UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione

Statistical learning applied to graduate admission dataset

Silviu Filote 1059252, Jonathan Bommarito 1068755

Introduction

- Apply some approaches described in “An Introduction to Statistical Learning” to the chosen dataset
- Select the best model which can explain better our response with the lowest complexity
- Focus on the interpretability rather than optimal model

Dataset

- Dataset: Graduate Admission 2 (available on Kaggle)
- 500 rows without missing values
- In the dataset there are 8 variables, 4 continuous and 4 discrete

Continuous variables

- Graduate Record Examinations Scores (GRE): test that measures abstract thinking in areas such as reading comprehension, writing, and mathematics
- Test Of English as a Foreign Language (TOEFL): it’s the score of English language test;
- Cumulative Grade Point Average (CGPA): is based on all coursework completed for bachelor’s degree, representing overall average;
- Chance of Admit (Admit): percentages that indicates the probability of being admitted.

Discrete variables

- University Rating (UniRatings): It refers to the applicant's undergraduate university. It would be fair to say that not all universities carry the same repute around the world;
- Statement of purpose (SOP): is a short essay that highlights the educational background, achievements, and goals;
- Common Letter of Recommendation (LOR): score from 1 to 5 assigned to the letter of recommendation;
- Research Experience (Research): boolean that indicates any professional or academic research activity acquired in any research field in the public or private sector.

Empirical variables distribution

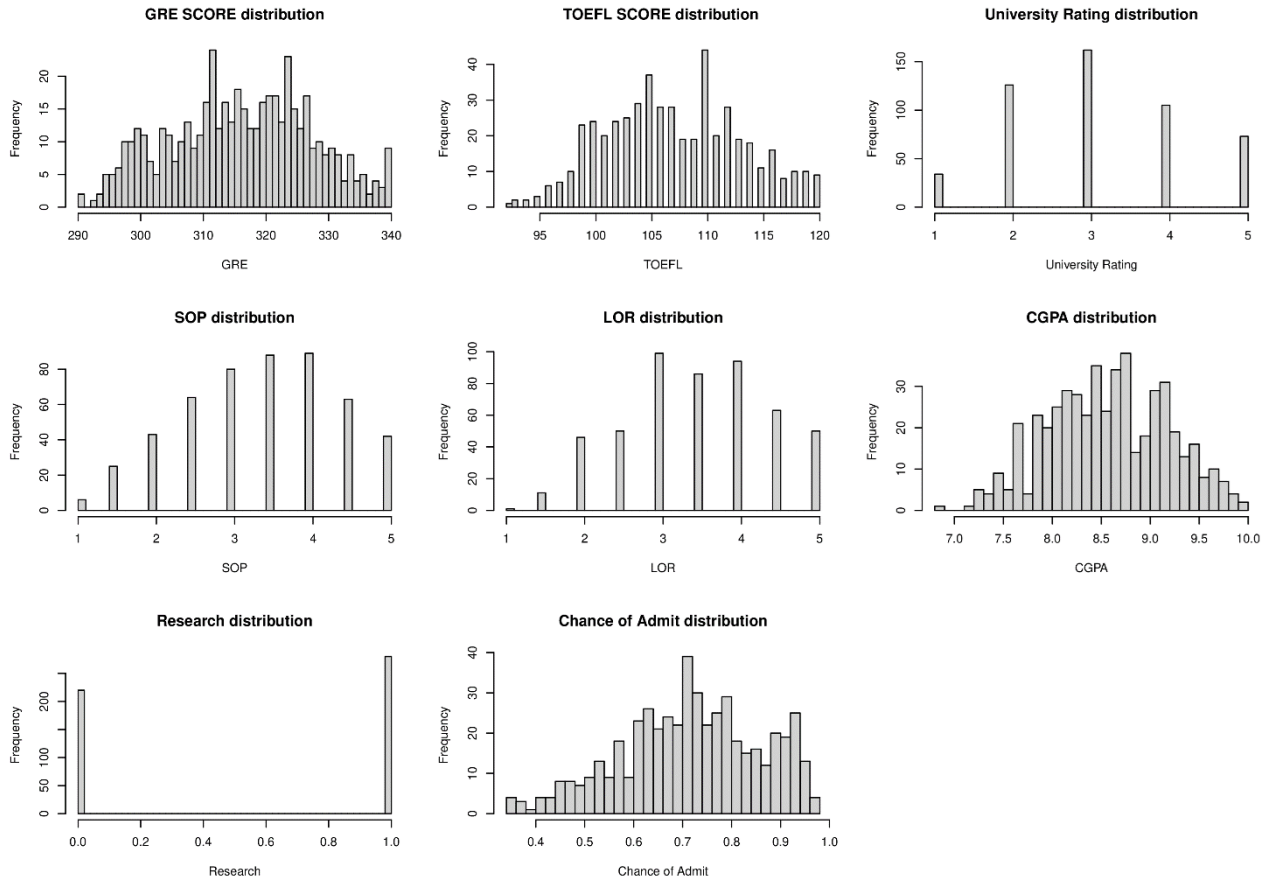


Figure 1: Empirical distribution of all the variables inside the dataset

Approaches implemented

- Linear regression: Estimated manually, Stepwise, Lasso, Model improvements
- Generalized Additive Models (GAMs)
- Regression trees: Pruned tree, Bagging, Random forest, Boosting

For each approach we will:

- Split dataset into: 70% training set 30% validation set of the dataset
- Perform residual analysis: Shapiro-Wilk and Breusch-Pagan tests

Outliers

The outliers are going to be removed from the dataset using IQR method.

