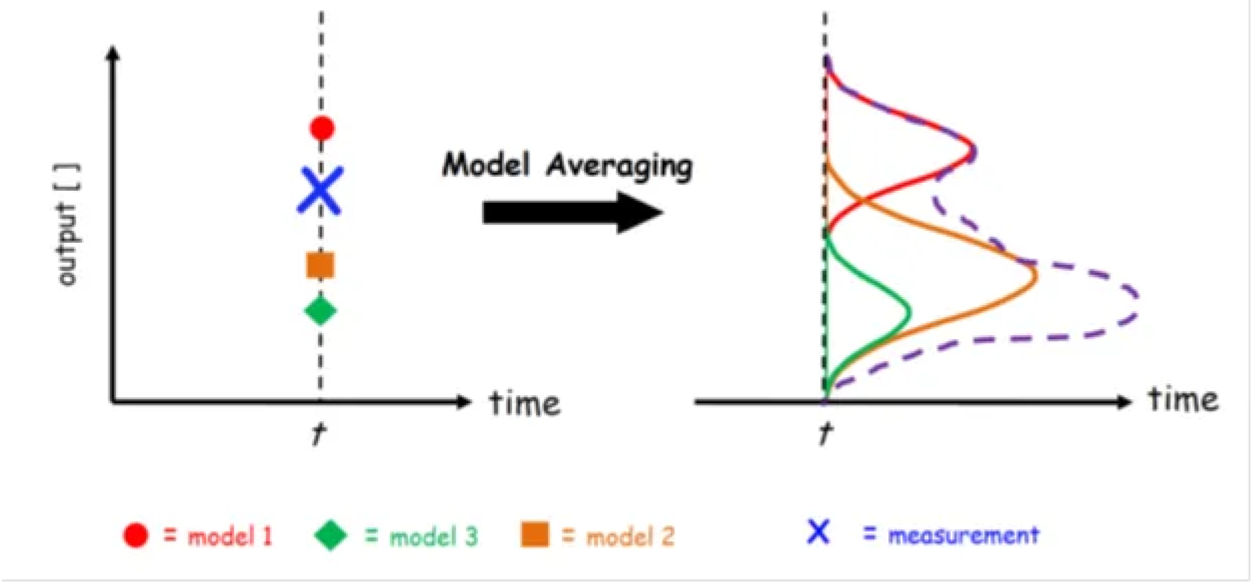
Source: http://rasbt.github.io/

Let us imagine we have trained multiple classifier and got the result. But we don't get the proper prediction result and in this situation stacking help us *by doing averaging that result* which we got from training the classifier or *by doing weighted sum as well.*

- In the above example we can see that we have training dataset which is trained on different classifier **C1, C2…. Cm** and which give the result **P1, P2… Pm** and We can take that all prediction and with help of **Stacking** and we got the final result **Pf.**

Stacking Methods:

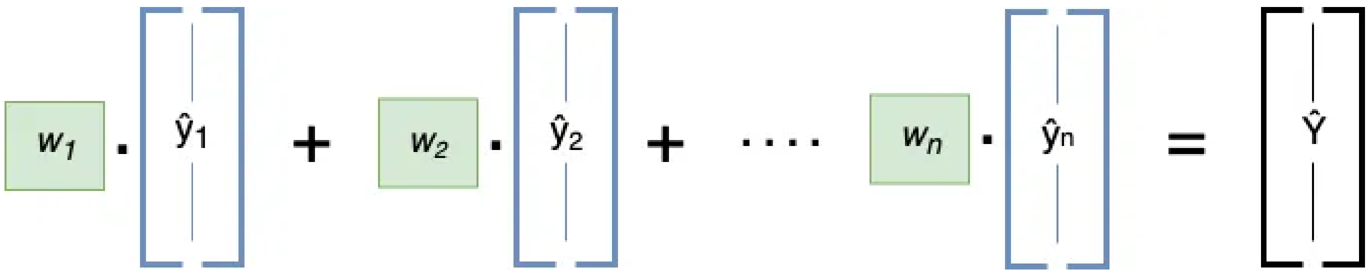
1. **Averaging** : Take P1,P2…Pm and do the averaging of that.



