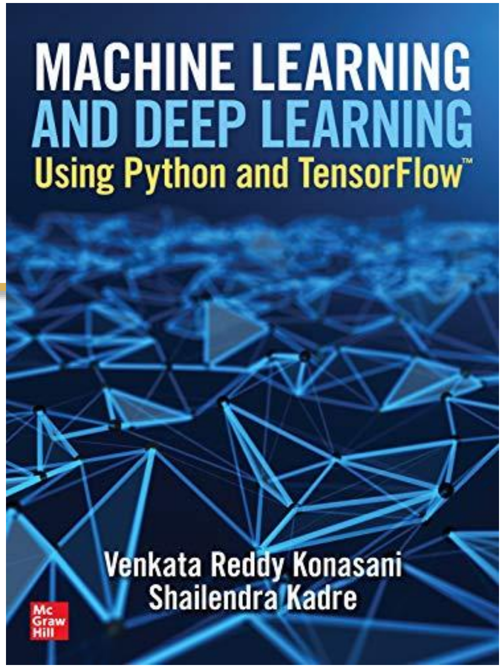
Ensemble Models and Random Forests

Venkat Reddy

Chapter 7 in the book



Contents

Contents

- Introduction
- Ensemble Learning
- How ensemble learning works
- Bagging
- Building models using Bagging
- Random Forest algorithm
- Random Forest model building

statinfer

The Wisdom of Crowds

The wisdom of crowds

- "One should not expend energy trying to identify an expert within a group but instead rely on the group's collective wisdom, however make sure that opinions must be independent and some knowledge of the truth must reside with some group members" – Surowiecki

  - So instead of trying to build one great model, its better to build some independent moderate models and take their average as final prediction

statinfer

The wisdom of crowds

Problem Statement: What is the estimated monthly expense of a family in our city.

An Eminent Professor built a model

Vs.

