Table of Content:

- Data
- How to measure importance of a feature
- Strategies Comparison
  - Strategies applied directly to the data
  - Strategies applied directly to the model
  - Strategies applied to the model's outputs
- Conclusion
- References

What is feature importance and feature selection importance

Feature importance is the process of determining the importance of individual features. Selecting which features to be included in a model can be a pivotal step in data science.

It is said **Values for computation cost** and are so-called for "engineering" some of the unimportant features and thus improve accuracy of the model.

It is said that **Values for cost associated with data collection** and are hard to interpret.

It is said **Improve interpretability** (the situation of "Black-box").

Data

We will use boston data set as an example through out this report.

Boston housing data set contains 506 rows and 14 columns. Target variable is the median value of owner-occupied homes in 1000's dollars.

- CRIM: per capita crime rate by town
- ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS: proportion of non-retail business acres per town.
- CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX: nitric oxides concentration (parts per 10 million)
- RM: average number of rooms per dwelling
- AGE: proportion of owner-occupied units built prior to 1940
- DIS: weighted distances to five Boston employment centres
- RAD: index of accessibility to radial highways
- TAX: full-value property-tax rate per $10,000
- PTRATIO: pupil-teacher ratio by town
- B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT: lower status of the population
- MEDV: Median value of owner-occupied homes in $1000's

```
library(MASS)
data(Boston)
```

How to measure importance of a feature

There are many different ways to measure the importance of a feature and this report covers the most popular methods. Although the importance of a feature depends on the model and the data, the importance of a feature is not to be confused with how well the model performs with that feature.

Category 1: strategies applied directly to the data

- Method 1: Univariate Feature Importance (Univariate)
- Method 2: Pearson Correlation Feature Selection (PCFS)
- Method 3: Minimal-redundancy-maximal relevance (mRMR)

Category 2: strategies applied directly to the model

- Method 4: Lasso Regression Feature Selection (Lasso)
- Method 5: Ridge Regression Feature Selection (Ridge)
- Method 6: Elastic Net Feature Selection (Elastic Net)

Category 3: strategies applied to the model's outputs

- Method 7: Boruta Feature Selection (Boruta)

Univariate Feature Selection

```
lapply(Boston, function(x) cor(x, Boston$medv))
```

```
$crim
[1] -0.3883046
$zn
[1] 0.3604453
$indus
[1] -0.4837252
$chas
[1] 0.1752602
$nox
[1] -0.4273208
$rm
[1] 0.6953599
$age
[1] -0.3769546
$dis
[1] 0.2499287
$rad
[1] -0.3819465
$tax
[1] -0.4685359
$ptratio
[1] -0.5077867
$black
[1] 0.3334608
$lstat
[1] -0.7376627
$medv
[1] 1
$cmedv
[1] -0.7377503
attr(,"class")
[1] "data.frame"
attr(,"rowsum")
[1] 506
attr(,"nrow")
[1] 506
attr(,"ncol")
[1] 1
```

The graph indicates that Lstat has the strongest correlation with the target variable (MEDV). LSTAT has the highest score which means this feature has the strongest predictive power with the target variable out of all other features.

Category 2: Pearson Correlation Feature Selection (PCFS)

Pearson correlation is a measure of the linear correlation between two variables. It's also known as a parametric test because it relies on data that conforms to certain assumptions (in the original variables).

The conclusion of Pearson's correlation test is based on the following null hypothesis: the correlation between two variables is zero (uncorrelated variables), which is strong by proving the original data (not the reduced PCA space), using degrees of the correlation coefficient distribution.

Here, the first correlation matrix the most useful correlation between variables and their correlation with the target variable. We will refer back to this correlation matrix.

```
pcfs_score_matrix(Boston, method = pearson, cmedv = TRUE)
```

```
pearson_score
```

```
$method
[1] "pearson"
$data
[1] "Boston"
$target
[1] "medv"
$cmedv
[1] -0.7377503
$cor_matrix
[1] -0.3883046 0.3604453 -0.4837252 0.1752602 -0.4273208 0.6953599 -0.3769546
[8] 0.2499287 -0.3819465 -0.4685359 -0.5077867 0.3334608 -0.7376627 1.0000000
[15] -0.7377503
attr(,"class")
[1] "data.frame"
attr(,"rowsum")
[1] 506
attr(,"nrow")
[1] 506
attr(,"ncol")
[1] 1
```

The graph indicates that Lstat has the strongest correlation with the target variable (MEDV). LSTAT has the highest score which means this feature has the strongest predictive power with the target variable out of all other features.

