Lecture 09: Machine learning 3

Meelis Kull

meelis.kull@ut.ee

Autumn 2017
Previous Lecture 08: Machine learning 2

✓ Example: Decision tree on MNIST
✓ Random forest
✓ Example: DT, RF, KNN not very good
✓ Basic linear classifier
✓ Support vector machine
✓ Underfitting and overfitting
✓ Parameter tuning
✓ Cross-validation
✓ Machine learning pipeline
✓ Learning on imbalanced data
Lecture 09: Machine learning 3

- Trade-off between true positives and false positives
- Scoring classifiers for TP/FP trade-off
- ROC curves
- Regression
- Simple linear regression
- Multi-variate linear regression
- Regularization
- Fitting non-linear regression curves
Lecture 09: Machine learning 3

- Trade-off between true positives and false positives
- Scoring classifiers for TP/FP trade-off
- ROC curves
- Regression
- Simple linear regression
- Multi-variate linear regression
- Regularization
- Fitting non-linear regression curves
Trade-off between true positives and false positives
Examples of imbalanced tasks (from previous lecture)

- Internet search:
  - Few relevant pages (positive class)
  - Many irrelevant pages (negative class)

- Medical diagnostic testing
  - Few disease cases (positive class)
  - Many healthy cases (negative class)
Imbalanced costs

• In imbalanced tasks the costs are also often imbalanced
  – False positives and false negatives can have very different costs

• We want to have many true positives, without having many false positives

• Let us define two more evaluation measures:
  – True positive rate (TPR)
  – False positive rate (FPR)
### Evaluation measures (from previous lecture)

<table>
<thead>
<tr>
<th>Actual = Yes</th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
<th>Positives (Pos)</th>
<th>Negatives (Neg)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positives (TP)</td>
<td><strong>True positives (TP)</strong></td>
<td>False negatives (FN) (Type II error)</td>
<td><strong>False negatives (FN)</strong></td>
<td><strong>True negatives (TN)</strong></td>
<td><strong>Total</strong></td>
</tr>
<tr>
<td>Actual = No</td>
<td>False positives (FP) (Type I error)</td>
<td>True negatives (TN)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predicted positives (PPos)</td>
<td>Predicted negatives (PNeg)</td>
<td>Positives (Pos)</td>
<td>Negatives (Neg)</td>
<td>Total</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ 10</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 30</td>
<td>50</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>60</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

- **Accuracy** = \((TP + TN)/Total\)
- **Precision** = \(TP/PPos\)
- **Recall** = \(TP/Pos\)

\[
\begin{align*}
\text{Accuracy} & = (10 + 50)/100 = 0.60 \\
\text{Precision} & = 10/40 = 0.25 \\
\text{Recall} & = 10/20 = 0.50
\end{align*}
\]
Evaluate measures of (from previous lecture)

<table>
<thead>
<tr>
<th></th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
<th>Positives (Pos)</th>
<th>Negatives (Neg)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual = Yes</td>
<td>True positives (TP)</td>
<td>False negatives (FN)</td>
<td>+</td>
<td>-</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>(Type II error)</td>
<td>(Type I error)</td>
<td>10</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Actual = No</td>
<td>False positives (FP)</td>
<td>True negatives (TN)</td>
<td>-</td>
<td>+</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>(Type I error)</td>
<td></td>
<td>10</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Predicted positives (PPos)</td>
<td></td>
<td>Predicted negatives (PNeg)</td>
<td></td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

**Accuracy** = \( \frac{TP + TN}{Total} \)

**Precision** = \( \frac{TP}{PPos} \)

**TPR** = **Recall** = \( \frac{TP}{Pos} \)

**FPR** = \( \frac{FP}{Neg} \)

**Example**

\[
\begin{align*}
\text{Accuracy} &= \frac{10 + 50}{100} = 0.60 \\
\text{Precision} &= \frac{10}{40} = 0.25 \\
\text{TPR} &= \text{Recall} = \frac{10}{20} = 0.50 \\
\text{FPR} &= \frac{30}{80} = 0.37
\end{align*}
\]
Trading off TPR and FPR

• A classifier that outputs binary label hits a particular balance between TPR and FPR
• This cannot be changed without learning a new classifier
• A better solution: ask classifiers to output scores:
  – Higher score means more likely positive
  – Lower score means more likely negative
• By choosing the decision threshold we can change trade-off between TPR and FPR
Lecture 09: Machine learning 3

✓ Trade-off between true positives and false positives

• Scoring classifiers for TP/FP trade-off

• ROC curves

• Regression

• Simple linear regression

• Multi-variate linear regression

• Regularization

• Fitting non-linear regression curves
Scoring classifiers for TP/FP trade-off
Scoring classifiers

• Most classification models can output scores in addition to labels

• KNN:
  – Score = proportion of positive neighbours

• SVM:
  – Score = signed distance to the decision boundary

• DT:
  – Score = proportion of positive instances in the decision leaf

• RF:
  – Score = proportion of trees predicting positive
Example

• Suppose:
  – We are given 5 instances
  – We have a classifier which outputs scores 0.6, 0.2, 0.7, 0.5, 0.4 on these instances
  – The true labels of these instances are 1, 0, 1, 0, 1 (where 1 is positive and 0 negative)
True labels

(1, 0, 1, 0, 1)