100 Assistant Professors built 100 models

M1

One Single Prediction

$6500

M1

M2

M3

……

M100

Average of all 100 predictions

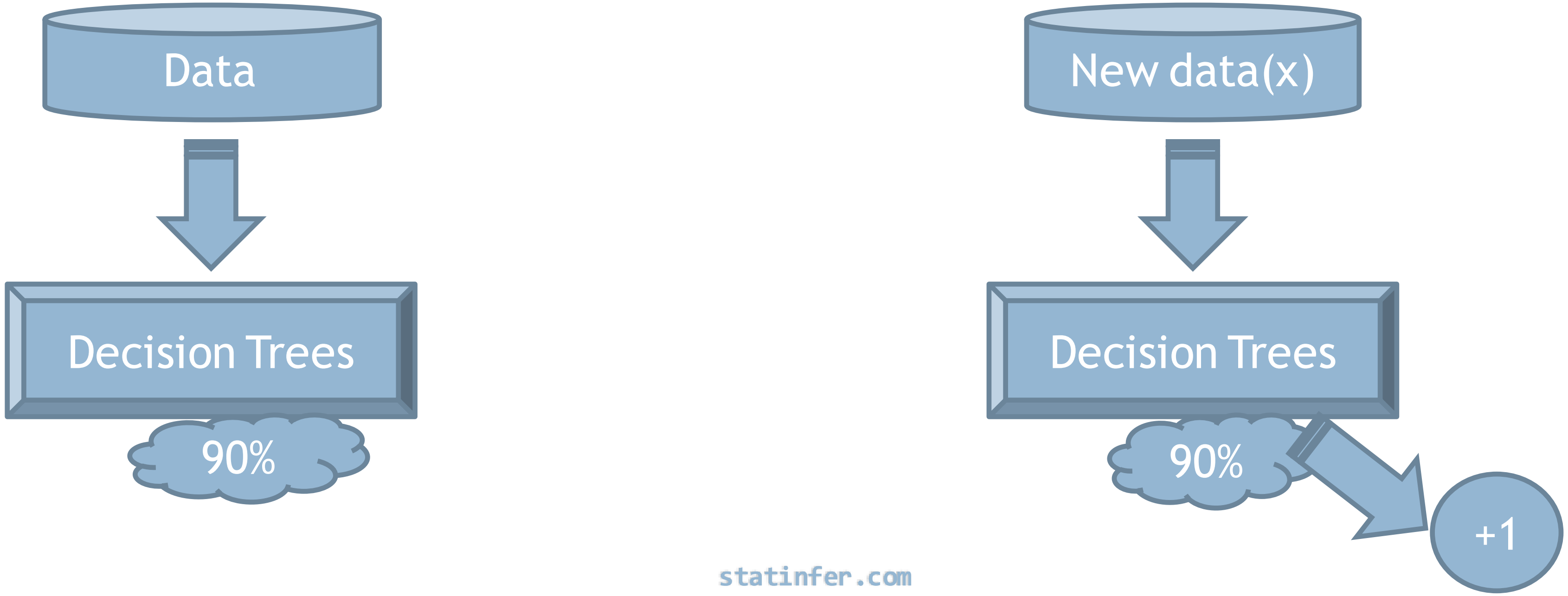
$7200

statinfer

What is Ensemble Learning

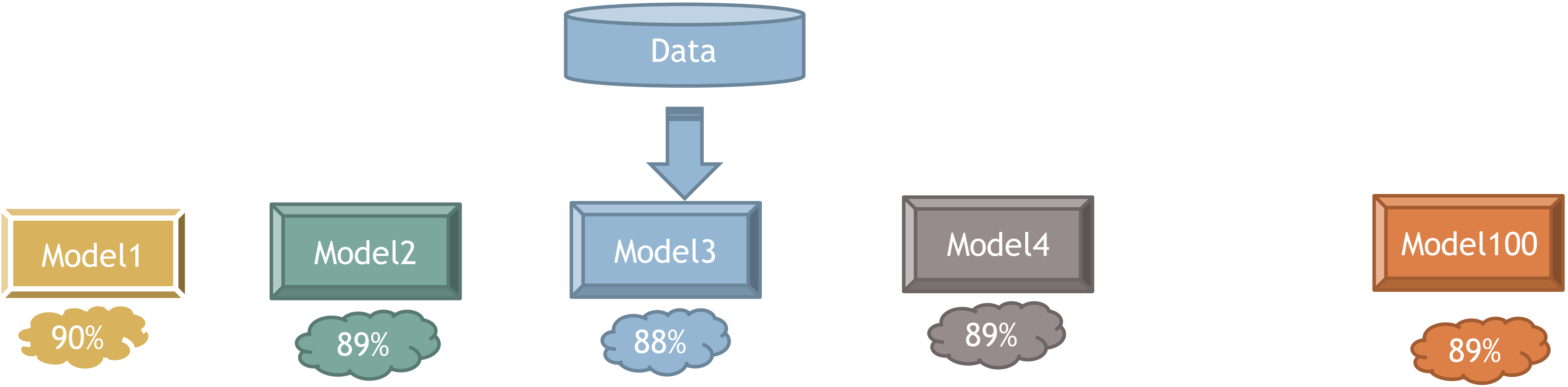
What is Ensemble Learning

- Imagine a classifier problem, there are two classes +1 & -1 in the target
- Imagine that we built a best possible decision tree, it has 91% accuracy
- Let x be the new data point and our decision tree predicts it to be +1. Is there a way we can do better than 91% by using the same data
- Lets build 3 more models on the same data. And see we can improve the performance