Average Stacking

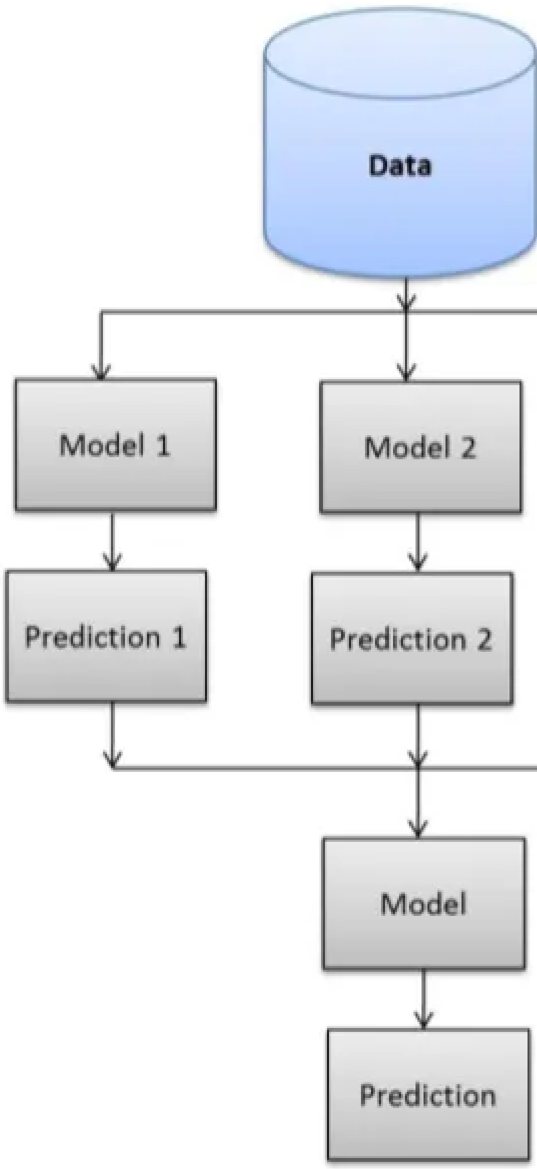
2. Weighted Averaging : Take weights for particular features as per their importance and we get the final result.

Weighted Averaging = P1 * w1 + P2 * w2 + ….. + Pm * wm


$w_1 \cdot \hat{y}_1 + w_2 \cdot \hat{y}_2 + \cdots + w_n \cdot \hat{y}_n = \hat{Y}$

Weighted Averaging Stacking

Important thing is that the sub-models (initial classifier) should deliver uncorrelated results; That said, their results should not match. A sub-model prediction is therefore an uncorrelated characteristic for the final classifier:



Stacking Generalization

In above Figure, we can see that different sample is not taken for training data to train classifiers.Instead of we are taking whole data-set for training for every individual classifier. In this process, each classifier is working independently, which permits the classifiers with different hypotheses and algorithms. For Instance, We can train our model on linear regression classifier, Decision Tree and random forest for training and then we can combine their prediction using a support vector machine.

Stacking Implementation:

Open in Colab

Import IRIS dataset from sklearn

In [0]:
```python
from sklearn import datasets
```

Impoert Random forest Logistic regression, naive bayes and knn classifier classes for creating stacking

In [0]:
```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
```

Import numpy for array based operations

Stacking.ipynb hosted with ❤ by **GitHub** **view raw**

We can see an improvement in prediction accuracy when we stack all the classifiers. This example shows how we can use predictions from other classifiers to train a new classifier on it to get a high performance prediction framework.

Thanks for reading…!!!

References:

1. https://medium.com/@rrfd/boosting-bagging-and-stacking-ensemble-methods-with-sklearn-and-mlens-a455c0c982de
2. https://en.wikipedia.org/wiki/Ensemble_learning

Machine Learning · Ensemble Learning · Stacking · Bagging · Boosting