Bagging Predictors
-LEO BREIMAN

Anuja Saira Abraham (5204982)
Alessia Marzotti (5108443)
Miriam Mercuri (5207057)

Introduction

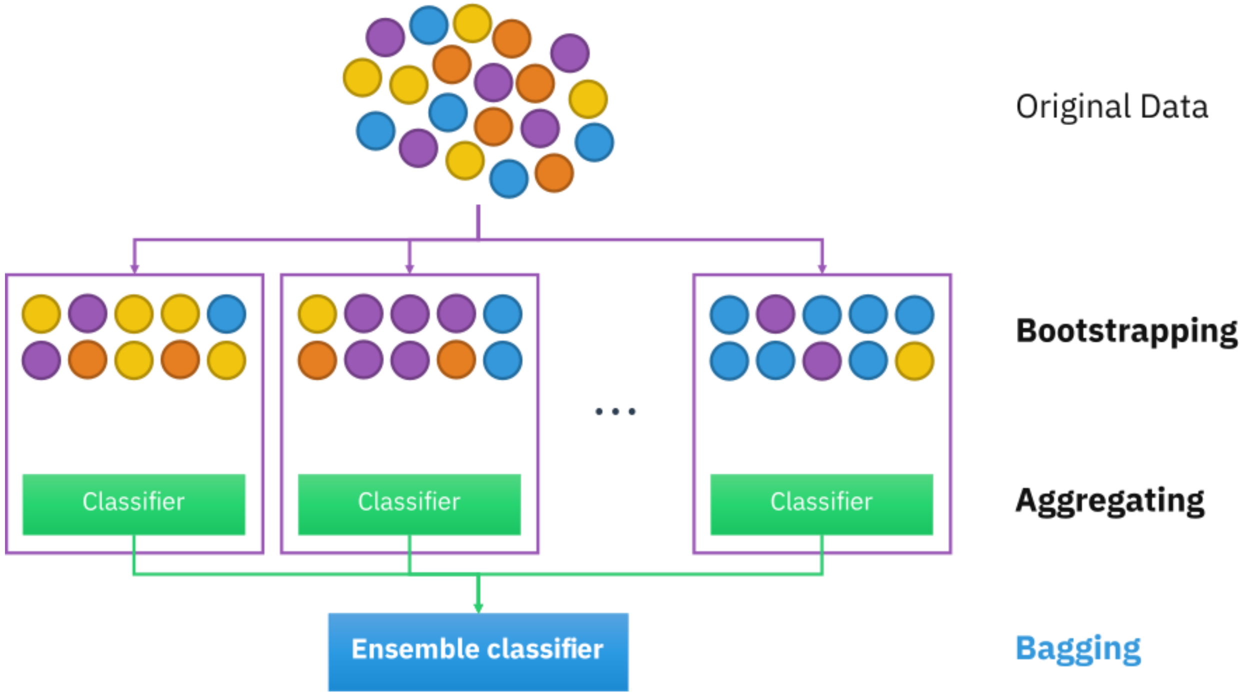
- When creating a prediction model there is always a trade-off between **BIAS** and **VARIANCE**.

- The former refers to the error caused due to simplification and the latter refers to the error cause by prediction variation for different data.

- To improve the fit of a prediction model Breiman (1996) introduced a technique called **BAGGING**.

- *"Instead of relying on a single model for fitting the model, here multiple bootstrap samples of data are created, and each sample has to fit the model, and the predictions from all the fitted models are averaged to obtain the bagged prediction."*

What is Bagging? Why Do We Need It ?



- Bagging is a method for generating multiple versions of a predictor and using these to get an **aggregated predictor**

- Multiple versions of a predictor is formed by making **bootstrap replicates** of the learning set and using these as new learning sets

- Bagging can give a gain in accuracy if the prediction method is **unstable**

Decision Tree

Decision tree is a **decision support** hierarchical model which consists of a root node, branches, internal nodes and leaf nodes that lead to predictions.