Classifier predicts

(0.6, 0.2, 0.7, 0.5, 0.4)

Adapted from slides by Dmytro Fishman
True labels

\[(1, 0, 1, 0, 1)\]

Let us sort the instances by the score descendingly

Classifier predicts

\[(0.6, 0.2, 0.7, 0.5, 0.4)\]

Adapted from slides by Dmytro Fishman
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

Adapted from slides by Dmytro Fishman
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

\[TPR = \frac{TP}{P}\]

Adapted from slides by Dmytro Fishman
True labels

\[(1, 1, 0, 1, 0)\]

TPR = \(\frac{TP}{P}\)
FPR = \(\frac{FP}{N}\)

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

Adapted from slides by Dmytro Fishman
True labels

\( (1, 1, 0, 1, 0) \)

\( TPR = \frac{TP}{P} \)

\( FPR = \frac{FP}{FP + TN} \)

\( (0.7, 0.6, 0.5, 0.4, 0.2) \)

Adapted from slides by Dmytro Fishman
True labels

\((1, 1, 0, 1, 0)\)

\[
\text{TPR} = \frac{TP}{P} \\
\text{FPR} = \frac{FP}{(FP + TN)}
\]

\((0.7, 0.6, 0.5, 0.4, 0.2)\)

Adapted from slides by Dmytro Fishman
True labels

(1, 1, 0, 1, 0)

TPR = TP/P
FPR = FP/(FP + TN)

(0.7, 0.6, 0.5, 0.4, 0.2)

We would like to evaluate different strictness levels of our classifier

Adapted from slides by Dmytro Fishman
True labels: $(1, 1, 0, 1, 0)$

$TPR = \frac{TP}{P}$

$FPR = \frac{FP}{(FP + TN)}$

What if consider as positive (1) only instances that were predicted positive with >= 0.7 probability?

Adapted from slides by Dmytro Fishman
True labels

\[(1, 1, 0, 1, 0)\]

\[\text{TPR} = \frac{TP}{P}\]
\[\text{FPR} = \frac{FP}{(FP + TN)}\]

What if consider as positive \((1)\) only instances that were predicted positive with \(\geq 0.7\) probability?

Adapted from slides by Dmytro Fishman
True labels

(1, 1, 0, 1, 0)

TPR = TP/P
FPR = FP/(FP + TN)

(0.7, 0.6, 0.5, 0.4, 0.2)

What if consider as positive (1) only instances that were predicted positive with $\geq 0.7$ probability?

What would TPR and FPR be in this case?

Adapted from slides by Dmytro Fishman
True labels

\[(1, 1, 0, 1, 0)\]

\[
\text{TPR} = \frac{TP}{P} \\
\text{FPR} = \frac{FP}{(FP + TN)}
\]

What if consider as positive \((1)\) only instances that were predicted positive with \(\geq 0.7\) probability?

\((0.7, 0.6, 0.5, 0.4, 0.2)\)

What would TPR and FPR be in this case?

\(\geq 0.7 \quad \text{TPR} = ? \quad \text{FPR} = ?\)

Adapted from slides by Dmytro Fishman
True labels

(1, 1, 0, 1, 0)

TPR = TP/P
FPR = FP/(FP + TN)

(0.7, 0.6, 0.5, 0.4, 0.2)

What if consider as positive (1) only instances that were predicted positive with >= 0.7 probability?

What would TPR and FPR be in this case?

>= 0.7
TPR = 1/3

FPR = 0/(0 + 2)

Adapted from slides by Dmytro Fishman
True labels

(1, 1, 0, 1, 0)

TPR = TP/P
FPR = FP/(FP + TN)

>= 0.7  TPR = 1/3  FPR = 0

(0.7, 0.6, 0.5, 0.4, 0.2)

Adapted from slides by Dmytro Fishman
True labels

$$\begin{pmatrix} 1, 1, 0, 1, 0 \end{pmatrix}$$

$$\text{TPR} = \frac{TP}{P}$$
$$\text{FPR} = \frac{FP}{FP + TN}$$

$$\geq 0.7 \quad \text{TPR} = \frac{1}{3} \quad \text{FPR} = 0$$

Let's plot this point on a graph

Adapted from slides by Dmytro Fishman
TPR = TP/P
FPR = FP/(FP + TN)

We shall do this procedure for all possible thresholds
True labels

(1, 1, 0, 1, 0)

TPR = TP/P
FPR = FP/(FP + TN)

>= 0.7  TPR = 1/3  FPR = 0

(0.7, 0.6, 0.5, 0.4, 0.2)

>= 0.6  TPR = ?  FPR = ?

Adapted from slides by Dmytro Fishman
True labels

\((1, 1, 0, 1, 0)\)

\(\text{TPR} = \frac{TP}{P}\)

\(\text{FPR} = \frac{FP}{FP + TN}\)

\(\geq 0.7\) \(\text{TPR} = \frac{1}{3}\) \(\text{FPR} = 0\)

\(\geq 0.6\) \(\text{TPR} = \frac{2}{3}\) \(\text{FPR} = 0\)

Adapted from slides by Dmytro Fishman
True labels

(1, 1, 0, 1, 0)

TPR = TP/P
FPR = FP/(FP + TN)

>= 0.7 TPR = 1/3 FPR = 0

>= 0.6 TPR = 2/3 FPR = 0

>= 0.5 TPR = ? FPR = ?