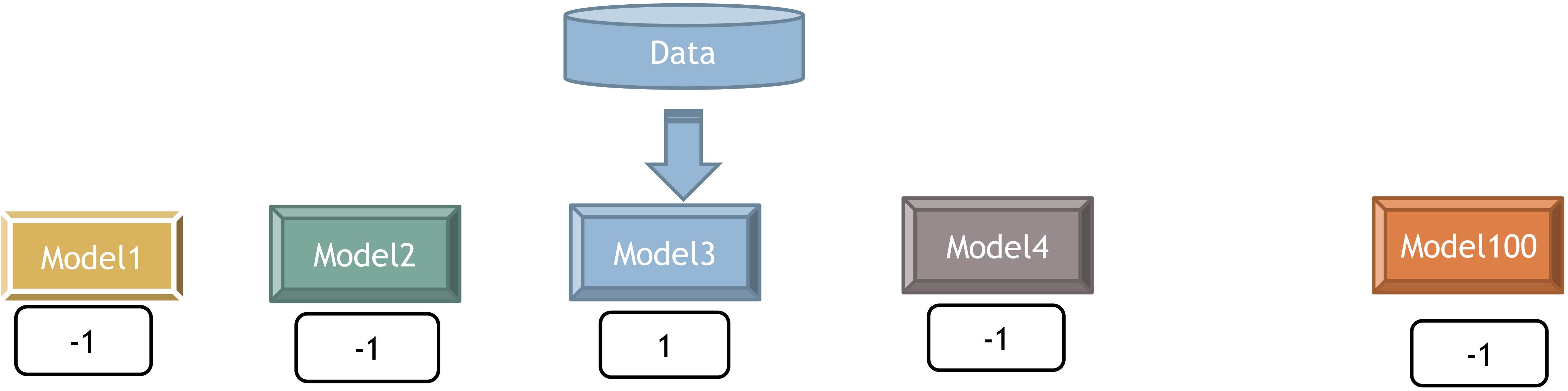
What is Ensemble Learning

- We have four models on the same dataset, Each of them have different accuracy. But unfortunately there seem to be no real improvement in the accuracy.



What is Ensemble Learning

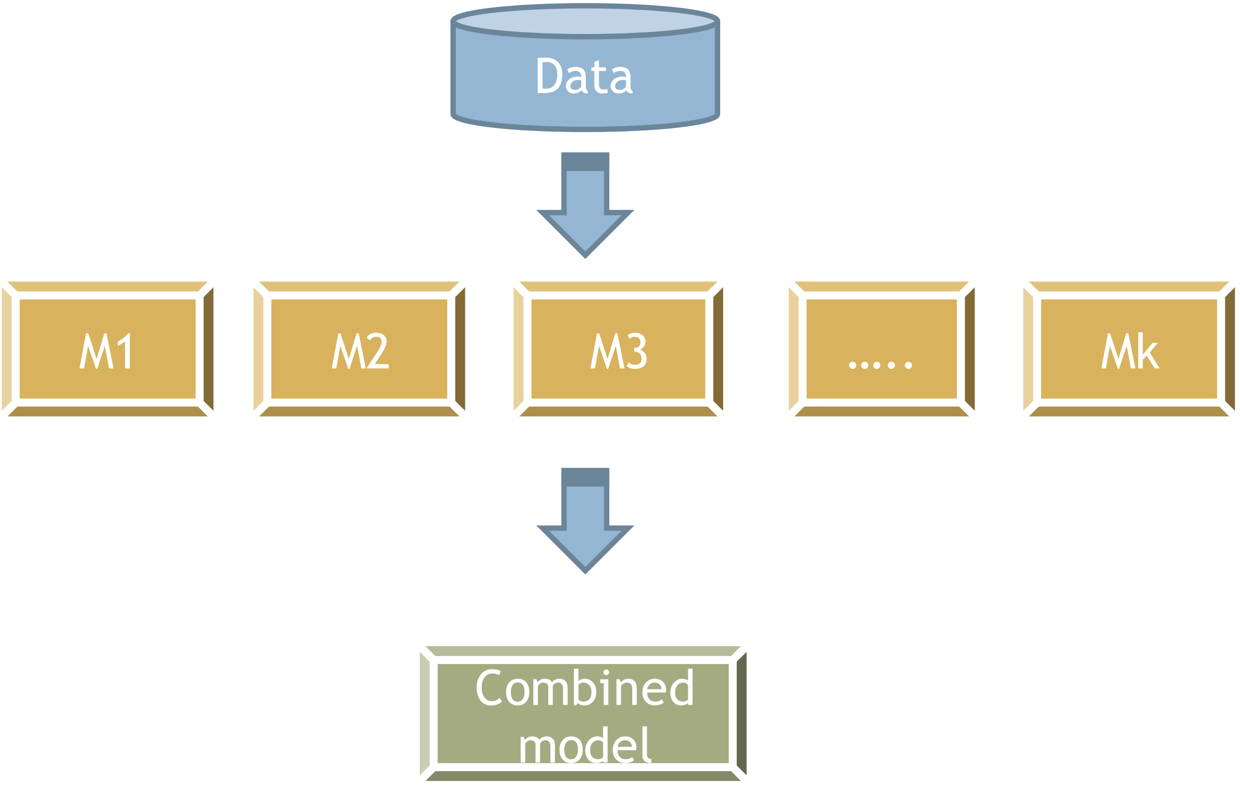
- What about prediction of the data point x?
- The combined voting model seem to be having less error than each of the individual models.
- This is the actual philosophy of ensemble learning