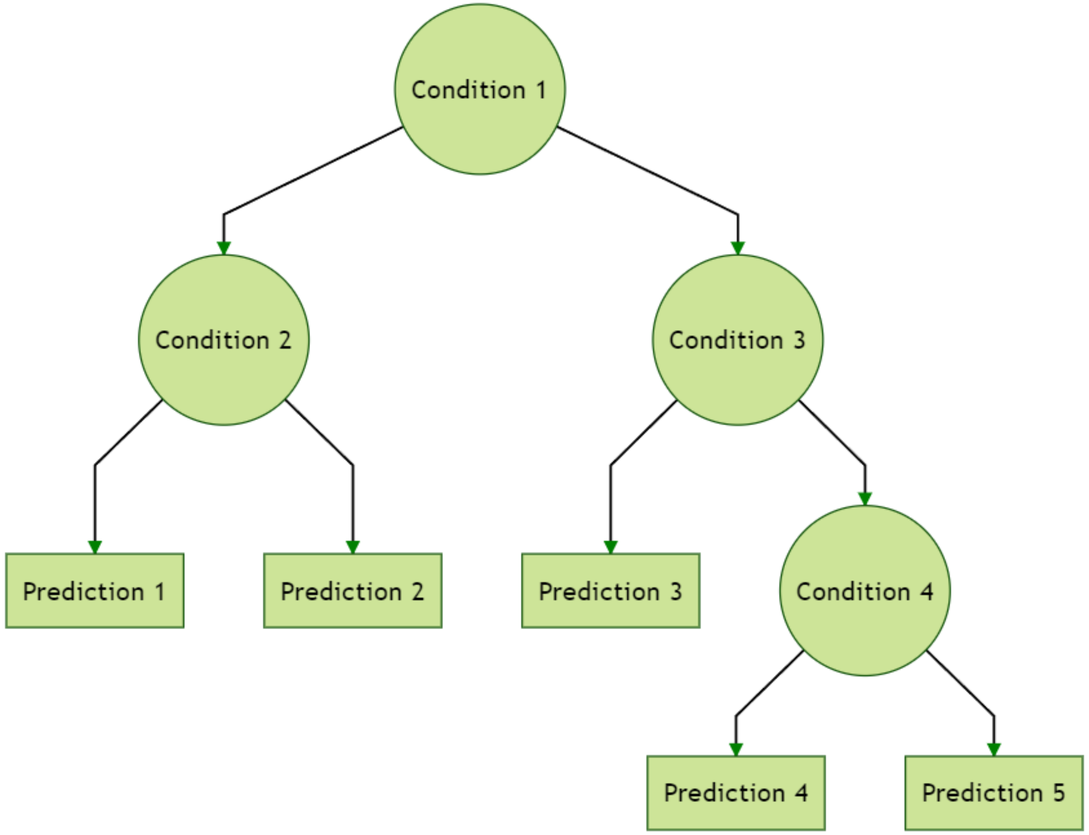
Classification And Regression Trees (**CART**).