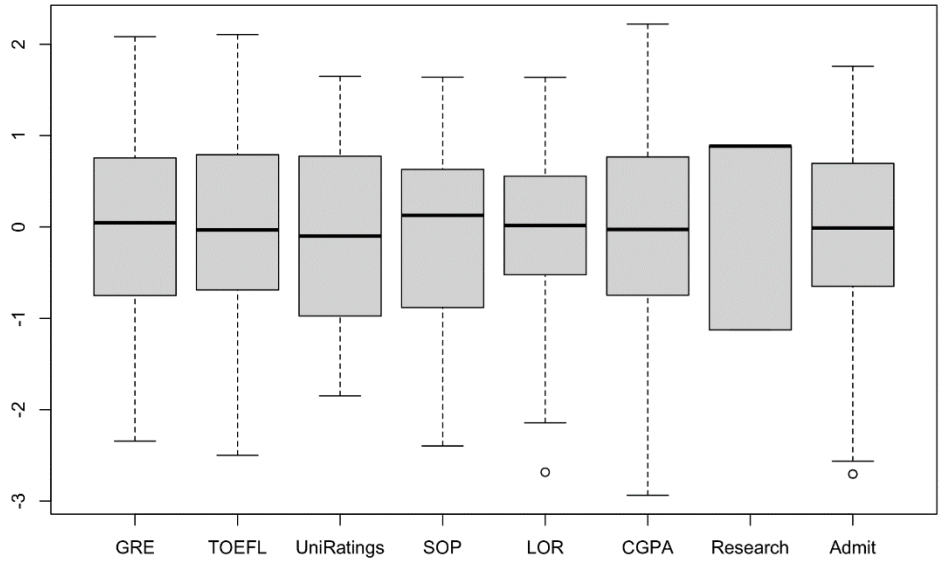
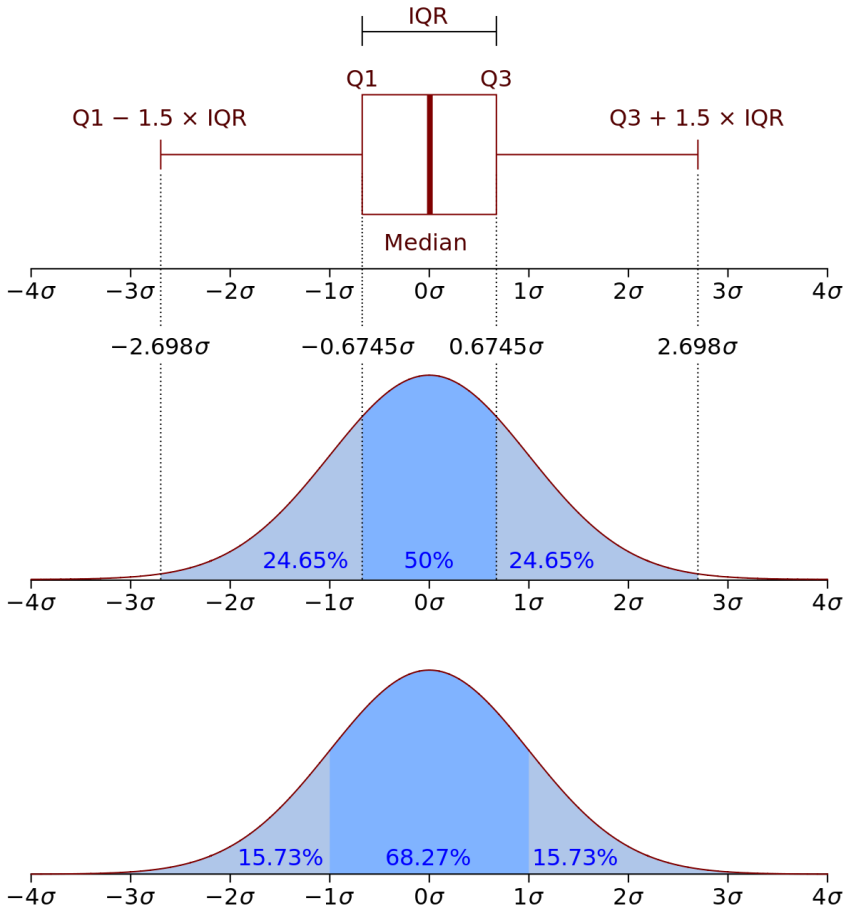


Figure 2: Detecting possible outliers in the regressors' distribution using boxplot

Figure 3: How IQR method works



Linear regression: Estimated manually

- We started including all regressors
- At each iteration we removed the least significant regressor
- Trade-off: R^2 adjusted and complexity

Linear regression: Estimated manually

```
lm_model <-  lm(formula = Admit ~ GRE + TOEFL + CGPA + LOR + Research,
                data = data, subset = train)
```

Table 1: Result of applying the linear model in order to explain the response variable using the most important regressors.

Coefficients:	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-1.257	0.113	-11.085	<2e-16	***
GRE	0.002	0.001	2.860	0.0045	**
TOEFL	0.002	0.001	2.639	0.0087	**
CGPA	0.129	0.011	11.918	<2e-16	***
LOR2	0.053	0.027	1.944	0.0527	.
LOR2.5	0.061	0.027	2.235	0.0260	*
LOR3	0.061	0.026	2.306	0.0217	*
LOR3.5	0.079	0.027	2.964	0.0033	**
LOR4	0.086	0.027	3.228	0.0014	**
LOR4.5	0.093	0.027	3.407	0.0007	***
LOR5	0.109	0.028	3.882	0.0001	***
Research1	0.029	0.007	3.973	8.71e-05	***