Adapted from slides by Dmytro Fishman
True labels

\[(1, 1, 0, 1, 0)\]

\[
\begin{align*}
TPR &= TP/P \\
FPR &= FP/(FP + TN)
\end{align*}
\]

- \(\geq 0.7\)  \(TPR = 1/3\)  \(FPR = 0\)
- \(\geq 0.6\)  \(TPR = 2/3\)  \(FPR = 0\)
- \(\geq 0.5\)  \(TPR = ?\)  \(FPR = ?\)

Oops, this is a false positive!

Adapted from slides by Dmytro Fishman
True labels

(1, 1, 0, 1, 0)

TPR = \frac{TP}{P}
FPR = \frac{FP}{(FP + TN)}

(0.7, 0.6, 0.5, 0.4, 0.2)

>= 0.7 \quad TPR = \frac{1}{3} \quad FPR = 0

>= 0.6 \quad TPR = \frac{2}{3} \quad FPR = 0

>= 0.5 \quad TPR = \frac{2}{3} \quad FPR = \frac{1}{2}

Adapted from slides by Dmytro Fishman
True labels

(1, 1, 0, 1, 0)

TPR = \frac{TP}{P}
FPR = \frac{FP}{FP + TN}

\begin{align*}
&>= 0.7 \quad \text{TPR} = \frac{1}{3} \quad \text{FPR} = 0 \\
&>= 0.6 \quad \text{TPR} = \frac{2}{3} \quad \text{FPR} = 0 \\
&>= 0.5 \quad \text{TPR} = \frac{2}{3} \quad \text{FPR} = \frac{1}{2}
\end{align*}

And so on…

Adapted from slides by Dmytro Fishman
True labels

\((1, 1, 0, 1, 0)\)

\(\text{TPR} = \frac{TP}{P}\)
\(\text{FPR} = \frac{FP}{(FP + TN)}\)

\(\geq 0.7\) \(\text{TPR} = 1/3\) \(\text{FPR} = 0\)
\(\geq 0.6\) \(\text{TPR} = 2/3\) \(\text{FPR} = 0\)
\(\geq 0.5\) \(\text{TPR} = 2/3\) \(\text{FPR} = 1/2\)
\(\geq 0.4\) \(\text{TPR} = 3/3\) \(\text{FPR} = 1/2\)
\(\geq 0.2\) \(\text{TPR} = 3/3\) \(\text{FPR} = 2/2\)

Adapted from slides by Dmytro Fishman
AUC = 0.83

True labels

\((1, 1, 0, 1, 0)\)

\((0.7, 0.6, 0.5, 0.4, 0.2)\)

\(\text{TPR} = \frac{TP}{P}\)

\(\text{FPR} = \frac{FP}{FP + TN}\)

\(\geq 0.7 \quad \text{TPR} = \frac{1}{3} \quad \text{FPR} = 0\)

\(\geq 0.6 \quad \text{TPR} = \frac{2}{3} \quad \text{FPR} = 0\)

\(\geq 0.5 \quad \text{TPR} = \frac{2}{3} \quad \text{FPR} = \frac{1}{2}\)

\(\geq 0.4 \quad \text{TPR} = \frac{3}{3} \quad \text{FPR} = \frac{1}{2}\)

\(\geq 0.2 \quad \text{TPR} = \frac{3}{3} \quad \text{FPR} = \frac{2}{2}\)

\(\text{AUC} = \text{area under curve}\)

\(\text{Adapted from slides by Dmytro Fishman}\)
True labels

\[(1, 1, 0, 1, 0)\]

AUC of 0.5 means random guess

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

AUC = 0.5

Adapted from slides by Dmytro Fishman
True labels

\((1, 1, 0, 1, 0)\)

\((0.7, 0.6, 0.5, 0.4, 0.2)\)

AUC of 0.5 means random guess

AUC of 1 means perfect classification

Adapted from slides by Dmytro Fishman
True labels

\[(1, 1, 0, 1, 0)\]

AUC of 0.5 means random guess

AUC of 1 means perfect classification, overfitting

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

Adapted from slides by Dmytro Fishman
Lecture 09: Machine learning 3

✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
  • ROC curves
  • Regression
  • Simple linear regression
  • Multi-variate linear regression
  • Regularization
  • Fitting non-linear regression curves
ROC curves
ROC = Receiver Operating Characteristic
Let us look at another way of drawing ROC curves
True labels

(1, 1, 0, 1, 0)

(0.7, 0.6, 0.5, 0.4, 0.2)

There are as many marks on y-axis as there are 1’s in our true labels.
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

There are as many marks on x-axis as there are 0’s in our true labels.
True labels

\((1, 1, 0, 1, 0)\)

\((0.7, 0.6, 0.5, 0.4, 0.2)\)

Go through true labels one by one, if 1 go up, if 0 go right
Go through true labels one by one, if 1 go up, if 0 go right.