- Classification Tree

  Target variable = categorical value

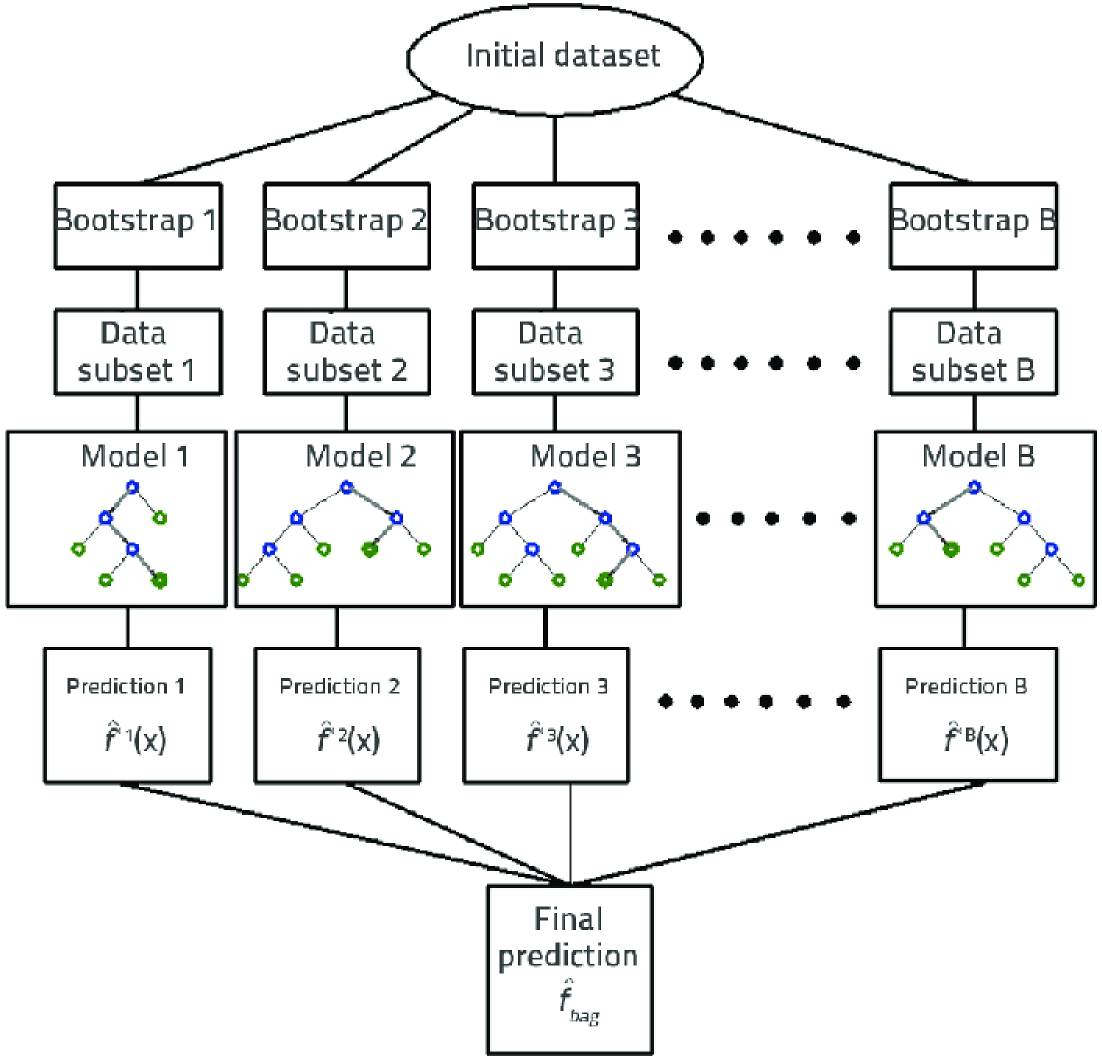
- Regression Tree

  Target variable = continuous value



Bagged Decision Tree

- Dataset sample is split into **learning** and **test** set.
- A **bootstrap** sample L_B is selected from L and a tree grown.
- This is **repeated** multiple times (i) giving tree predictors $\Phi_1(x),\ldots,\Phi_i(x)$.
- Bagged regression predictor $y_n = av_k\Phi_k(x_n)$ (**averaging**).
- Bagged classification predictor is estimated as which class has the plurality in $\Phi_1(x),\ldots,\Phi_i(x)$ (**voting**).


Initial dataset
Bootstrap 1
Bootstrap 2
Bootstrap 3
Bootstrap B
Data subset 1
Data subset 2
Data subset 3
Data subset B
Model 1
Model 2
Model 3
Model B
Prediction 1 $\hat{f}^1(x)$
Prediction 2 $\hat{f}^2(x)$
Prediction 3 $\hat{f}^3(x)$
Prediction B $\hat{f}^B(x)$
Final prediction $\hat{f}_{bag}$