Linear regression: Estimated manually

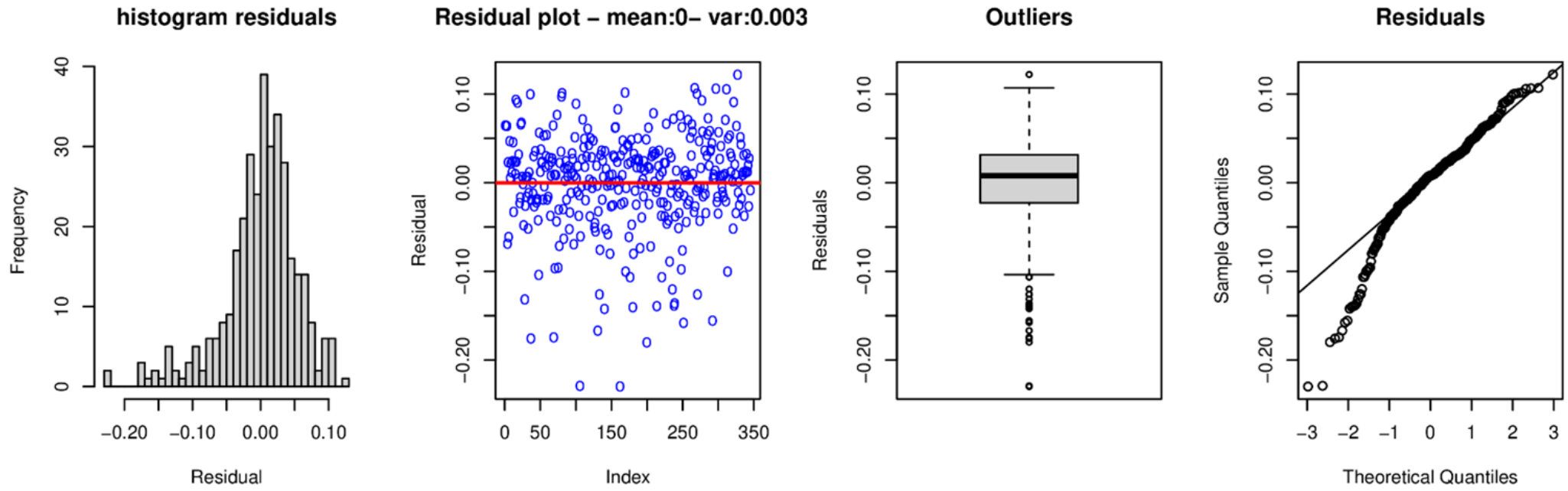


Figure 4: Residuals plotting made by the linear model

Linear regression: Estimated manually

- The Shapiro-Wilk test that the distribution of the residuals is non-normal
- Studentized Breusch-Pagan test confirms that the distribution of the residuals is heteroscedastic

Linear regression: Estimated manually

- We can’t trust the coefficients estimated by the model
- We implement boostrap to compare the coefficients values

Table 2: Boostrap test applied on the training dataset in order to estimate the coefficients of the linear model

Coefficients	original	std. error
(Intercept)	-1.257	0.131
GRE	0.002	0.001
TOEFL	0.002	0.001
CGPA	0.129	0.011
LOR2	0.053	0.026
LOR2.5	0.061	0.027
LOR3	0.061	0.025
LOR3.5	0.079	0.026
LOR4	0.086	0.025
LOR4.5	0.093	0.026
LOR5	0.109	0.024
Research1	0.029	0.009

Table 3: most important statistics of the linear model estimated before

Coefficients:	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-1.257	0.113	-11.085	<2e-16	***
GRE	0.002	0.001	2.860	0.0045	**
TOEFL	0.002	0.001	2.639	0.0087	**
CGPA	0.129	0.011	11.918	<2e-16	***
LOR2	0.053	0.027	1.944	0.0527	.
LOR2.5	0.061	0.027	2.235	0.0260	*
LOR3	0.061	0.026	2.306	0.0217	*
LOR3.5	0.079	0.027	2.964	0.0033	**
LOR4	0.086	0.027	3.228	0.0014	**
LOR4.5	0.093	0.027	3.407	0.0007	***
LOR5	0.109	0.028	3.882	0.0001	***
Research1	0.029	0.007	3.973	8.71e-05	***

Linear regression: Stepwise

- We include all the regressors and also: log, exponential, squared and cubic of all the continuous regressors
- Selection criterion: AIC
- Started with 42 coefficients and the algorithm selected only 20
- Only 10 regressors selected
- Residuals tests confirm again heteroscedasticity and non-normal distribution;

```
stepwise_model <-  lm(formula = Admit ~ UniRatings + LOR + Research + exp(CGPA) +
                                log10(GRE) + log10(TOEFL) + log10(CGPA) + poly(CGPA, 3),
                                data = data, subset = train)
```

Linear regression: Lasso

- Reduce complexity of the model
- Cross-validation on the penalization λ
- Using "`lambda.1se`" instead of "`lambda.min`"

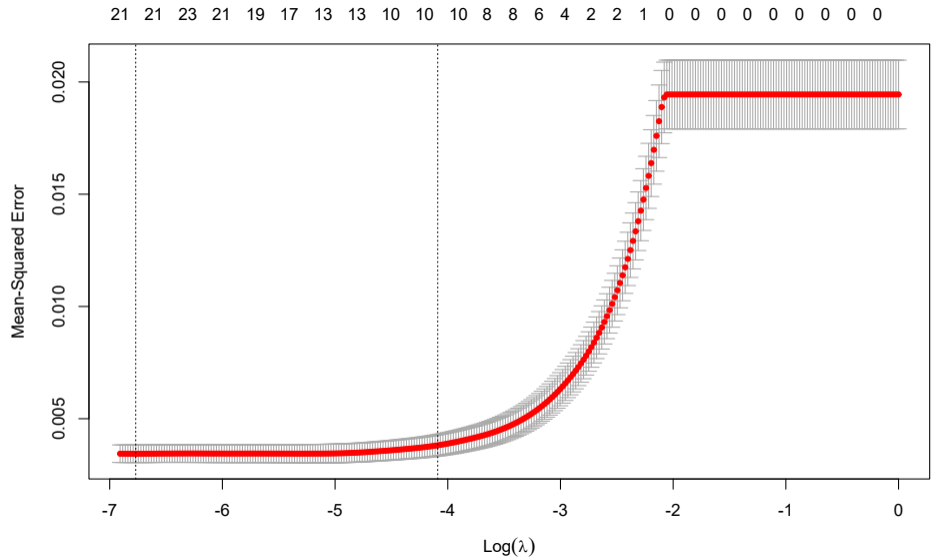


Figure 5: Cross-validation method applied on the training dataset for the purpose of choosing the best penalization (lambda) that reduces the model complexity. The two vertical lines highlight two possible values for lambda

Linear regression: Model improvements

- Same regressors of the “Estimated manually” model
- We applied $arcsin$ to the response variable in order to reduce variability of the residual distribution
- Breusch-Paga test unlike before confirms that residual distribution is now homoscedastic
- Increase of complexity
- Shapiro-Wilk test reveals that residuals aren’t normal

Final considerations about linear models

- The "`lm_model`" is the most interpretable model and the less complex one
- We will use this one to do other comparisons with other models

Table 4: All the linear regression models implemented during this section with the most important statistics

Model	RMSE test	CV RMSE	Shapiro-Wilk train	Breusch-Pagan train	Adj. R^2 train
`lm_model`	0.064	0.059	pv = 3.345e-11	pv = 0.002	0.837
`stepwise_model`	0.064	0.059	pv = 3.999e-11	pv = 0.008	0.841
`lasso_model`	0.066	0.060	pv = 1.628e-08		0.809
`lm_imp_model`	0.140	0.067	pv = 1.004e-08	pv = 0.095	0.857

Generalized Additive Models

- GAM with all regressors
- GAM with only significant regressors
- GAM with the most performing degrees of freedom

GAM with all regressors

```
gam_all_regressors <-  gam(formula = Admit ~ s(TOEFL) + s(CGPA) +  s(GRE) + Research +
                      SOP + LOR + UniRatings, data = data, subset = train)
```