True labels

(1, 1, 0, 1, 0)

(0.7, 0.6, 0.5, 0.4, 0.2)
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

Go through true labels one by one, if 1 go up, if 0 go right
Go through true labels one by one, if 1 go up, if 0 go right
Go through true labels one by one, if 1 go up, if 0 go right.
Go through true labels one by one, if 1 go up, if 0 go right
True labels

(1, 1, 0, 1, 0)

(0.7, 0.6, 0.5, 0.4, 0.2)

Go through true labels one by one, if 1 go up, if 0 go right
Go through true labels one by one, if 1 go up, if 0 go right.
True labels

\[(1, 1, 0, 1, 0)\]

Go through true labels one by one, if 1 go up, if 0 go right

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]
True labels

(1, 1, 0, 1, 0)

(0.7, 0.6, 0.5, 0.4, 0.2)

Go through true labels one by one, if 1 go up, if 0 go right
Go through true labels one by one, if 1 go up, if 0 go right.
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

This is called Receiver Operating Characteristic (ROC)
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

This is square has sides of length 1 and 1
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

AUC = area under the (ROC) curve
True labels

\[(1, 1, 0, 1, 0)\]

\[(0.7, 0.6, 0.5, 0.4, 0.2)\]

AUC = 0.83

AUC = area under the (ROC) curve
…can be made better than random by inverting its predictions

A random classifier ($p=0.5$)

A worse than random classifier…

Adapted from slides by Peter Flach
A typical comparison of classifiers with ROC curves

- NetChop C-term 3.0
- TAP + ProteaSMM-i
- ProteaSMM-i
Lecture 09: Machine learning 3

✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ ROC curves

• **Regression**
• Simple linear regression
• Multi-variate linear regression
• Regularization
• Fitting non-linear regression curves
Regression
## Classification on Lenses dataset

<table>
<thead>
<tr>
<th>Presbyopic</th>
<th>Young</th>
<th>Spectacle prescription</th>
<th>Astigmatic</th>
<th>Tear production rate</th>
<th>Can use contact lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>Myope</td>
<td>No</td>
<td>Normal</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Myope</td>
<td>Yes</td>
<td>Reduced</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Reduced</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Normal</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Myope</td>
<td>No</td>
<td>Reduced</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Myope</td>
<td>No</td>
<td>Normal</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Myope</td>
<td>Yes</td>
<td>Normal</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Reduced</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Reduced</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Normal</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Myope</td>
<td>No</td>
<td>Normal</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Myope</td>
<td>Yes</td>
<td>Normal</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Myope</td>
<td>No</td>
<td>Reduced</td>
<td>???</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Myope</td>
<td>Yes</td>
<td>Normal</td>
<td>???</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Reduced</td>
<td>???</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Normal</td>
<td>???</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Myope</td>
<td>Yes</td>
<td>Reduced</td>
<td>???</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Normal</td>
<td>???</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Myope</td>
<td>No</td>
<td>Reduced</td>
<td>???</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Myope</td>
<td>Yes</td>
<td>Normal</td>
<td>???</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Reduced</td>
<td>???</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Normal</td>
<td>???</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Reduced</td>
<td>???</td>
</tr>
</tbody>
</table>
### Classification on Lenses dataset

<table>
<thead>
<tr>
<th>Features</th>
<th>Categorical Label</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>Myope</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Myope</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hypermetrope</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hypermetrope</td>
</tr>
</tbody>
</table>

#### Training data

<table>
<thead>
<tr>
<th>Features</th>
<th>Categorical Label</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>Hypermetrope</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Hypermetrope</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Myope</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
</tr>
</tbody>
</table>

#### Test data

<table>
<thead>
<tr>
<th>Features</th>
<th>Categorical Label</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>Myope</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hypermetrope</td>
</tr>
</tbody>
</table>
Classification

Features

Categorical Label

Training data

Test data

(labels known)

(must predict)
## Features

<table>
<thead>
<tr>
<th>Training data</th>
<th>Test data</th>
</tr>
</thead>
<tbody>
<tr>
<td>(labels known)</td>
<td>(must predict)</td>
</tr>
</tbody>
</table>
Regression

Features

Numeric Label

Training data

(labels known)