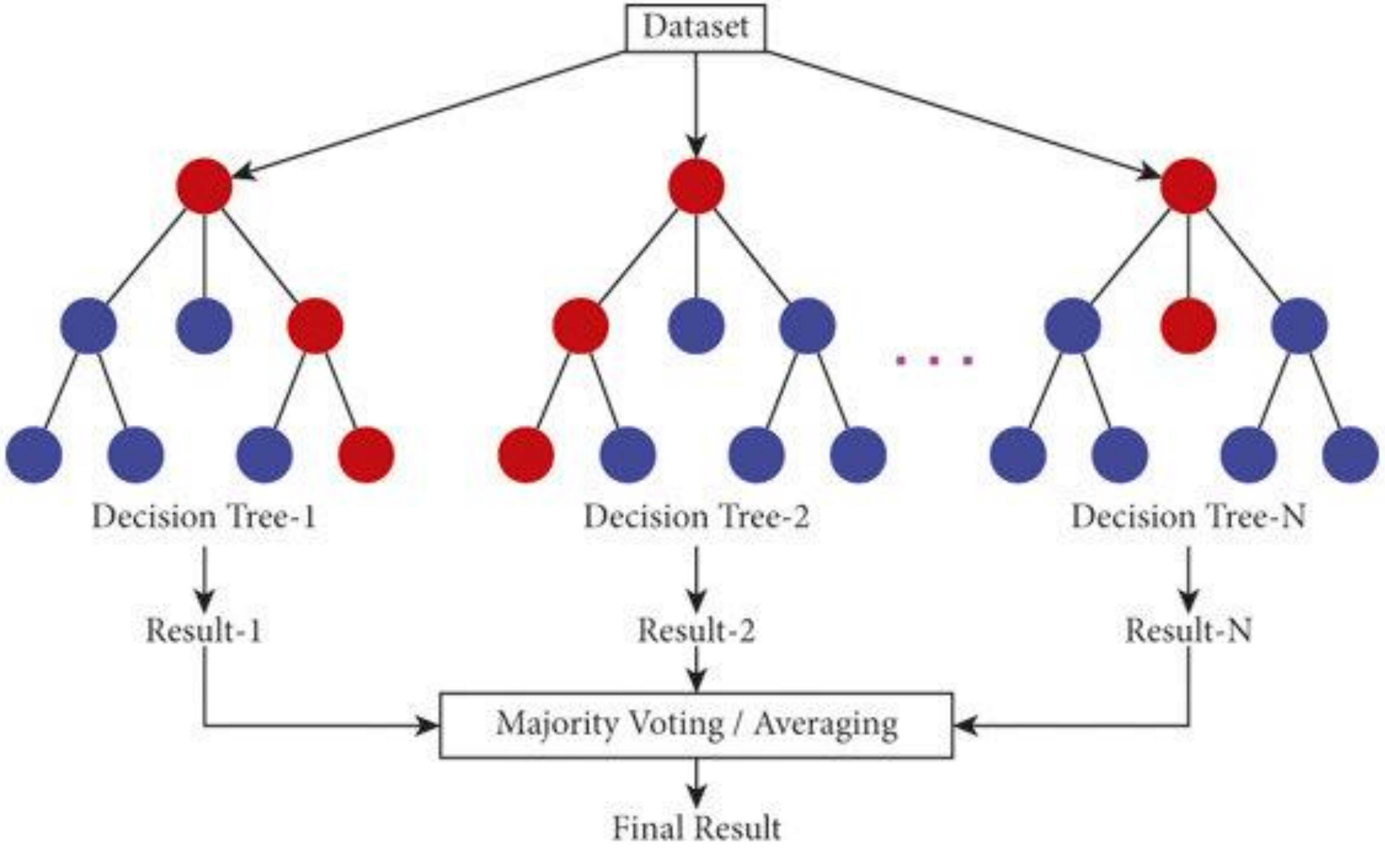
Random Forest

- Random forests are an **improvement** over bagged trees.

- Aims to make the trees **less similar** to each other.

- We randomly select a subset of predictors and choose only one from that subset for the split.

- We refresh this subset for each split, usually picking the square root of the total number of predictors.



Why Bagging Works? Numeric Prediction

- Let each (y, $\mathbf{x}$) case in dataset $\mathcal{L}$ be independently drawn from the probability distribution P. Suppose y is numerical, and $\phi(\mathbf{x}, \mathcal{L})$ is the predictor.
- The aggregated predictor is $\phi_a(\mathbf{x}, P) = E[\phi(\mathbf{x}, \mathcal{L})]$.
- Take Y, $\mathbf{X}$ to be random variables having distribution P and independent of $\mathcal{L}$. The average prediction error e in $\phi(\mathbf{x}, \mathcal{L})$ is $E_L[E_{y,x}[(y - \phi(\mathbf{x}, \mathcal{L}))^2]]$. Define the error in the aggregated predictor to be $e_a = E_{y,x}[(y - \phi_A(\mathbf{x}, P))^2]$.
- Using the inequality $(EZ)^2 \leq E(Z)^2$ gives $e = E(Y^2) - 2E(Y)\phi_a + E(E[\phi^2(\mathbf{X}, \mathcal{L})]) \geq E[(Y - \phi_a)^2 = e_a$. Thus, ϕ_a has a lower mean squared prediction error than ϕ.
- How much lower depends on how unequal the two sides of $[E_L\phi(\mathbf{x}, \mathcal{L})]^2 \leq E_L[\phi^2(\mathbf{x}, \mathcal{L})]$ are.
- The effect of instability is clear:
  - If $\phi(\mathbf{x}, \mathcal{L})$ does not change too much with replicate $\mathcal{L}$, the two sides will be nearly equal, and aggregation will not help. But $\phi_a(\mathbf{x}, \mathcal{L})$ always improves on ϕ.
- Now, the bagged estimate is not $\phi_a(\mathbf{X}, P)$, but rather $\phi_B = \phi_A(\mathbf{X}, \mathcal{P}_L)$. If the procedure is stable, ϕ_B can improve through aggregation.