Table 5: GAM with all regressors

	Df	Sum Sq	Mean Sq	F value	Pr(>|f|)	
s(TOEFL)	1	4.272	4.272	1375.793	<2.2e-16	***
s(CGPA)	1	1.166	1.166	375.439	<2.2e-16	***
s(GRE)	1	0.045	0.045	14.634	0.0002	***
Research	1	0.055	0.055	17.606	3.547e-05	***
SOP	8	0.062	0.008	2.507	0.0118	*
LOR	7	0.049	0.007	2.248	0.0304	*
UniRatings	4	0.027	0.007	2.193	0.0696	.
Residuals	312	0.969	0.003			

Table 6: non-linear splines component in GAM with all regressors

	Df	Npar F	Pr(F)	
s(TOEFL)	3	1.339	0.262	
s(CGPA)	3	2.778	0.041	*
s(GRE)	3	1.109	0.346	

GAM with all regressors

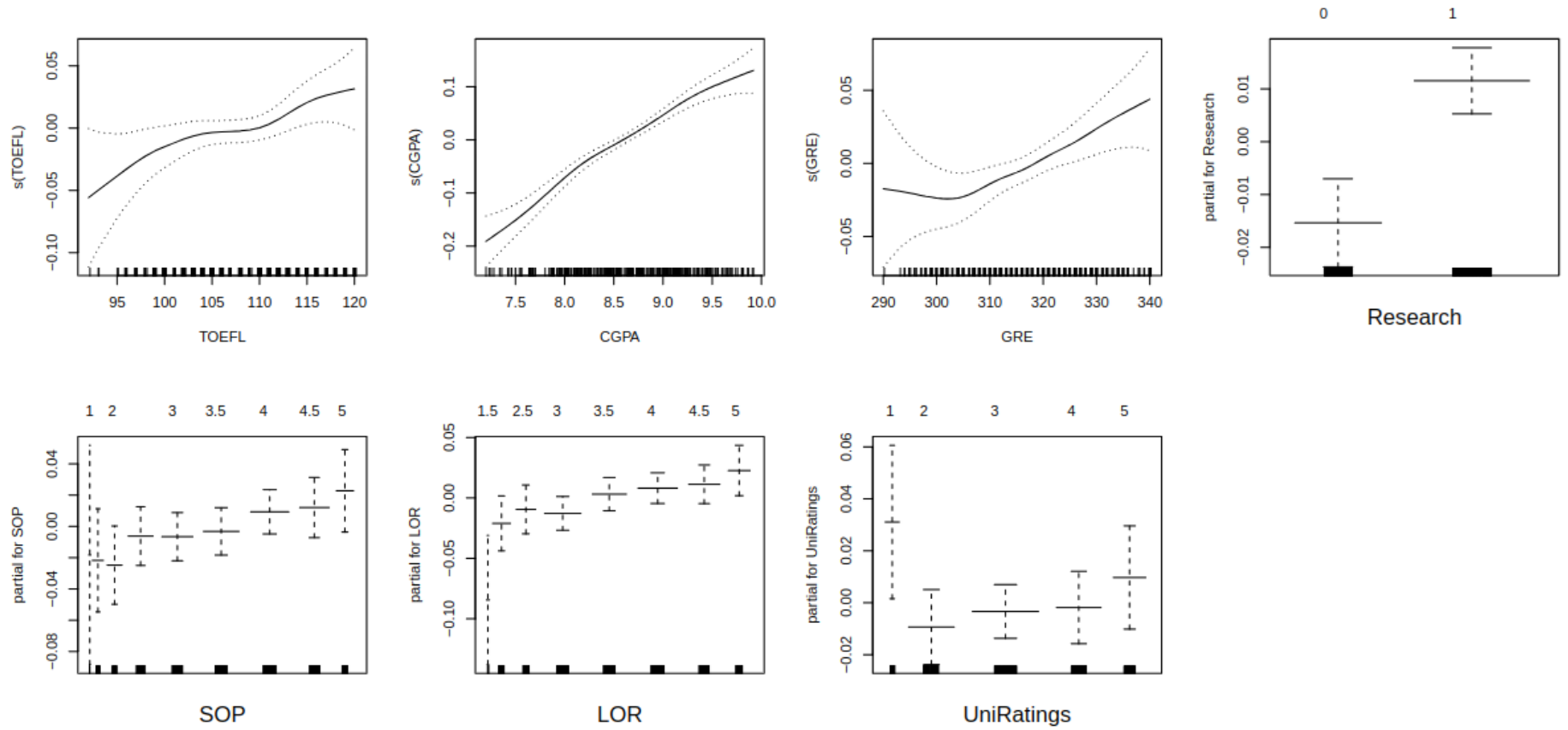


Figure 6: GAM with all regressors

GAM with only significant regressors

```
gam_sigreg <- gam(formula = Admit ~ s(TOEFL) + s(CGPA) + s(GRE) + Research,
                  data = data, subset = train)
```

Table 7: GAM with only significant regressors

	Df	Sum Sq	Mean Sq	F value	Pr(>|f|)	
s(TOEFL)	1	4.230	4.230	1276.445	<2.2e-16	***
s(CGPA)	1	1.205	1.205	363.609	<2.2e-16	***
s(GRE)	1	0.040	0.040	11.920	6.275e-04	***
Research	1	0.054	0.054	16.334	6.607e-05	***
Residuals	331	1.097	0.003			

Table 8: non-linear splines component in GAM with only significant regressors

	Df	Npar F	Pr(F)	
s(TOEFL)	3	1.406	0.241	
s(CGPA)	3	2.170	0.091	.
s(GRE)	3	0.931	0.426	

GAM with the most performing degrees of freedom

```
# Ricerca del numero di nodi migliore
models = list()
for(i in 1:5){
  for(j in 1:5){
    for(l in 1:5){
      model <- gam(Admit~s(TOEFL,i)+s(CGPA,j)+s(GRE,l)+Research,data=data,subset = train)
      model <- RMSE_function(model)
      model$nodes <- c(i,j,l)
      models <- append(models,list(model))
    }
  }
}
```

GAM with the most performing degrees of freedom

```
RMSE_function <- function(m) {
  yhat <- predict(m,data = data[train])
  RMSE_train <- sqrt(mean((data$Admit[train]-yhat)^2))


  yhat <- predict(m, newdata = data[-train,])
  RMSE_test <- sqrt(mean((data$Admit[-train]-yhat)^2))

  m$RMSE_train <- RMSE_train
  m$RMSE_test <- RMSE_test
  return(m)
}
```