Ensemble Models

Ensemble Models

- Obtaining a better predictions using multiple models on the same dataset
- Not every time it is possible to find single best fit model for our data, ensemble model combines multiple models to come up with one consolidated model
- Ensemble models work on the principle that multiple moderately accurate models can give us a highly accurate model
- Understandably, the Building and Evaluating the ensemble models is computationally expensive
- Build one really good model is the usual statistical approach. Build many models and average the results is the philosophy of Ensemble learning



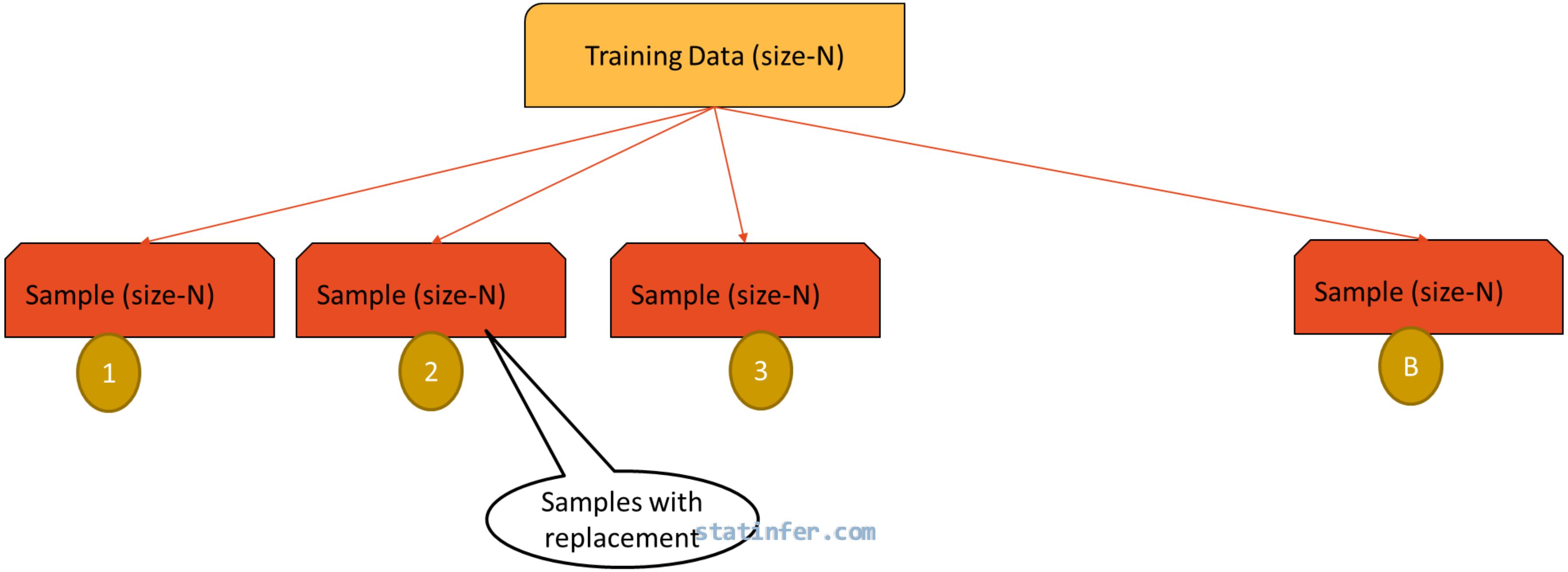
Bagging

Bagging

- Take multiple boot strap samples from the population and build classifiers on each of the samples. For prediction take mean or mode of all the individual model predictions.