Classification

In classification, predictor $\Phi(\mathbf{x},L)$ predicts class label $j \in \{1,\ldots,J\}$ with probability

$$\mathbf{Q(j|x) = P(\Phi(x,L)=j)}$$

Then the overall probability of correct classification for L (learning set) is

$$\mathbf{r= \int [Q(j|x)P(j|x)]P_x(dx)}$$

Φ is **order-correct** at the input x: $\quad \text{argmax}_j Q(j|x) = \text{argmax}_j P(j|x)$

The aggregator predictor, $\mathbf{\Phi_A(x) = argmax_j Q(j|(x))}$

C is a set of order-correct predictors: $C = \{x \,;\, \text{argmax}_j P(j|x) = \text{argmax}_j Q(j|x)\}$

Then the correct classification probability of Φ_A is :

$$\mathbf{r_A= \int_{x \in C} max_j P(j|x)]P_x(dx) + \int_{x \in C'} [\Sigma_j \mathbb{1}(\Phi_A(x) = j)P(j|x)]P_x(dx)}$$

Even if Φ is order-correct at $\mathbf{x}$ its correct classification rate can be far from optimal.

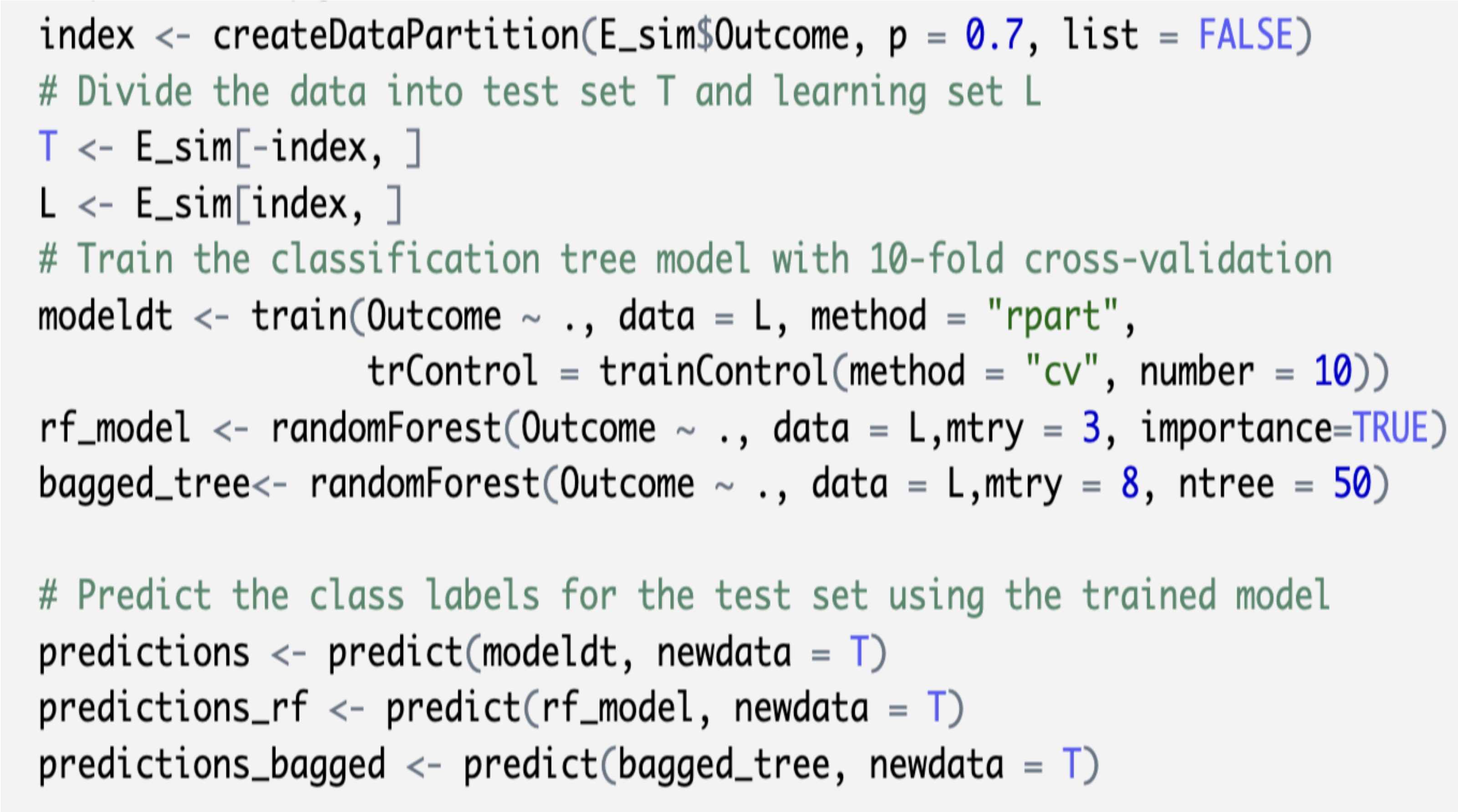
- Bagging **unstable** classifiers usually **improves** them.
- Bagging **stable** classifiers is **not a good idea**.

