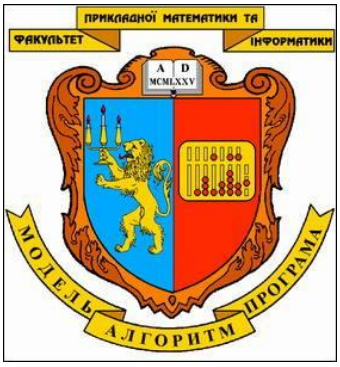
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Факультет прикладної математики та інформатики
Кафедра програмування



Індивідуальне завдання № 7

Дерева рішень

Виконала:
студентка групи ПМОм-11
Кравець Ольга

Львів 2025

Хід роботи

Варіант - 3

Визначила variant. Встановила set.seed та згенерувала redundant.

```
> variant=3
> variant
[1] 3
> set.seed(variant)
> redundant=floor(runif(1,5,25))
> redundant
[1] 8
```

Завдання 1.

Модифікувала дані, поділила на навчальну і тестову вибірки.

```
> set.seed(variant)
> test_indices = sample(1:nrow(Auto), round((redundant / 100) * nrow(Auto)))
> Auto_train = Auto[-test_indices, ]
> Auto_test = Auto[test_indices, ]
```

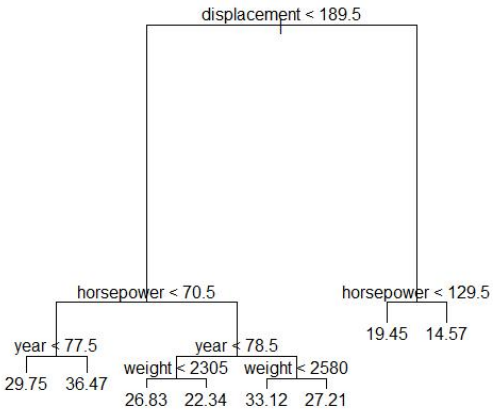
	row.names	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name	var11	var12	var13
1	1	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu			
2	2	15	8	350	165	3693	11.5	70	1	buick skylark 320			
3	3	18	8	318	150	3436	11	70	1	plymouth satellite			
4	4	16	8	304	150	3433	12	70	1	amc rebel sst			
5	5	17	8	302	140	3449	10.5	70	1	ford torino			
6	6	15	8	429	198	4341	10	70	1	ford galaxie 500			
7	7	14	8	454	220	4354	9	70	1	chevrolet impala			
8	8	14	8	440	215	4312	8.5	70	1	plymouth fury iii			

Побудувала дерево регресії.

```
> tree_model = tree(mpg ~ . - name, data = Auto_train)
> summary(tree_model)

Regression tree:
tree(formula = mpg ~ . - name, data = Auto_train)
Variables actually used in tree construction:
[1] "displacement" "horsepower"   "year"         "weight"
Number of terminal nodes:  8
Residual mean deviance:  9.255 = 3267 / 353
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-9.4170 -1.5650 -0.2514  0.0000  1.5490 18.5500

> plot(tree_model)
> text(tree_model, pretty = 0)
```



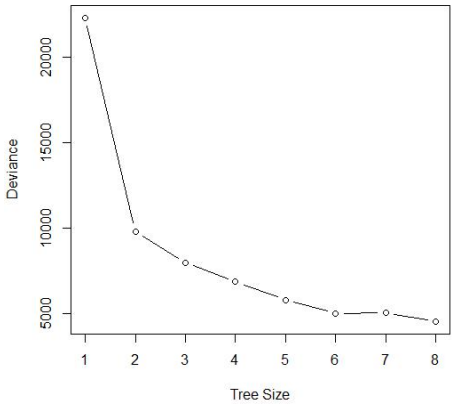
Основними предикторами mpg є displacement, horsepower, year і weight. Найекономніші - легкі й малопотужні машини. Великі й потужні авто мають найнижчий mpg.

Отримала прогноз на тестовому наборі та обчислила помилку.

```
> preds = predict(tree_model, newdata = Auto_test)
> mse = mean((Auto_test$mpg - preds)^2)
> mse
[1] 11.6477
```

Використала перехресну перевірку.

```
> cv_result = cv.tree(tree_model)
> plot(cv_result$size, cv_result$dev, type = "b", xlab = "Tree Size", ylab = "Deviance")
>
> best_size = cv_result$size[which.min(cv_result$dev)]
> pruned_tree = prune.tree(tree_model, best = best_size)
> plot(pruned_tree)
> text(pruned_tree, pretty = 0)
>
> pruned_preds = predict(pruned_tree, newdata = Auto_test)
> pruned_mse = mean((Auto_test$mpg - pruned_preds)^2)
> pruned_mse
[1] 11.6477
```

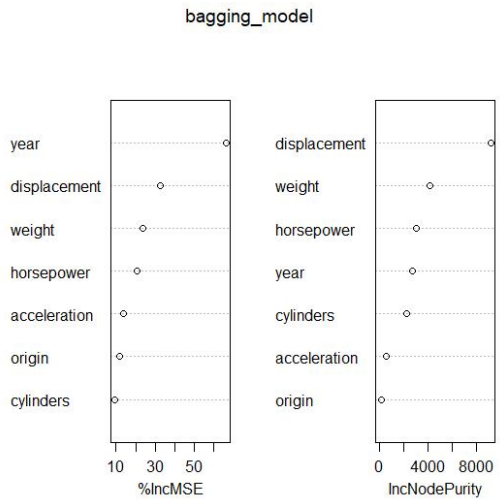


Чи покращує обрізання тестову помилку?

Помилка до обрізання і після не змінилася.