- Bagging has two major parts 1) Boot strap sampling 2) Aggregation of learners
- **Bagging** = Bootstrap **Aggregating**
- In Bagging we combine many unstable models to produce a stable model. Hence the predictors will be very reliable(less variance in the final model).

Boot strapping

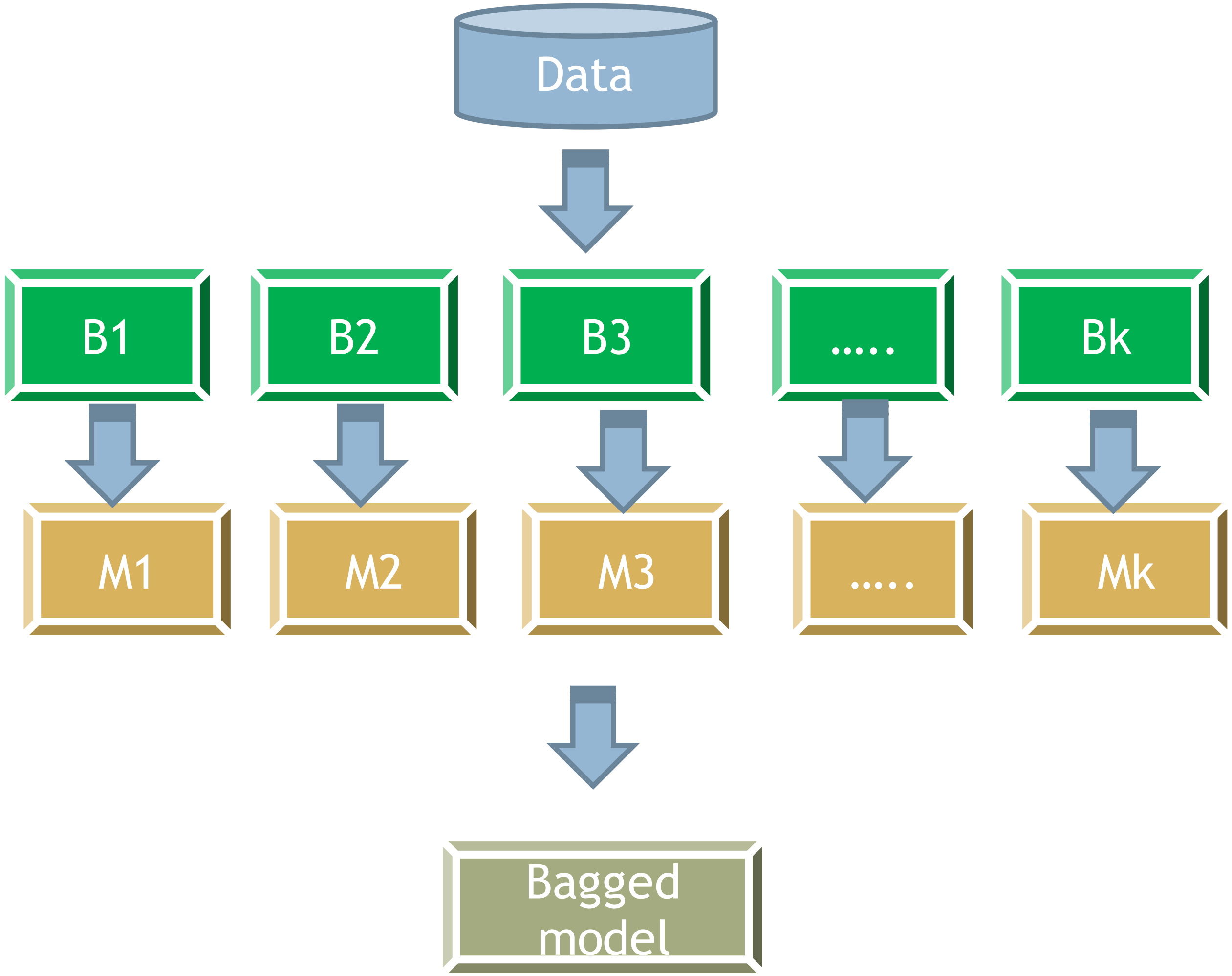
- We have a training data is of size N
- Draw random sample with replacement of size N – This gives a new dataset, it might have repeated observations, some observations might not have even appeared once.
- We are selecting records one-at-a-time, returning each selected record back in the population, giving it a chance to be selected again
- Create B such new datasets. These are called boot strap datasets