Example in R

We are going to use the diabetes dataset gathered among the Pima Indians by the National Institute of Diabetes and Digestive and Kidney Diseases and compare the result of a classification tree, bagged and random forest.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0

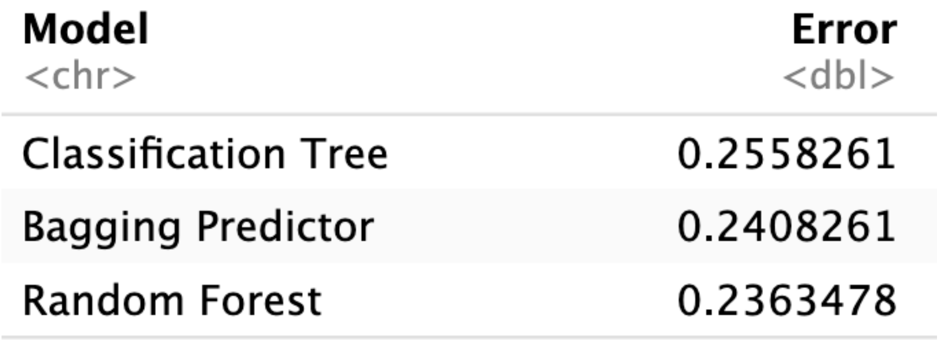
Code Snippet



```
index <- createDataPartition(E_sim$Outcome, p = 0.7, list = FALSE)
# Divide the data into test set T and learning set L
T <- E_sim[-index, ]
L <- E_sim[index, ]
# Train the classification tree model with 10-fold cross-validation
modeldt <- train(Outcome ~ ., data = L, method = "rpart",
                 trControl = trainControl(method = "cv", number = 10))
rf_model <- randomForest(Outcome ~ ., data = L,mtry = 3, importance=TRUE)
bagged_tree<- randomForest(Outcome ~ ., data = L,mtry = 8, ntree = 50)

# Predict the class labels for the test set using the trained model
predictions <- predict(modeldt, newdata = T)
predictions_rf <- predict(rf_model, newdata = T)
predictions_bagged <- predict(bagged_tree, newdata = T)
```

Result



Model <chr>	Error <dbl>
Classification Tree	0.2558261
Bagging Predictor	0.2408261
Random Forest	0.2363478

A Linear Regression Illustration

Forward variable selection

- Data: L=(y_n, $\mathbf{x_n}$), n=1...N with $\mathbf{x}$=(x_1,…,x_m) and Predictors: $\Phi_1(x)$...$\Phi_M(x)$
- Unstable procedure

Simulation structure

- Model: y= $\Sigma_m \beta_m x_m$+ε, ε~N(0,1), M=30, n=60
- Subset selection is poor if there are many small but

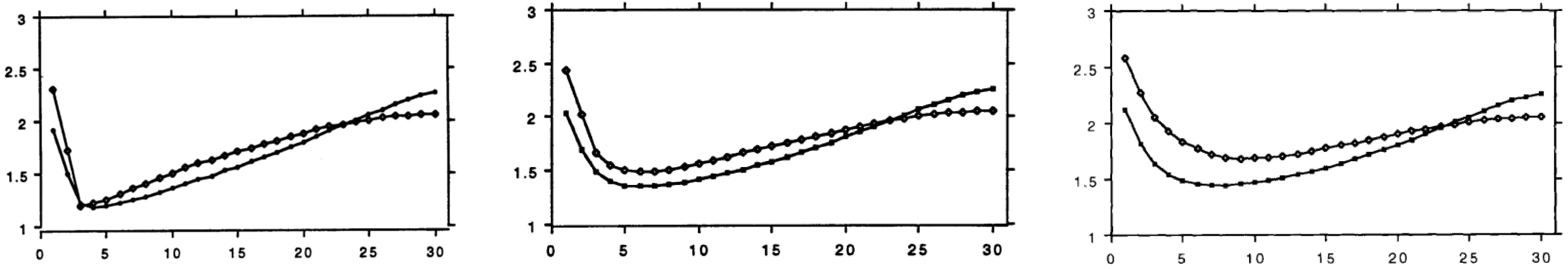
 non-zero β_m

- Bagging can improve only if the unbagged is not optimal

Prediction Error for Subset Selection and Bagged Subset Selection vs. Number of Variables