Category 3: mRMR (Minimal-redundancy-maximal relevance) (mRMR)

Feature selection based on mutual information: a fast filter method for supervised learning. It is intended to be a fast alternative to wrapper methods based on mutual information. Specifically, the criterion of MRMR, through iterations, is to find a score that measures a feature's performance (to derive top-ranked combinations of the original variables/features).

The conclusion of mRMR test is based on looking at the p-value. The null hypothesis is that the score is equal to "feature's importance". In other words, certain features (specifically at "important" ones) at high (multo)collinearity, but, in fact, can be dropped out of the model without affecting the score. In other words, it is similar to the test in Pearson by calculating the relationship between dependent variable and independent variables.

```
library(mRMRe)
data.subset <- mRMR.data(data = Boston)
feature.count <- ncol(Boston) - 1
mRMR <- mRMR.ensemble(data.subset, target = 14, feature.count, solution = 'OTHER')
```

```
Ensemble 1 of 1.
Starting 1 of 1.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Selection
 1  13  12  10   6   5   9   8   7   3   4   2  11  14
Estimates of binary mutual information:
 1 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472
solutioncount_1
 1 14
solutionscores_1
 1 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472
```

```
solution(mRMR, 1)[[1]][solution(mRMR, 1)[[2]] > 0]
```

```
 1 13 12 10  6  5  9  8  7  3  4  2 11 14
 1 14
```

Option 1: Pearson Correlation Coefficient

The Spearman correlation coefficient (similar to Pearson correlation) can take values from +1 to -1 (taking perfect negative monotonic relationship into account). Spearman's correlation coefficient is a measure of the monotonic association between two variables (monotonic means that, as one variable increases, the other variable tends to increase). The value of MRMR, through iterations, is to find a score that measures a feature's performance (to derive top-ranked combinations of the original variables/features).