GAM with the most performing degrees of freedom

```
# ritorna il valore minimo del RMSE di test e il numero dei nodi
minRMSE <- function(){
  min <- models[[1]]$RMSE_test
  index <- 0
  for(i in 1:length(models)){
    if(models[[i]]$RMSE_test < min){
      min <- models[[i]]$RMSE_test
      index <- i
    }
  }
  print(min)
  print(models[[index]]$nodes)
}

minRMSE()
```

GAM with the most performing degrees of freedom

```
gam_suggested <- gam(formula = Admit ~ TOEFL + s(CGPA, 5) + GRE + Research, data =
                     data, subset = train)
```

Table 9: GAM suggested by the algorithm

	Df	Sum Sq	Mean Sq	F value	Pr(>|f|)	
TOEFL	1	4.212	4.212	1278.946	<2.2e-16	***
s(CGPA, 5)	1	1.227	1.227	372.395	<2.2e-16	***
GRE	1	0.041	0.041	12.519	4.595e-4	***
Research	1	0.050	0.050	15.126	1.212e-4	***
Residuals	336	1.107	0.003			

Table 10: non-linear splines component in GAM suggested by the algorithm

	Df	Npar F	Pr(>|f|)	
s(CGPA, 5)	4	2.3813	0.0514	.

Final considerations about GAMs

- Each model fails the Breusch-Paga and Shapiro-Wilk
- All the models have similar RMSE test value
- Linear model suits better our goals than the GAM models

Table 11: Comparing all estimated models

Model	RMSE train	RMSE test	Shapiro-Wilk train	Breusch-Pagan train
`lm_model`	0.05531	0.06425	pv = 3.345e-11	pv = 0.0019
`gam_all_regressors`	0.05299	0.06628	pv = 1.614e-11	pv = 0.0156
`gam_sigreg`	0.05639	0.06443	pv = 1.519e-11	pv = 0.0017
`gam_suggested`	0.05663	0.06397	pv = 9.060e-12	pv = 0.0017

Regression trees

- In this section we are going to apply trees to our dataset
- Also, most complex algorithms based on trees: Bagging, Random forest and Boosting
- Apply optimization approaches where possible
- Compare all the tree-based models and choose the best one that suits our goals

Regression trees: Pruned tree

- First regression tree to complex to read
- We applied the pruning method
- The penalization factor that involves pruning is chosen using a cross-validation approach

Regression trees: Pruned tree

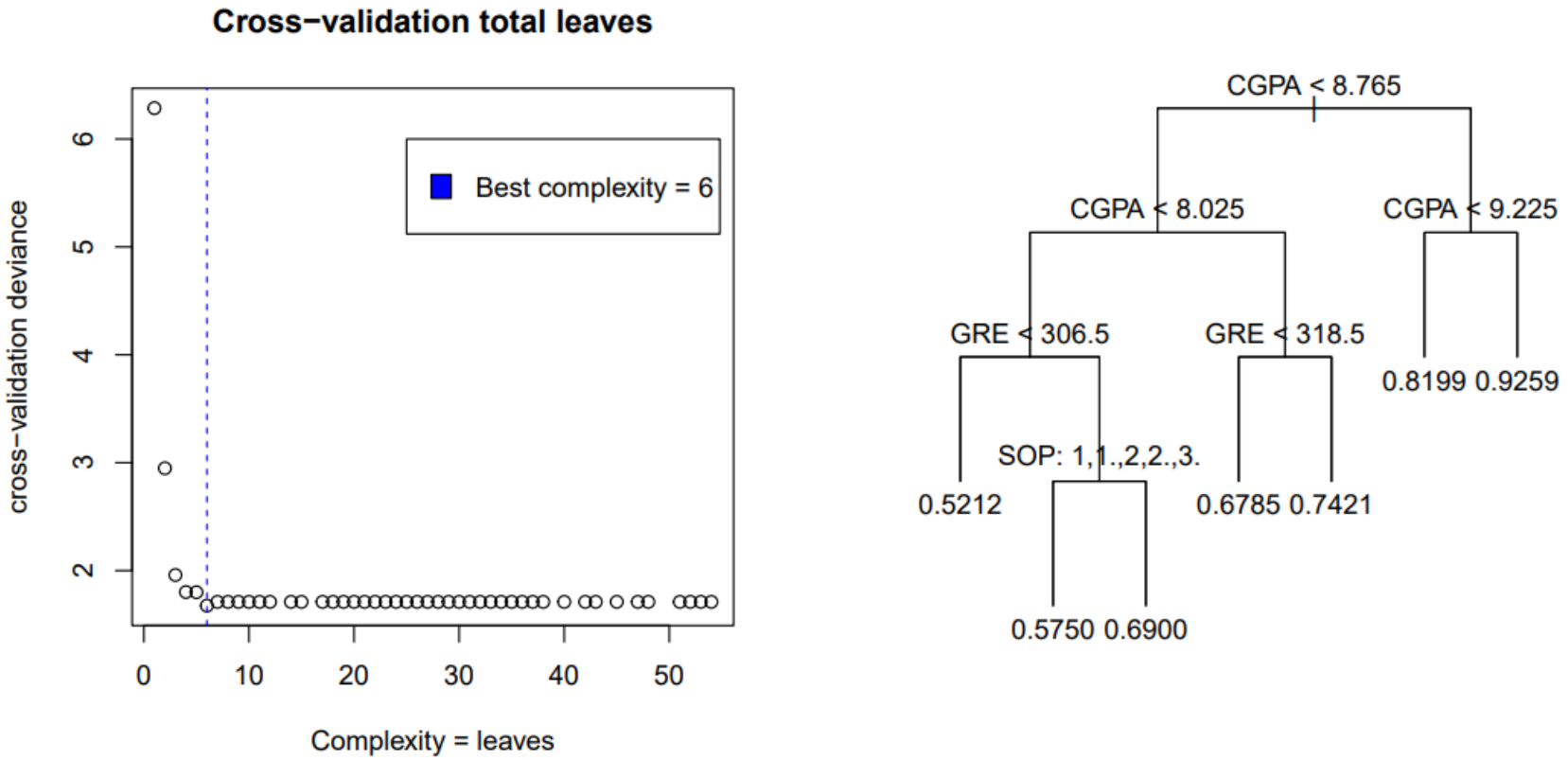


Figure 7: **Left:** The cross-validation chooses the number of total leaves of the tree, that indicate the complexity of the tree and where to stop the splitting - **Right:** Tree pruned model using the best complexity parameter discovered in cross-validation