Test data

(must predict)
```r
> data(mtcars)
> mtcars

<table>
<thead>
<tr>
<th></th>
<th>mpg</th>
<th>cyl</th>
<th>disp</th>
<th>hp</th>
<th>drat</th>
<th>wt</th>
<th>qsec</th>
<th>vs</th>
<th>am</th>
<th>gear</th>
<th>carb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mazda RX4</td>
<td>21.0</td>
<td>6</td>
<td>160.0</td>
<td>110</td>
<td>3.90</td>
<td>2.620</td>
<td>16.46</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Mazda RX4 Wag</td>
<td>21.0</td>
<td>6</td>
<td>160.0</td>
<td>110</td>
<td>3.90</td>
<td>2.875</td>
<td>17.02</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Datsun 710</td>
<td>22.8</td>
<td>4</td>
<td>108.0</td>
<td>93</td>
<td>3.85</td>
<td>2.320</td>
<td>18.61</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Hornet 4 Drive</td>
<td>21.4</td>
<td>6</td>
<td>258.0</td>
<td>110</td>
<td>3.08</td>
<td>3.215</td>
<td>19.44</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Hornet Sportabout</td>
<td>18.7</td>
<td>8</td>
<td>360.0</td>
<td>175</td>
<td>3.15</td>
<td>3.440</td>
<td>17.02</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Valiant</td>
<td>18.1</td>
<td>6</td>
<td>225.0</td>
<td>105</td>
<td>2.76</td>
<td>3.460</td>
<td>20.22</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Duster 360</td>
<td>14.3</td>
<td>8</td>
<td>360.0</td>
<td>245</td>
<td>3.21</td>
<td>3.570</td>
<td>15.84</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Merc 240D</td>
<td>24.4</td>
<td>4</td>
<td>146.7</td>
<td>62</td>
<td>3.69</td>
<td>3.190</td>
<td>20.00</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Merc 230</td>
<td>22.8</td>
<td>4</td>
<td>140.8</td>
<td>95</td>
<td>3.92</td>
<td>3.150</td>
<td>22.90</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Merc 280</td>
<td>19.2</td>
<td>6</td>
<td>167.6</td>
<td>123</td>
<td>3.92</td>
<td>3.440</td>
<td>18.30</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Merc 280C</td>
<td>17.8</td>
<td>6</td>
<td>167.6</td>
<td>123</td>
<td>3.92</td>
<td>3.440</td>
<td>18.90</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Merc 450SE</td>
<td>16.4</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>4.070</td>
<td>17.40</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Merc 450SL</td>
<td>17.3</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>3.730</td>
<td>17.60</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Merc 450SLC</td>
<td>15.2</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>3.780</td>
<td>18.00</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Cadillac Fleetwood</td>
<td>10.4</td>
<td>8</td>
<td>472.0</td>
<td>205</td>
<td>2.93</td>
<td>5.250</td>
<td>17.98</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Lincoln Continental</td>
<td>10.4</td>
<td>8</td>
<td>460.0</td>
<td>215</td>
<td>3.00</td>
<td>5.424</td>
<td>17.82</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Chrysler Imperial</td>
<td>14.7</td>
<td>8</td>
<td>440.0</td>
<td>230</td>
<td>3.23</td>
<td>5.345</td>
<td>17.42</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Fiat 128</td>
<td>32.4</td>
<td>4</td>
<td>78.7</td>
<td>66</td>
<td>4.08</td>
<td>2.200</td>
<td>19.47</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Honda Civic</td>
<td>30.4</td>
<td>4</td>
<td>75.7</td>
<td>52</td>
<td>4.93</td>
<td>1.615</td>
<td>18.52</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Toyota Corolla</td>
<td>33.9</td>
<td>4</td>
<td>71.1</td>
<td>65</td>
<td>4.22</td>
<td>1.835</td>
<td>19.90</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Toyota Corona</td>
<td>21.5</td>
<td>4</td>
<td>120.1</td>
<td>97</td>
<td>3.70</td>
<td>2.465</td>
<td>20.01</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Dodge Challenger</td>
<td>15.5</td>
<td>8</td>
<td>318.0</td>
<td>150</td>
<td>2.76</td>
<td>3.520</td>
<td>16.87</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>AMC Javelin</td>
<td>15.2</td>
<td>8</td>
<td>304.0</td>
<td>150</td>
<td>3.15</td>
<td>3.435</td>
<td>17.30</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Camaro Z28</td>
<td>13.3</td>
<td>8</td>
<td>350.0</td>
<td>245</td>
<td>3.73</td>
<td>3.840</td>
<td>15.41</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Pontiac Firebird</td>
<td>19.2</td>
<td>8</td>
<td>400.0</td>
<td>175</td>
<td>3.08</td>
<td>3.845</td>
<td>17.05</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Fiat X1-9</td>
<td>27.3</td>
<td>4</td>
<td>79.0</td>
<td>66</td>
<td>4.08</td>
<td>1.935</td>
<td>18.90</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Porsche 914-2</td>
<td>26.0</td>
<td>4</td>
<td>120.3</td>
<td>91</td>
<td>4.43</td>
<td>2.140</td>
<td>16.70</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Lotus Europa</td>
<td>30.4</td>
<td>4</td>
<td>95.1</td>
<td>113</td>
<td>3.77</td>
<td>1.513</td>
<td>16.90</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Ford Pantera L</td>
<td>15.8</td>
<td>8</td>
<td>351.0</td>
<td>264</td>
<td>4.22</td>
<td>3.170</td>
<td>14.50</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Ferrari Dino</td>
<td>19.7</td>
<td>6</td>
<td>145.0</td>
<td>175</td>
<td>3.62</td>
<td>2.770</td>
<td>15.50</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Maserati Bora</td>
<td>15.0</td>
<td>8</td>
<td>301.0</td>
<td>335</td>
<td>3.54</td>
<td>3.570</td>
<td>14.60</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Volvo 142E</td>
<td>21.4</td>
<td>4</td>
<td>121.0</td>
<td>109</td>
<td>4.11</td>
<td>2.780</td>
<td>18.60</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
```
Example: Mtcars dataset

```
data(mtcars)
> mtcars
  mpg  cyl  disp  hp  drat  wt  qsec  vs  am  gear  carb
Mazda RX4         21.0   6  160.0  110  3.90  2.620  16.46    0  1    4    4
Mazda RX4 Wag     21.0   6  160.0  110  3.90  2.875  17.02    0  1    4    4
Datsun 710        22.8   4  108.0   93  3.85  2.320  18.61    1  1    4    1
Hornet 4 Drive    21.4   6  258.0  110  3.08  3.215  19.44    1  0    3    1
Hornet Sportabout 18.7   8  360.0  175  3.15  3.440  17.02    0  0    3    2
Valiant           18.1   6  225.0  105  2.76  3.460  20.22    1  0    3    1
Duster 360        14.3   8  360.0  245  3.21  3.570  15.84    0  0    3    4
Merc 240D         22.8   4 110.0   95  3.15  2.770  18.61    0  0    3    2
Merc 230           22.8   4 110.0   95  3.15  2.770  18.61    0  0    3    2
Merc 280           19.2   6 160.0  110  3.90  2.770  18.61    0  0    3    2
Merc 450SE         20.2   8 220.0  168  3.90  3.440  18.61    0  0    3    2
Merc 450SL         17.3   8 220.0  168  3.90  3.440  18.61    0  0    3    2
Merc 450SLC        15.2   8 220.0  168  3.90  3.440  18.61    0  0    3    2
Cadillac Fleetwood 10.4   8 472.0  205  2.93  5.250  17.98    0  0    3    4
Lincoln Continental 10.4   8 460.0  215  3.00  5.424  17.82    0  0    3    4
Chrysler Imperial  14.7   8 440.0  230  3.23  5.345  17.42    0  0    3    4
Fiat 128           32.4   4    78.7  66   4.08  2.200  19.47    1  1    4    1
Honda Civic        30.4   4    75.7  52   4.93  1.615  18.52    1  1    4    2
Toyota Corolla     33.9   4    71.1  65   4.22  1.835  19.90    1  1    4    1
Toyota Corona      21.5   4 120.1  97   3.70  2.465  20.01    1  0    3    1
Dodge Challenger   15.5   8 318.0  150  2.76  3.520  16.87    0  0    3    2
AMC Javelin        15.2   8 304.0  150  3.15  3.435  17.30    0  0    3    2
Camaro Z28         13.3   8 350.0  245  3.73  3.840  15.41    0  0    3    4
Pontiac Firebird   19.2   8 400.0  175  3.08  3.845  17.05    0  0    3    2
Lotus Europa       30.4   4  95.1  113  3.77  1.513  16.90    1  1    5    2
Ford Pantera L     15.8   8 351.0  264  4.22  3.170  14.50    0  1    5    4
Ferrari Dino       19.7   6 145.0  175  3.62  2.770  15.50    0  1    5    6
Maserati Bora      15.0   8 301.0 335  3.54  3.570  14.60    0  1    5    8
Volvo 142E         21.4   4 121.0 109  4.11  2.780  18.60    1  1    4    2
```