Spearman's correlation coefficient and Spearman's ranking is the only test that "checks" if a set the same way as "rank's features". In other words, certain features specifically at "important" ones at high (multi)collinearity, but, in fact, can be dropped out of the model without affecting the score. In other words, it is similar to the test in Pearson by calculating the relationship between dependent variable and independent variables.

```
library(Hmisc)
rcorr(as.matrix(Boston))
```

```
      crim   zn indus chas  nox   rm  age  dis  rad  tax ptratio black
crim  1.00 0.41  0.56 0.07 0.76 0.20 0.68 0.69 0.62 0.61    0.29  0.38
zn    0.41 1.00  0.77 0.09 0.73 0.36 0.57 0.66 0.31 0.39    0.39  0.18
indus 0.56 0.77  1.00 0.10 0.91 0.39 0.72 0.71 0.40 0.72    0.38  0.35
chas  0.07 0.09  0.10 1.00 0.09 0.12 0.10 0.10 0.04 0.07    0.12  0.05
nox   0.76 0.73  0.91 0.09 1.00 0.22 0.83 0.77 0.61 0.67    0.19  0.43
rm    0.20 0.36  0.39 0.12 0.22 1.00 0.31 0.21 0.21 0.29    0.36  0.13
age   0.68 0.57  0.72 0.10 0.83 0.31 1.00 0.75 0.46 0.51    0.26  0.34
dis   0.69 0.66  0.71 0.10 0.77 0.21 0.75 1.00 0.49 0.53    0.23  0.29
rad   0.62 0.31  0.40 0.04 0.61 0.21 0.46 0.49 1.00 0.91    0.46  0.29
tax   0.61 0.39  0.72 0.07 0.67 0.29 0.51 0.53 0.91 1.00    0.46  0.44
ptratio 0.29 0.39  0.38 0.12 0.19 0.36 0.26 0.23 0.46 0.46    1.00  0.17
black   0.38 0.18  0.35 0.05 0.43 0.13 0.34 0.29 0.29 0.44    0.17  1.00
lstat   0.59 0.42  0.61 0.05 0.59 0.61 0.60 0.49 0.49 0.54    0.37  0.37
medv    0.39 0.36  0.48 0.18 0.43 0.69 0.38 0.25 0.38 0.47    0.51  0.33
      lstat medv
crim   0.59 0.39
zn     0.42 0.36
indus  0.61 0.48
chas   0.05 0.18
nox    0.59 0.43
rm     0.61 0.69
age    0.60 0.38
dis    0.49 0.25
rad    0.49 0.38
tax    0.54 0.47
ptratio 0.37 0.51
black   0.37 0.33
lstat   1.00 0.74
medv    0.74 1.00

n
      crim  zn indus chas  nox  rm  age  dis  rad  tax ptratio black
crim   506 506   506  506  506 506  506  506  506  506   506  506
zn     506 506   506  506  506 506  506  506  506  506   506  506
indus  506 506   506  506  506 506  506  506  506  506   506  506
chas   506 506   506  506  506 506  506  506  506  506   506  506
nox    506 506   506  506  506 506  506  506  506  506   506  506
rm     506 506   506  506  506 506  506  506  506  506   506  506
age    506 506   506  506  506 506  506  506  506  506   506  506
dis    506 506   506  506  506 506  506  506  506  506   506  506
rad    506 506   506  506  506 506  506  506  506  506   506  506
tax    506 506   506  506  506 506  506  506  506  506   506  506
ptratio 506 506   506  506  506 506  506  506  506  506   506  506
black   506 506   506  506  506 506  506  506  506  506   506  506
lstat   506 506   506  506  506 506  506  506  506  506   506  506
medv    506 506   506  506  506 506  506  506  506  506   506  506
      lstat medv
crim    506 506
zn      506 506
indus   506 506
chas    506 506
nox     506 506
rm      506 506
age     506 506
dis     506 506
rad     506 506
tax     506 506
ptratio 506 506
black   506 506
lstat   506 506
medv    506 506

P
      crim   zn   indus  chas   nox    rm     age    dis    rad    tax    ptratio black
crim        0.0000 0.0000 0.1120 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
zn    0.0000       0.0000 0.0420 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
indus 0.0000 0.0000       0.0250 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
chas  0.1120 0.0420 0.0250       0.0420 0.0070 0.0250 0.0250 0.3800 0.1120 0.0070  0.2500
nox   0.0000 0.0000 0.0000 0.0420       0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
rm    0.0000 0.0000 0.0000 0.0070 0.0000       0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
age   0.0000 0.0000 0.0000 0.0250 0.0000 0.0000       0.0000 0.0000 0.0000 0.0000  0.0000
dis   0.0000 0.0000 0.0000 0.0250 0.0000 0.0000 0.0000       0.0000 0.0000 0.0000  0.0000
rad   0.0000 0.0000 0.0000 0.3800 0.0000 0.0000 0.0000 0.0000       0.0000 0.0000  0.0000
tax   0.0000 0.0000 0.0000 0.1120 0.0000 0.0000 0.0000 0.0000 0.0000       0.0000  0.0000
ptratio 0.0000 0.0000 0.0000 0.0070 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000       0.0000
black   0.0000 0.0000 0.0000 0.2500 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
lstat   0.0000 0.0000 0.0000 0.2500 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
medv    0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  0.0000
      lstat  medv
crim  0.0000 0.0000
zn    0.0000 0.0000
indus 0.0000 0.0000
chas  0.2500 0.0000
nox   0.0000 0.0000
rm    0.0000 0.0000
age   0.0000 0.0000
dis   0.0000 0.0000
rad   0.0000 0.0000
tax   0.0000 0.0000
ptratio 0.0000 0.0000
black   0.0000 0.0000
lstat        0.0000
medv    0.0000
```

```
cor(Boston, Boston$medv, method = "spearman")
```

```
            [,1]
crim   -0.3954159
zn      0.4089123
indus  -0.5465930
chas   0.1796536
nox    -0.5398988
rm      0.6626731
age    -0.4570671
dis     0.2615338
rad    -0.3915798
tax    -0.5314893
ptratio -0.5507866
black   0.3046672
lstat  -0.7189561
medv    1.0000000
```

Option 2: Minimal-redundancy-maximal-relevance (mRMR)

Correlation coefficients measure the strength of association between two variables. Pearson's correlation coefficient (PCC) is a measure of the linear correlation between two variables. Its value lies between +1 and -1, inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation.