Використала бутстрап агрегацію для аналізу даних.

```
> library(randomForest)
> set.seed(variant)
> bagging_model = randomForest(mpg ~ . - name, data = Auto_train, mtry = ncol(Auto_train) - 2, importance = TRUE)
>
> bagging_preds = predict(bagging_model, newdata = Auto_test)
> bagging_mse = mean((Auto_test$mpg - bagging_preds)^2)
> bagging_mse
[1] 6.74399
> importance(bagging_model)
               %IncMSE IncNodePurity
cylinders     9.454542     2218.8806
displacement 32.406446     9220.1975
horsepower   20.792834     3013.8106
weight       23.525655     4162.4750
acceleration 13.680474      504.8431
year         66.243641     2711.6318
origin       11.812692      116.7190
> varImpPlot(bagging_model)
```



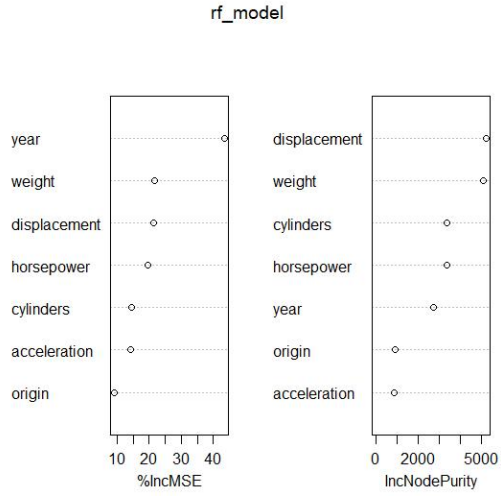
Яку тестову помилку отримано?

Помилка 6.74 і це краще, ніж дерево рішень, де 11.65.

Найбільший вплив на mpg має year - чим новіше авто, тим вищий mpg. Далі йдуть displacement, weight і horsepower.

Використала випадкові ліси для аналізу даних.

```
> varImpPlot(bagging_model)
> set.seed(variant)
> rf_model = randomForest(mpg ~ . - name, data = Auto_train, importance = TRUE)
> rf_preds = predict(rf_model, newdata = Auto_test)
> rf_mse = mean((Auto_test$mpg - rf_preds)^2)
> rf_mse
[1] 7.34664
> importance(rf_model)
               %IncMSE IncNodePurity
cylinders     14.47522     3378.3902
displacement  21.45088     5236.3920
horsepower    19.48159     3372.6856
weight        21.84484     5091.2069
acceleration  14.19882      861.7575
year          43.39828     2714.6314
origin         9.24772      900.8732
> varImpPlot(rf model)
```



Яку тестову помилку отримано?

Помилка 7.34 і це гірше, ніж при бутстрап, де 6.74, але все ще краще, ніж просте дерево 11.65.

Найбільший вплив на mpg має year. Далі йдуть weight, displacement і horsepower.

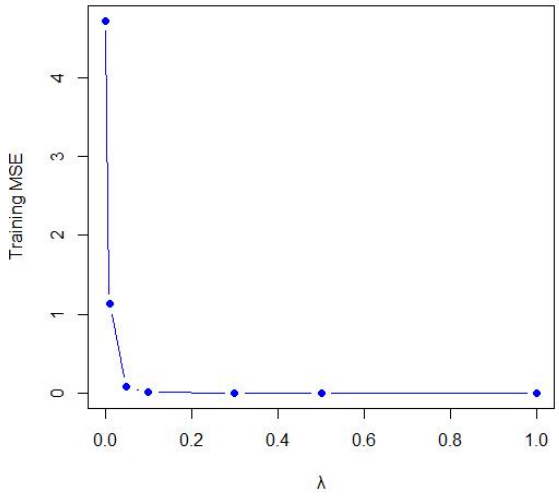
Вплив m на тестову помилку: зменшення m у випадковому лісі підвищило тестову помилку (7.35) порівняно з бутстрапом (m = p, помилка 6.74). У цьому випадку краще працює повний набір змінних.

Застосувала підсилення для навчальних даних для діапазону значень параметра стискання λ.

```
> library(gbm)
> set.seed(variant)
>
> lambdas = c(0.001, 0.01, 0.05, 0.1, 0.3, 0.5, 1)
> train_mse = numeric(length(lambdas))
> test_mse = numeric(length(lambdas))
> for (i in seq_along(lambdas)) {
+   boost_model = gbm(mpg ~ . - name, data = Auto_train, distribution = "gaussian", n.trees = 5000, interaction.depth = 4, shrinkage = lambdas[i], verbose = FALSE)
+
+   train_pred = predict(boost_model, newdata = Auto_train, n.trees = 5000)
+   test_pred = predict(boost_model, newdata = Auto_test, n.trees = 5000)
+
+   train_mse[i] = mean((Auto_train$mpg - train_pred)^2)
+   test_mse[i] = mean((Auto_test$mpg - test_pred)^2)
+ }
>
> results = data.frame(lambda = lambdas, Test_MSE = mse_values)
> print(results)
  lambda  Test_MSE
1  0.001  8.186263
2  0.010  8.260144
3  0.050  7.517827
4  0.100  6.111195
5  0.300  6.724209
6  0.500  9.541475
7  1.000 18.960005
```

Побудувала графік залежності навчального MSE від λ.

```
> plot(lambdas, train_mse, type = "b", pch = 19, col = "blue", xlab = "λ", ylab = "Training MSE")
```



Побудувала графік залежності тестового MSE від λ.

```
> plot(lambdas, test mse, type = "b", pch = 19, col = "red", xlab = "λ", ylab = "Test MSE")
```