—◇— subset selection

—●— bagged subset selection

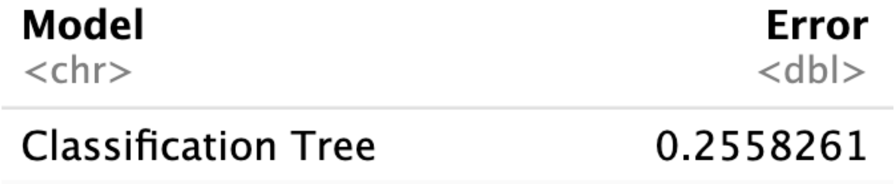


3 NONZERO COEFFICIENTS

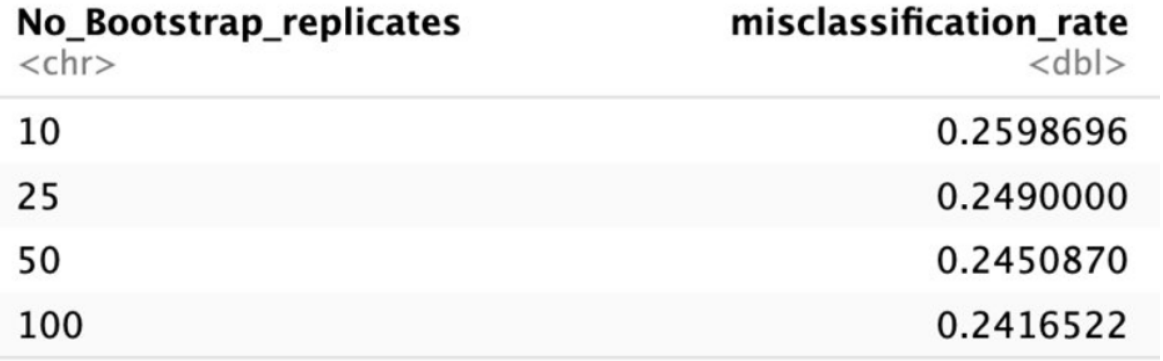
15 NONZERO COEFFICIENTS

27 NONZERO COEFFICIENTS

How many Bootstrap Replicate are enough?



Model	Error
<chr>	<dbl>
Classification Tree	0.2558261



No_Bootstrap_replicates	misclassification_rate
<chr>	<dbl>
10	0.2598696
25	0.2490000
50	0.2450870
100	0.2416522