Regression trees: Pruned tree

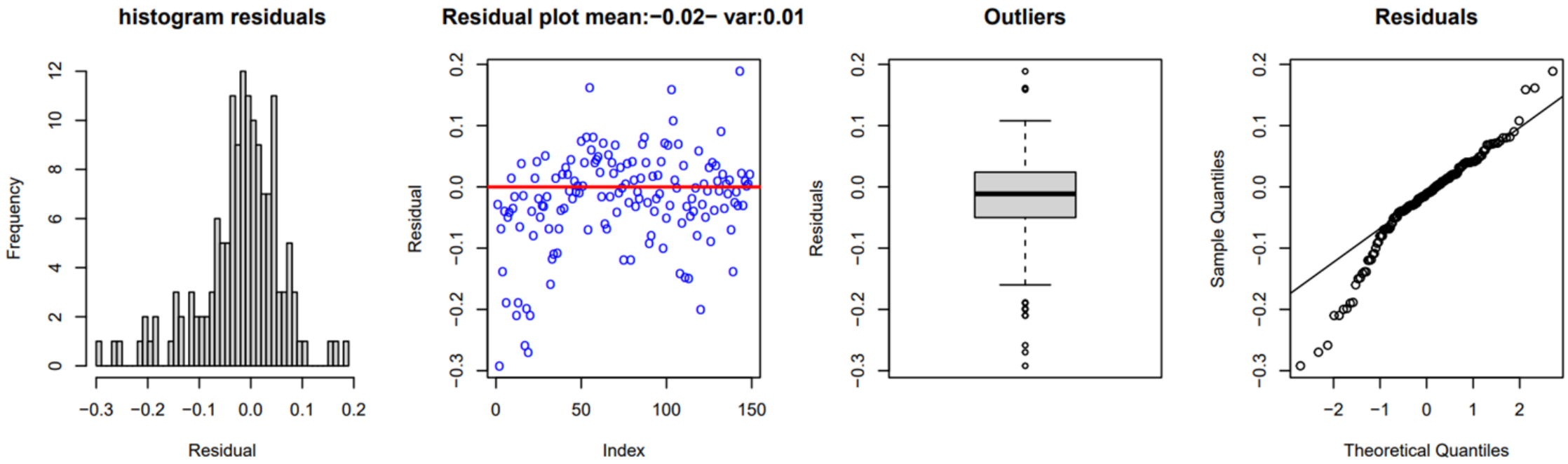


Figure 8: Residuals obtained in the validation/prediction phase from tree model after pruning

Regression trees: Bagging

- Bagging, also known as Bootstrap aggregating, is an ensemble learning technique
- Bagging models performs the splits of each tree considering all the regressors in the dataset
- It’s important to carefully choose the right number of trees we want to build in the training phase to not achieve any overfit
- Optimization: optimal number of trees

Regression trees: Bagging

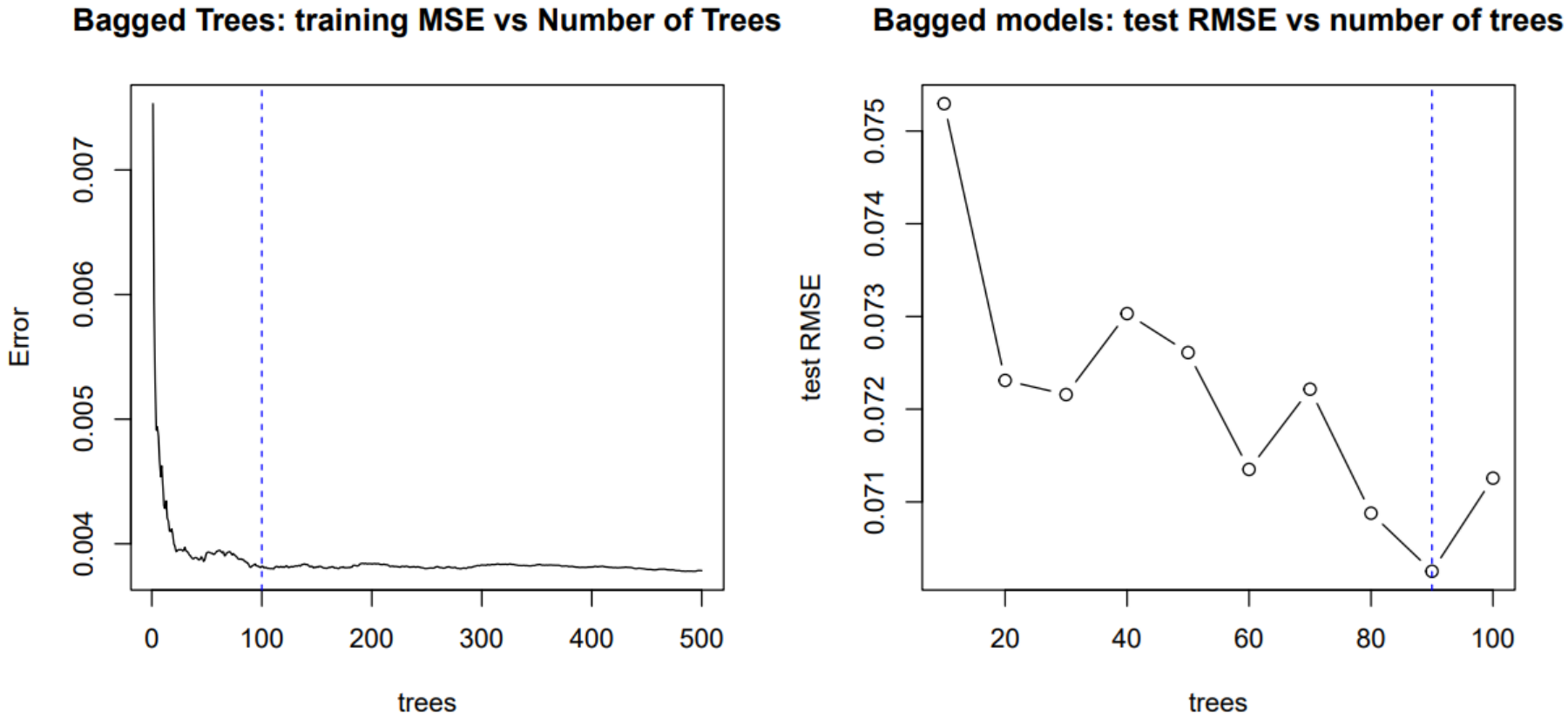


Figure 9: **Left:** The behavior of a bagging model using a high number of trees compared to the training MSE. - **Right:** We fitted 10 different bagging models changing `ntree` parameter from 10 to 100 with a span of 10 and then calculating for each iteration the test RMSE.