Pearson's correlation coefficient and Spearman's ranking is the only test that "checks" if a set the same way as "rank's features". In other words, certain features specifically at "important" ones at high (multi)collinearity, but, in fact, can be dropped out of the model without affecting the score. In other words, it is similar to the test in Pearson by calculating the relationship between dependent variable and independent variables.

```
lm(medv ~ lstat, data = Boston)
```

```
Call:
lm(formula = medv ~ lstat, data = Boston)
Coefficients:
(Intercept)        lstat
      34.55        -0.95
```

```
lm(medv ~ lstat + crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black, data = Boston)
```

```
Call:
lm(formula = medv ~ lstat + crim + zn + indus + chas + nox +
    rm + age + dis + rad + tax + ptratio + black, data = Boston)
Coefficients:
(Intercept)        lstat         crim           zn        indus         chas          nox           rm          age          dis          rad          tax      ptratio        black
   36.3411      -0.5226      -0.1080       0.0464       0.0206       2.6786     -17.7666       3.8099       0.0007      -1.4756       0.3060      -0.0123      -0.9527       0.0093
```

Option 3: Pearson Correlation Coefficient

Pearson's correlation coefficient (PCC) is a measure of the linear correlation between two variables. Its value lies between +1 and -1, inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. It is widely used in the sciences. It was also known as the "product-moment coefficient".

```
library(MASS)
data(Boston)
lm_boston = lm(medv ~ ., data = Boston)
step_lm_boston = stepAIC(lm_boston, direction = "both")
summary(step_lm_boston)
```

```
Start:  AIC=1590.83
medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
    tax + ptratio + black + lstat
Df Sum of Sq    RSS    AIC
<none>                11079 1590.8
- age     1    0.06  11079 1588.8
- indus   1    2.31  11081 1589.0
- chas    1    7.283  11086 1589.5
- tax     1    8.565  11088 1589.6
- crim    1   31.305  11110 1591.9
- zn      1   33.616  11112 1592.1
- black   1   34.811  11114 1592.2
- rad     1   79.373  11159 1596.7
- nox     1  168.592  11248 1605.6
- ptratio 1  170.351  11250 1605.8
- dis     1  188.018  11267 1607.5
- rm      1  707.796  11787 1658.5
- lstat   1  848.070  11927 1671.7
Call:
lm(formula = medv ~ crim + zn + indus + chas + nox + rm + age +
    dis + rad + tax + ptratio + black + lstat, data = Boston)
Residuals:
    Min      1Q  Median      3Q     Max
-15.5984  -2.7386  -0.5046   1.8773  26.4848
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.341145   5.067743   7.171  3.28e-12 ***
crim         -0.108011   0.032779  -3.295  0.00105 **
zn            0.046420   0.013879   3.344  0.00090 ***
indus         0.020559   0.061772   0.333  0.73912
chas          2.678562   0.859884   3.116  0.00196 **
nox         -17.766611   3.804594  -4.669  3.78e-06 ***
rm            3.809865   0.418084   9.112  < 2e-16 ***
age           0.000692   0.013161   0.053  0.95809
dis          -1.475567   0.198456  -7.434  4.62e-13 ***
rad           0.306049   0.063402   4.827  1.87e-06 ***
tax          -0.012329   0.003755  -3.283  0.00110 **
ptratio      -0.952747   0.130827  -7.282  1.31e-12 ***
black         0.009312   0.002686   3.467  0.00057 ***
lstat        -0.524758   0.047832 -10.971  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared,  0.7406,    Adjusted R-squared:  0.7348
F-statistic: 127.8 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
library(Boruta)
set.seed(111)
boruta.train <- Boruta(medv~., data = Boruta.Boston, doTrace = 2)
print(boruta.train)
```

```
Boruta performing  rough fixing =
mRMR.data(data = Boruta.Boston)
Ensemble 1 of 1.
Starting 1 of 1.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Dropping 0 variables.
Selection
 1  13  12  10   6   5   9   8   7   3   4   2  11  14
Estimates of binary mutual information:
 1 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472
solutionscores_1
 1 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472
solutionscores_1
 1 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472
```