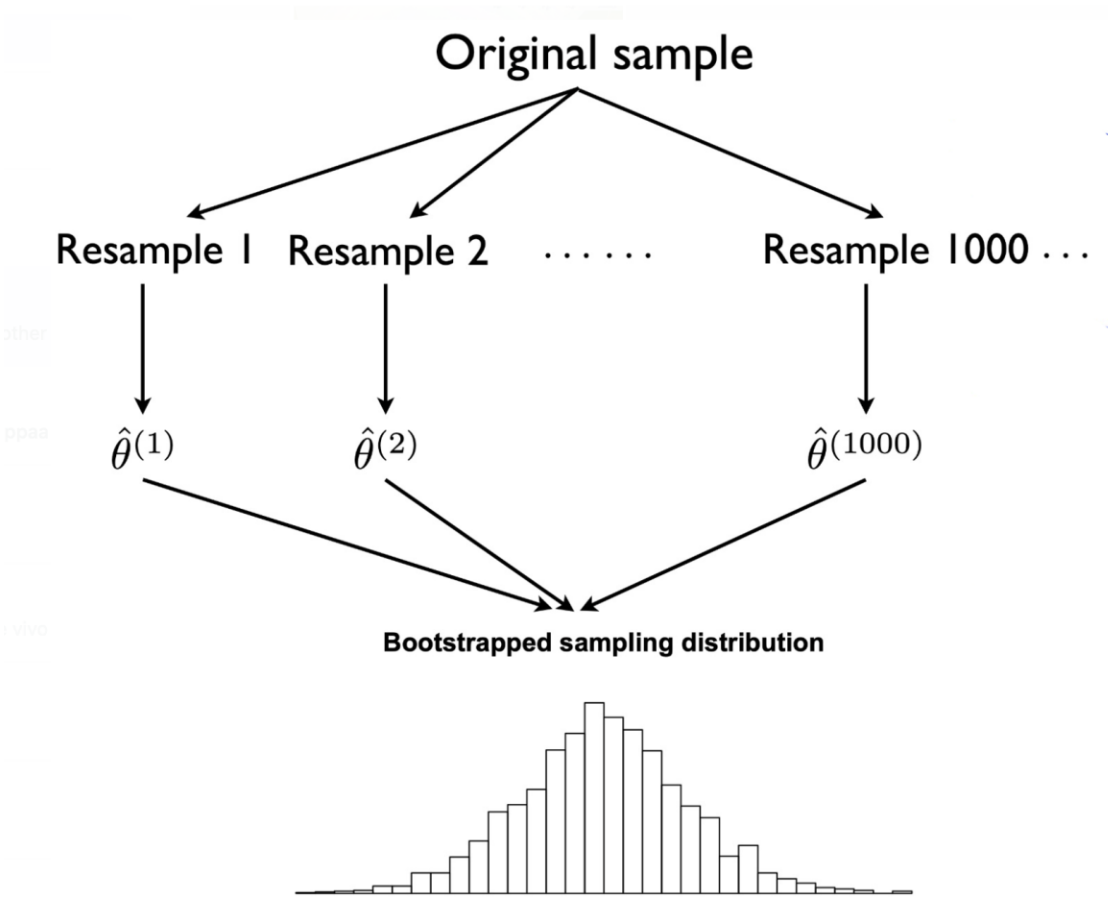
Results from Diabetes Dataset

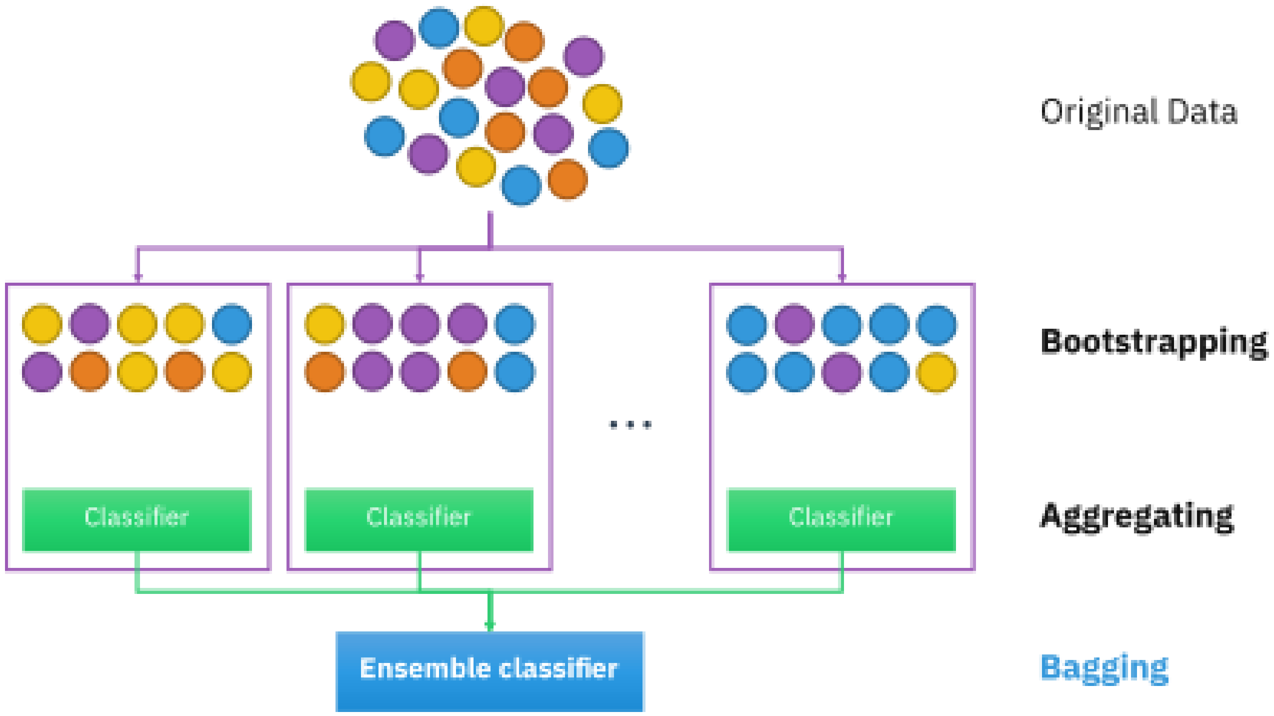
How big the Learning Set should be?

- Bootstrap learning sets were used of the **same size as the initial learning set**.
- It leaves out some instances.
- With a bootstrap learning sets twice the size of L there was no improvement in accuracy.

Bootstrap

Bagging (Bootstrap Aggregation)





Conclusions

- In summary, bagging is a valuable technique for improving the accuracy of a predictive model.

- Our application in R demonstrated the effectiveness of Bagging in reducing errors when dealing with unstable datasets, such as the "diabetes" dataset, using classification trees.

- Additionally, we explored the Random Forest technique and compared it with Bagging.

- Finally, we compared the concepts of Bootstrap and Bagging.