- **mpg**: Miles/(US) gallon
- **cyl**: Number of cylinders
- **disp**: Displacement (cu.in.)
- **hp**: Gross horsepower
- **drat**: Rear axle ratio
- **wt**: Weight (1000 lbs)
- **qsec**: 1/4 mile time
- **vs**: V/S (0 = V-engine, 1 = straight engine)
- **am**: Transmission (0 = automatic, 1 = manual)
- **gear**: Number of forward gears
- **carb**: Number of carburetors
Example: Mtcars dataset

```r
> data(mtcars)
> mtcars

<table>
<thead>
<tr>
<th>mpg</th>
<th>cyl</th>
<th>disp</th>
<th>hp</th>
<th>drat</th>
<th>wt</th>
<th>qsec</th>
<th>vs</th>
<th>am</th>
<th>gear</th>
<th>carb</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.0</td>
<td>6</td>
<td>150.0</td>
<td>110</td>
<td>3.90</td>
<td>2.620</td>
<td>16.46</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>21.0</td>
<td>6</td>
<td>160.0</td>
<td>170</td>
<td>3.80</td>
<td>2.875</td>
<td>17.08</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>22.8</td>
<td>4</td>
<td>108.0</td>
<td>93</td>
<td>3.80</td>
<td>2.320</td>
<td>18.61</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>21.4</td>
<td>6</td>
<td>258.0</td>
<td>110</td>
<td>3.08</td>
<td>3.845</td>
<td>19.90</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>18.7</td>
<td>8</td>
<td>360.0</td>
<td>175</td>
<td>3.15</td>
<td>3.770</td>
<td>15.42</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>18.1</td>
<td>6</td>
<td>225.0</td>
<td>105</td>
<td>2.76</td>
<td>3.440</td>
<td>17.02</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>14.3</td>
<td>8</td>
<td>360.0</td>
<td>245</td>
<td>3.21</td>
<td>3.570</td>
<td>15.81</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>15.0</td>
<td>8</td>
<td>360.0</td>
<td>245</td>
<td>3.21</td>
<td>3.570</td>
<td>15.81</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>14.7</td>
<td>8</td>
<td>440.0</td>
<td>220</td>
<td>3.00</td>
<td>3.520</td>
<td>16.46</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>14.1</td>
<td>8</td>
<td>460.0</td>
<td>215</td>
<td>3.00</td>
<td>3.520</td>
<td>16.46</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>11.4</td>
<td>8</td>
<td>472.0</td>
<td>205</td>
<td>2.93</td>
<td>3.460</td>
<td>16.63</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>15.4</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>3.920</td>
<td>16.11</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>15.2</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>3.920</td>
<td>16.11</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>13.3</td>
<td>6</td>
<td>167.6</td>
<td>123</td>
<td>3.92</td>
<td>3.440</td>
<td>17.98</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>13.3</td>
<td>6</td>
<td>167.6</td>
<td>123</td>
<td>3.92</td>
<td>3.440</td>
<td>17.98</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>14.4</td>
<td>4</td>
<td>140.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.4</td>
<td>4</td>
<td>146.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.0</td>
<td>8</td>
<td>440.0</td>
<td>220</td>
<td>3.00</td>
<td>3.520</td>
<td>16.46</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>12.0</td>
<td>8</td>
<td>460.0</td>
<td>215</td>
<td>3.00</td>
<td>3.520</td>
<td>16.46</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>12.0</td>
<td>8</td>
<td>472.0</td>
<td>205</td>
<td>2.93</td>
<td>3.460</td>
<td>16.11</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>95.1</td>
<td>95</td>
<td>4.43</td>
<td>2.140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79.0</td>
<td>66</td>
<td>4.08</td>
<td>1.935</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78.7</td>
<td>65</td>
<td>4.22</td>
<td>1.835</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>52</td>
<td>4.93</td>
<td>1.615</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>62</td>
<td>3.69</td>
<td>3.190</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>91</td>
<td>3.77</td>
<td>1.513</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Can we predict 'mpg' from other features?

- **mpg**: Miles/(US) gallon
- **cyl**: Number of cylinders
- **disp**: Displacement (cu.in.)
- **hp**: Gross horsepower
- **drat**: Rear axle ratio
- **wt**: Weight (1000 lbs)
- **qsec**: 1/4 mile time
- **vs**: V/S (0 = V-engine, 1 = straight engine)
- **am**: Transmission (0 = automatic, 1 = manual)
- **gear**: Number of forward gears
- **carb**: Number of carburetors
Training dataset (50% of all data)

<table>
<thead>
<tr>
<th>Model</th>
<th>mpg</th>
<th>cyl</th>
<th>disp</th>
<th>hp</th>
<th>drat</th>
<th>wt</th>
<th>qsec</th>
<th>vs</th>
<th>am</th>
<th>gear</th>
<th>carb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mazda RX4 Wag</td>
<td>21.0</td>
<td>6</td>
<td>160.0</td>
<td>110</td>
<td>3.90</td>
<td>2.875</td>
<td>17.02</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Hornet 4 Drive</td>
<td>21.4</td>
<td>6</td>
<td>258.0</td>
<td>110</td>
<td>3.08</td>
<td>3.215</td>
<td>19.44</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Hornet Sportabout</td>
<td>18.7</td>
<td>8</td>
<td>360.0</td>
<td>175</td>
<td>3.15</td>
<td>3.440</td>
<td>17.02</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Valiant</td>
<td>18.1</td>
<td>6</td>
<td>225.0</td>
<td>105</td>
<td>2.76</td>
<td>3.460</td>
<td>20.22</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Merc 240D</td>
<td>24.4</td>
<td>4</td>
<td>146.7</td>
<td>62</td>
<td>3.69</td>
<td>3.190</td>
<td>20.00</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Merc 230</td>
<td>22.8</td>
<td>4</td>
<td>140.8</td>
<td>95</td>
<td>3.92</td>
<td>3.150</td>
<td>22.90</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Merc 450SE</td>