Regression trees: Bagging

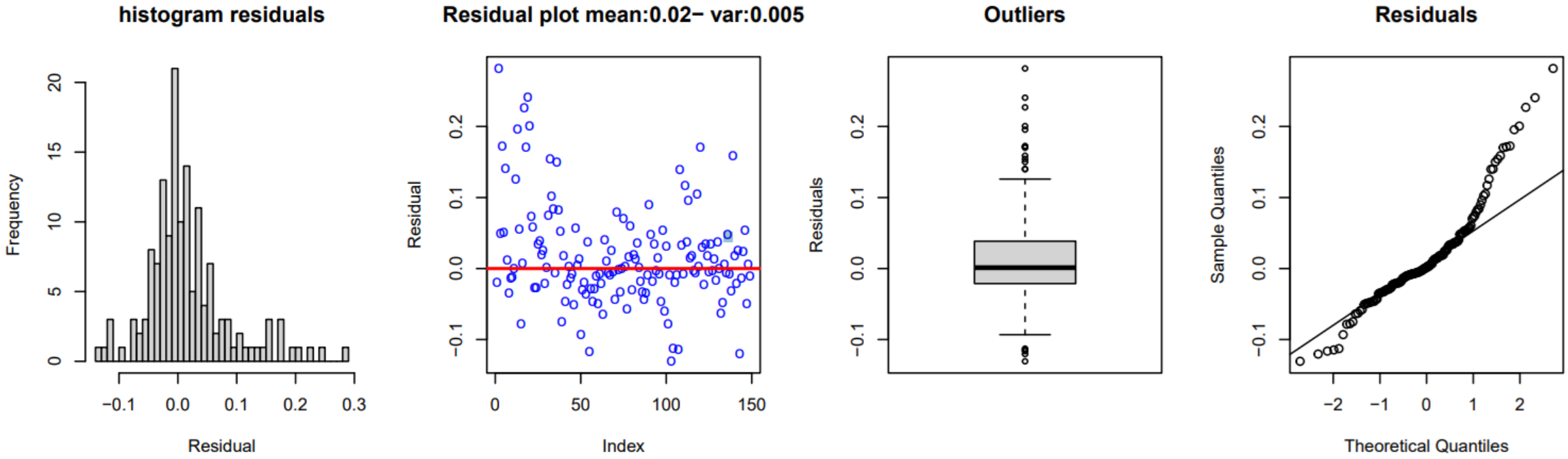


Figure 10: Residuals obtained in the validation/prediction phase from the optimized bagging model

Regression trees: Random forest

- Work in the same way as the bagging, but splits are constrained to a subset of the total regressors in the dataset
- Has lower learning rate compared to Bagging
- RMSE stabilizes after 100/150 trees
- Optimization: optimal number of trees
- Optimization: regressors used at each split

Regression trees: Random forest

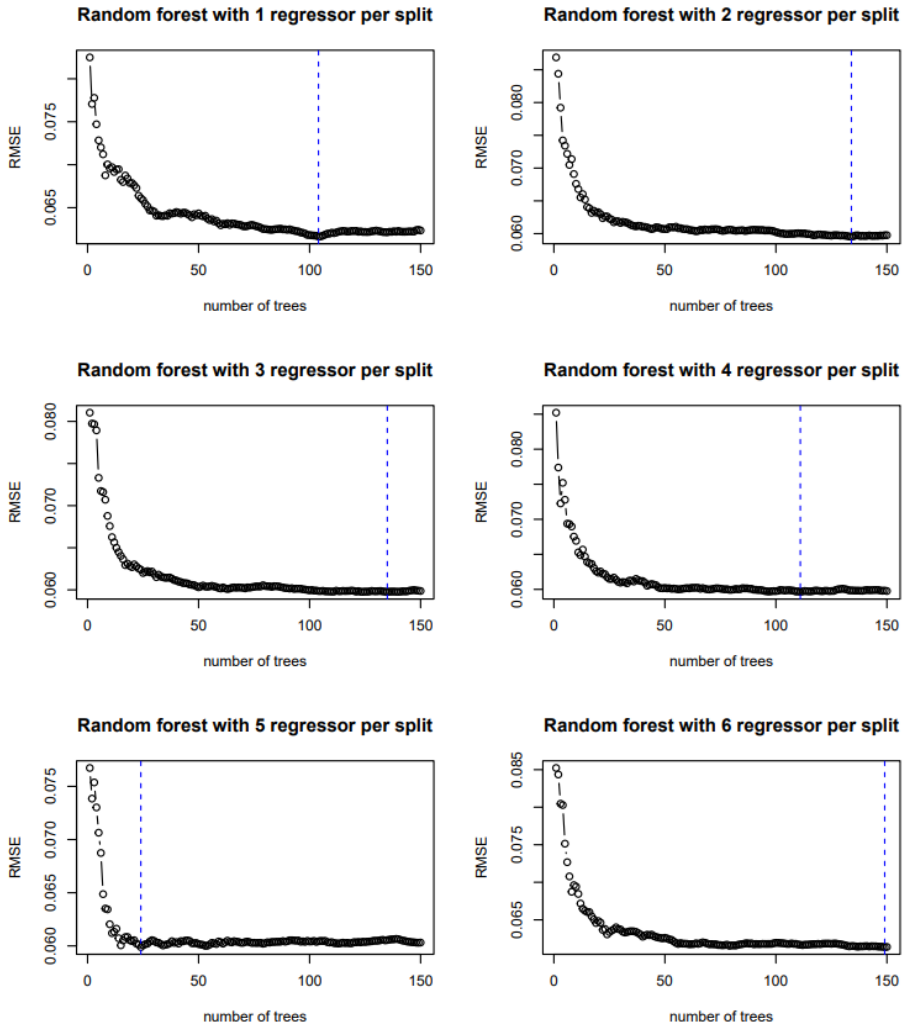


Table 12: Number of regressors used per each split in the random forest approach followed by the lowest RMSE (training and validation) and the number of tree associated

nRegressors per slipt	RMSE train	RMSE test	min nTrees
1	0.0616	0.0710	104
2	0.0596	0.0668	134
3	0.0598	0.0661	135
4	0.0596	0.0685	111
5	0.0599	0.0695	24
6	0.0614	0.0709	149