The Bagging Algorithm

The Bagging Algorithm

- The training dataset D
- Draw k boot strap sample sets from dataset D
- For each boot strap sample i
  - Build a classifier model M_i
  - We will have total of k classifiers M_1 , M_2 ,….. M_k
  - Vote over for the final classifier output and take the average for regression output



Why Bagging works

- We are selecting records one-at-a-time, returning each selected record back in the population, giving it a chance to be selected again
- Note that the variance in the consolidated prediction is reduced, if we have independent samples. That way we can reduce the unavoidable errors made by the single model.
- In a given boot strap sample, some observations have chance to select multiple times and some observations might not have selected at all.
- There a proven theory that boot strap samples have only 63% of overall population and rest 37% is not present.
- So the data used in each of these models is not exactly same, This makes our learning models independent. This helps our predictors have the uncorrelated errors.
- Finally the errors from the individual models cancel out and give us a better ensemble model with higher accuracy
- Bagging is really useful when there is lot of variance in our data

Random Forest

Random Forest

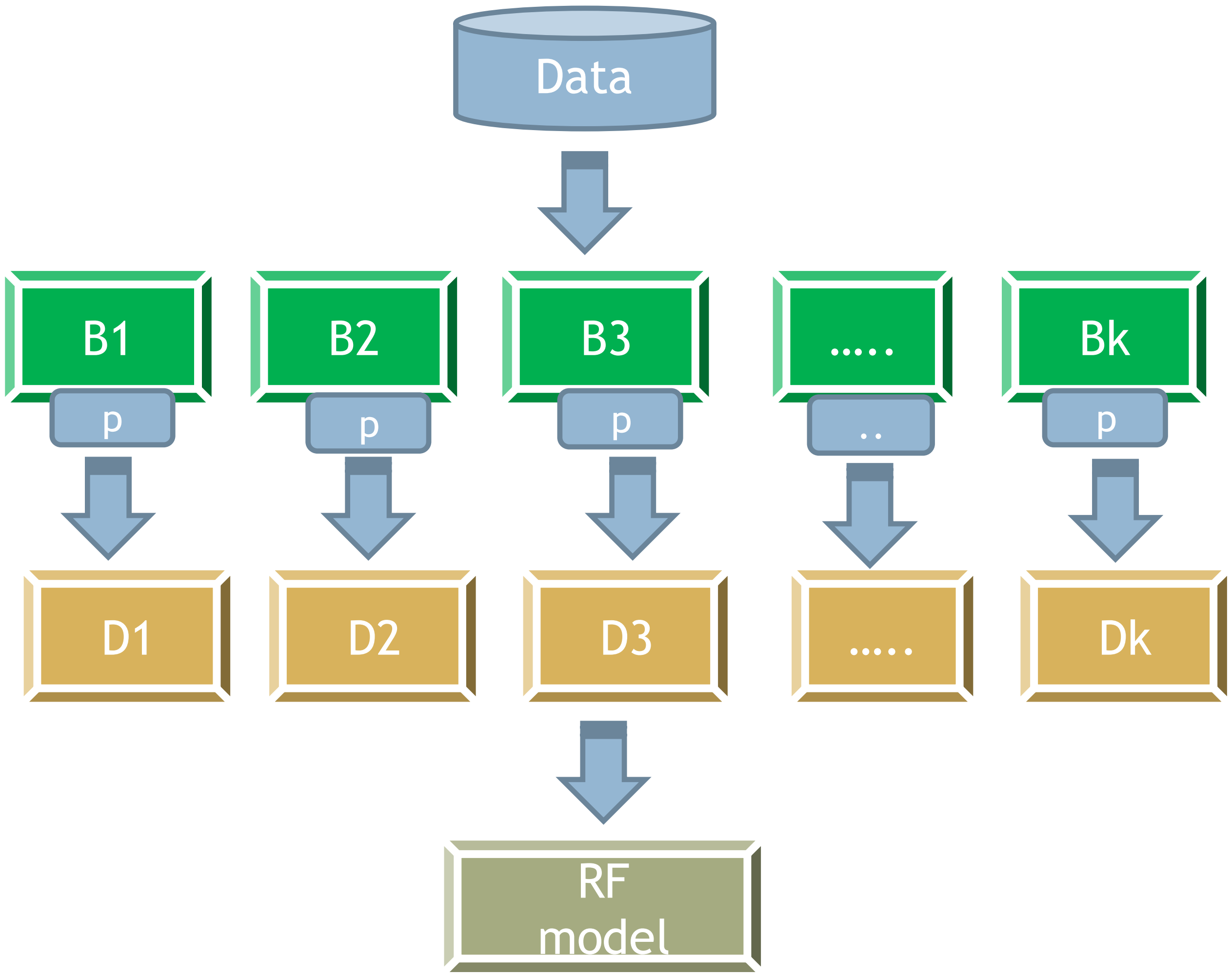
- Random forest is a specific case of bagging methodology. Bagging on decision trees is random forest
- Like many trees form a forest, many decision tree model together form a Random Forest model