<td>16.4</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>4.070</td>
<td>17.40</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Merc 450SLC</td>
<td>15.2</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>3.780</td>
<td>18.00</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Lincoln Continental</td>
<td>10.4</td>
<td>8</td>
<td>460.0</td>
<td>215</td>
<td>3.00</td>
<td>5.424</td>
<td>17.82</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Chrysler Imperial</td>
<td>14.7</td>
<td>8</td>
<td>440.0</td>
<td>230</td>
<td>3.23</td>
<td>5.345</td>
<td>17.42</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Fiat 128</td>
<td>32.4</td>
<td>4</td>
<td>78.7</td>
<td>66</td>
<td>4.08</td>
<td>2.200</td>
<td>19.47</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Toyota Corolla</td>
<td>33.9</td>
<td>4</td>
<td>71.1</td>
<td>65</td>
<td>4.22</td>
<td>1.835</td>
<td>19.90</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Pontiac Firebird</td>
<td>19.2</td>
<td>8</td>
<td>400.0</td>
<td>175</td>
<td>3.08</td>
<td>3.845</td>
<td>17.05</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Porsche 914-2</td>
<td>26.0</td>
<td>4</td>
<td>120.3</td>
<td>91</td>
<td>4.43</td>
<td>2.140</td>
<td>16.70</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Ford Pantera L</td>
<td>15.8</td>
<td>8</td>
<td>351.0</td>
<td>264</td>
<td>4.22</td>
<td>3.170</td>
<td>14.50</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Volvo 142E</td>
<td>21.4</td>
<td>4</td>
<td>121.0</td>
<td>109</td>
<td>4.11</td>
<td>2.780</td>
<td>18.60</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
Car with more horsepower (hp) can drive less miles per gallon (mpg)
Correlations of features and label

Car with more weight (wt) can drive less miles per gallon (mpg)
Correlations of features and label

Let us first try to predict ‘mpg’ from only ‘wt’ gallon (mpg)
How to predict ‘mpg’ from ‘wt’?

![Graph showing the relationship between weight (wt) and miles per gallon (mpg). The graph is a scatter plot with individual data points plotted on the x-axis for weight and on the y-axis for miles per gallon. The points form a downward trend, indicating a negative correlation.]
How to predict ‘mpg’ from ‘wt’?

Suppose we have a test instance with wt=4.5
What value for ‘mpg’ to predict?
How to predict ‘mpg’ from ‘wt’?

Suppose we have a test instance with wt=4.5
What value for ‘mpg’ to predict?
Trade-off between true positives and false positives

Scoring classifiers for TP/FP trade-off

ROC curves

Regression

- Simple linear regression
- Multi-variate linear regression
- Regularization
- Fitting non-linear regression curves
Simple linear regression
Simple linear regression

\[ mpg = \alpha + \beta \text{wt} + \epsilon \]

\[ y = \alpha + \beta x + \epsilon \]

output
(dependent variable, response)

input
(independent variable, feature, explanatory variable, etc)

Adapted from slides by Anna Leontjeva
Simple linear regression

\[ y = \alpha + \beta x + \epsilon \]

- **Intercept (bias)**
  - Mean of \( y \) when \( x = 0 \)

- **Coefficient (slope, or weight \( w \))**
  - Shows how output increases
  - If input increases by one unit

- **Noise (error term, residual)**
  - Shows what we are not able to predict with \( x \)

Adapted from slides by Anna Leontjeva
Simple linear regression
Simple linear regression

\[ \text{Intercept: } \alpha = 38.86 \]
Simple linear regression

Intercept: $\alpha = 38.86$

Slope: $\beta = -5.38$
Simple linear regression

\[ \text{Intercept: } \alpha = 38.86 \]

\[ \text{Slope: } \beta = -5.38 \]

\[ \text{Residual at this particular training instance: } \epsilon = -6.01 \]
Which regression line is best?
Which regression line is best?

We want to minimize residuals!
We want to minimize residuals!

In practice usually:
minimize sum of squares of residuals
(known as OLS = ordinary least squares)
Simple linear regression with ordinary least squares

We search for a function \( \hat{y} = f(x) \)

which minimizes mean squared error (MSE) :

\[
\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta x_i - \alpha)^2
\]

which means to find derivatives wrt \( \alpha \) and \( \beta \)

and solve the system of equations:

\[
\begin{align*}
\frac{\partial \text{MSE}}{\partial \alpha} &= 0 \\
\frac{\partial \text{MSE}}{\partial \beta} &= 0
\end{align*}
\]

Adapted from slides by Anna Leontjeva
Simple linear regression with ordinary least squares

\[
\begin{align*}
\frac{\partial \text{MSE}}{\partial \alpha} &= 0 \\
\frac{\partial \text{MSE}}{\partial \beta} &= 0
\end{align*}
\]

\[\Leftrightarrow\]

\[
\begin{align*}
\alpha &= \bar{y} - \beta \bar{x} \\
\beta &= \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N} (x_i - \bar{x})^2}
\end{align*}
\]

where

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

\[
\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i
\]

Adapted from slides by Anna Leontjeva
Linear model to predict ‘mpg’ from ‘wt’

> fit = lm(formula = mpg ~ wt, data = mtcars_train)
> summary(fit)

Call:
  lm(formula = mpg ~ wt, data = mtcars_train)

Residuals:
      Min       1Q   Median       3Q      Max
-6.01300 -2.21470 -0.37136  1.43709  5.36920

Coefficients:
            Estimate Std. Error t value  Pr(>|t|)
(Intercept)   38.8649     2.9878 13.0084 3.30e-09 ***
     wt         -5.3792     0.8521 -6.3132 1.91e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.299 on 14 degrees of freedom
Multiple R-squared:  0.74,     Adjusted R-squared:  0.7214
F-statistic: 39.85 on 1 and 14 DF,  p-value: 1.915e-05
Linear model to predict ‘mpg’ from ‘wt’