Regression trees: Boosting

- The most complex tree-based approach
- The learning rate, also called shrinkage parameter λ
- We fixed the shrinkage parameter to λ = 0.001
- The maximum number of trees which overfitting starts is 6000 trees
- Optimization: different numbers of depths

Regression trees: Boosting

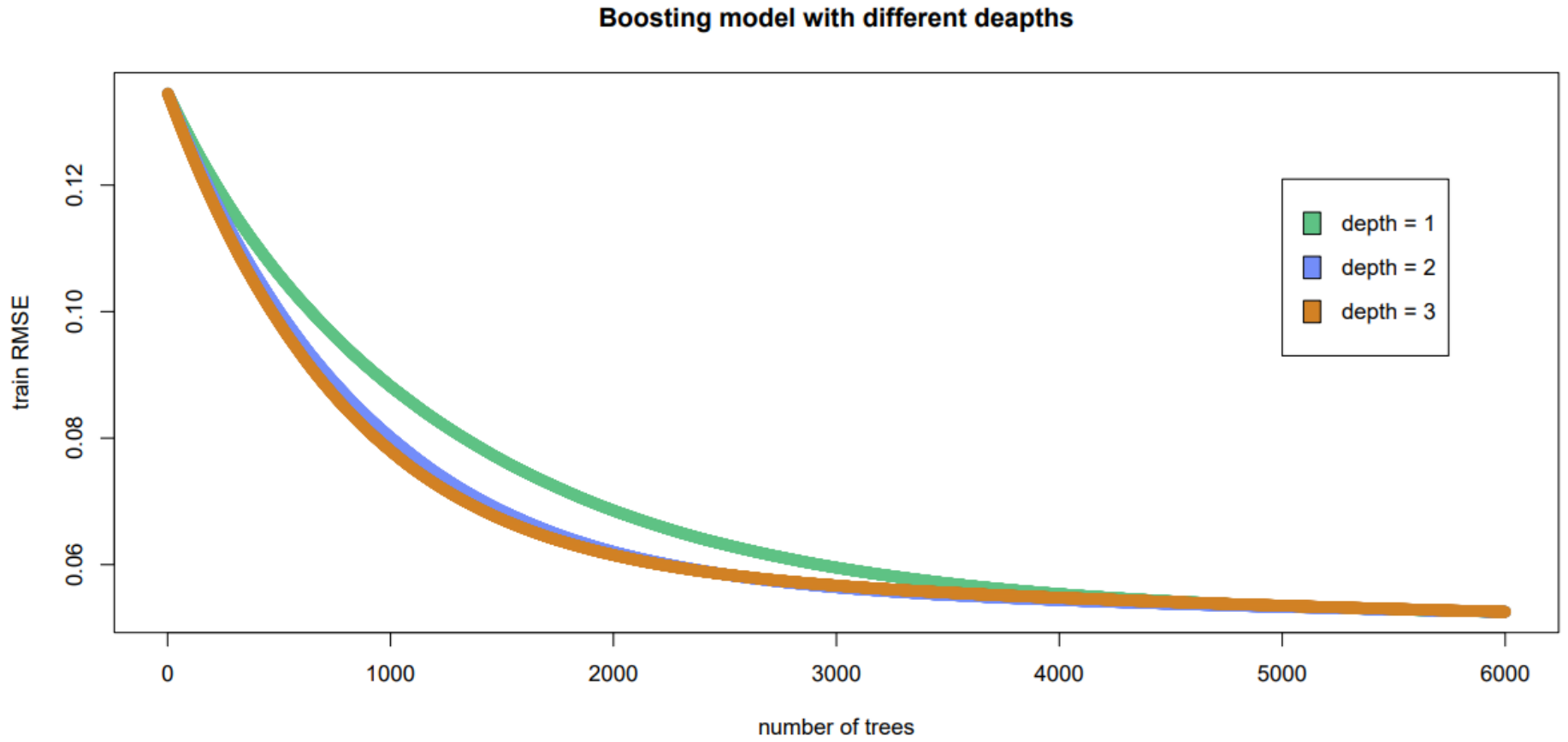


Figure 11: Training 3 boosting models with different depths but fixed number of trees

Regression trees: Boosting

Table 13: Boosting models with different depths using 6000 trees to train the models

Depth	RMSE test
1	0.06848
2	0.06859
3	0.06872

Final considerations about trees

- The best performing model in validation is the `forest_model` instead the worst one is `prune_model`
- The normality distribution of the `prune_model` is the best one
- `boost_model` which is the most complex model and the most time expensive is the worst in terms of residual behavior

Table 14: All the best models analyzed and their most important statistics after the optimization

Model	RMSE test	Shapiro-Wilk	nTrees	Split regressors
`prune_model`	0.082	pv = 8.130e-06	1	7
`bagg_model`	0.072	pv = 1.642e-07	90	7
`forest_model`	0.066	pv = 1.835e-08	135	3
`boost_model`	0.068	pv = 7.738e-09	6000	7

Final considerations about trees

Table 15: Percentage of importance of each regressor and the total percentage of variability of the model in explaining the dataset

Model	GRE	TOEFL	UniRatings	SOP	LOR	CGPA	Research	%var
`prune_model`	33.3	0.0	0.0	16.7	0.0	50.0	0.0	
`bagg_model`	10.2	5.1	3.0	3.9	1.3	36.6	2.2	78.7
`forest_model`	11.9	6.1	6.4	5.7	3.3	21.5	4.0	80.1

Table 16: Ordering the regressors per model starting from the most important one

Model	Importance or regressor
`prune_model`	CGPA > GRE > SOP
`bagg_model`	CGPA > GRE > TOEFL > SOP > UniRatings > Research > LOR
`forest_model`	CGPA > GRE > UniRatings > TOEFL > SOP > Research > LOR

Final considerations about trees

- Using ensemble methods we lose interpretability
- Trees are non robust methods
- The model that suits better our purposes is the `prune_model`

Conclusions

- The models that reflect our goals are `lm_model` and `prune_model`
- The residuals of both models act in the same way
- Probably there is/are missing variable/s inside the dataset
- In validation the `lm_model` performs better than the `prune_model`
- `lm_model` keeps being the model which better explains our dataset.