Random Forest

- In random forest we induce two types of randomness
  - Firstly, we take the boot strap samples of the population and build decision trees on each of the sample.
  - While building the individual trees on boot strap samples, we take a subset of the features randomly
- Random forests are very stable they are as good as NN and SVMs sometimes better

Random Forest algorithm

- The training dataset D with t number of features
- Draw k boot strap sample sets from dataset D
- For each boot strap sample i
  - Build a decision tree model M_i using only p number of features (where p<<t)
  - Each tree has maximal strength they are fully grown and not pruned.
  - We will have total of k decision treed M_1 , M_2 ,….. M_k; Each of these trees are built on reactively different training data and different set of features
  - Vote over for the final classifier output and take the average for regression output



The Random Factors in Random Forest

- We need to note the most important aspect of random forest, i.e inducing randomness into the bagging of trees. There are two major sources of randomness
  - Randomness in data: Boot strapping, this will make sure that any two samples data is somewhat different
  - Randomness in features: While building the decision trees on boot strapped samples we consider only a random subset of features.

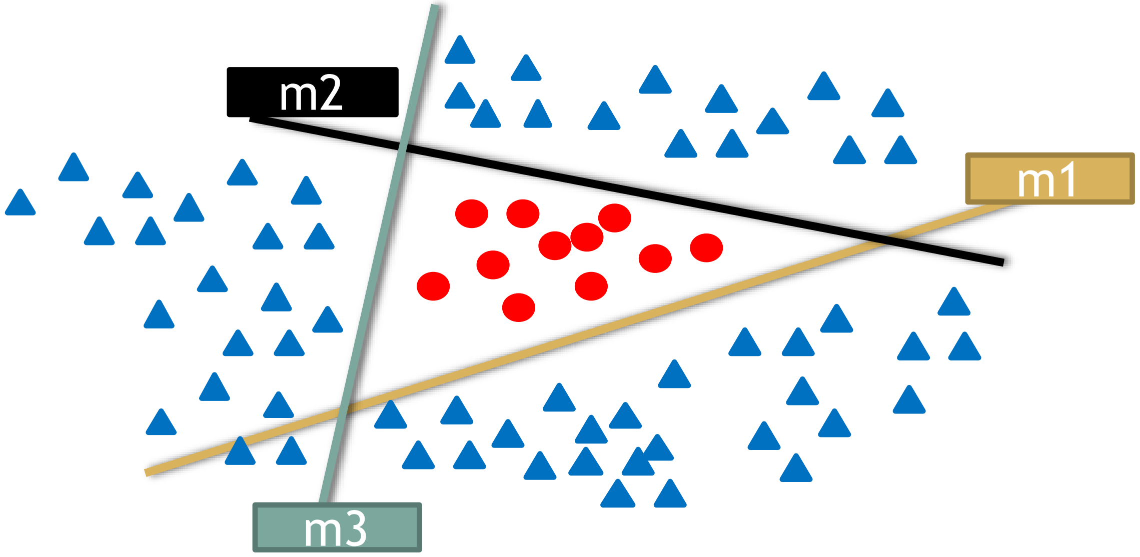
Why to induce the randomness?

- The major trick of ensemble models is the independence of models.
- If we take the same data and build same model for 100 times, we will not see any improvement
- To make all our decision trees independent, we take independent samples set and independent features set
- As a rule of thumb we can consider square root of the number features, if ‘t’ is very large else p=t/3

Why Random Forest Works

- For a training data with 20 features we are building 100 decision trees with 5 features each, instated of single great decision.
- The individual trees may be weak classifiers.
- Its like building weak classifiers on subsets of data. The grouping of large sets of random trees generally produces accurate models.

Why Random Forest Works



- In this example we have three simple classifiers.
  - m1 classifies anything above the line as +1 and below as -1
  - m2 classifies all the points above the line as -1 and below as +1
  - m3 classifies everything on the left as -1 and right as +1
- Each of these models have fair amount of misclassification error.
- All these three weak models together make a strong model.

Car accidents IOT

LAB: Random Forest

statinfer

LAB: Random Forest

- https://www.kaggle.com/c/stayalert
- Dataset: /Car Accidents IOT/Train.csv
- Build a decision tree model to predict the fatality of accident
- Build a decision tree model on the training data.
- On the test data, calculate the classification error and accuracy.
- Build a random forest model on the training data.
- On the test data, calculate the classification error and accuracy.
- What is the improvement of the Random Forest model when compared with the single tree?

Code: Random Forest

```
features=list(car_train.columns[1:22])
X_train=car_train[features]
y_train=car_train['Fatal']

###buildng Decision tree on the training data ####
clf = tree.DecisionTreeClassifier()
clf.fit(X_train,y_train)

#####predicting on test data ####
tree_predict=clf.predict(car_test[features])

from sklearn.metrics import confusion_matrix###for using confusion matrix###
cm1 = confusion_matrix(car_test[['Fatal']],tree_predict)
print(cm1)
```

Import data and build a decision tree

Code: Random Forest

```
#####predicting on test data ####
tree_predict=clf.predict(car_test[features])

from sklearn.metrics import confusion_matrix###for using confusion matrix###
cm1 = confusion_matrix(car_test[['Fatal']],tree_predict)
print(cm1)

#####from confusion matrix calculate accuracy
total1=sum(sum(cm1))
accuracy_tree=(cm1[0,0]+cm1[1,1])/total1
accuracy_tree
```

Code: Random Forest

```
from sklearn.ensemble import RandomForestClassifier
forest=RandomForestClassifier(n_estimators=10, max_features=5, max_depth=11)

forest.fit(X_train,y_train)

predict_y_test=forest.predict(car_test[features])
actual_y_test=car_test['Fatal']

###check the accuracy on test data
from sklearn.metrics import confusion_matrix###for using confusion ma
cm2 = confusion_matrix(actual_y_test,predict_y_test)
print(cm2)
total2=sum(sum(cm2))

#####from confusion matrix calculate accuracy
accuracy_forest=(cm2[0,0]+cm2[1,1])/total2
accuracy_forest
```

Random forest model and its accuracy

```
....
[[3392  500]
 [ 484 4689]]
Out[179]: 0.89145063430777716
```

Thank you