```r
> fit = lm(formula = mpg ~ wt, data = mtcars_train)
> summary(fit)

Call:  
  lm(formula = mpg ~ wt, data = mtcars_train)

Residuals: 
       Min       1Q    Median       3Q      Max 
-6.01300 -2.21472 -0.37134  1.43712  5.36919

Coefficients: 
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)    38.864904   2.987810  13.008 3.30e-09 ***
wt              -5.379199   0.852123  -6.313 1.91e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.299 on 14 degrees of freedom 
Multiple R-squared:  0.74,    Adjusted R-squared:  0.7214 
F-statistic: 39.85 on 1 and 14 DF,  p-value: 1.915e-05
```

Linear model with ordinary least squares method
Linear model to predict ‘mpg’ from ‘wt’

```
> fit = lm(formula = mpg ~ wt, data = mtcars_train)
> summary(fit)
```

Call:
`lm(formula = mpg ~ wt, data = mtcars_train)`

Residuals:
```
    Min     1Q Median     3Q    Max
-6.0130 -2.2147 -0.3713  1.4371  5.3692
```

Coefficients:
```
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  38.8649     2.9878 13.008 3.30e-09 ***
      wt         -5.3792     0.8521 -6.313 1.91e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error: 3.299 on 14 degrees of freedom
Multiple R-squared: 0.74,
Adjusted R-squared: 0.7214
F-statistic: 39.85 on 1 and 14 DF,  p-value: 1.915e-05

Predict ‘mpg’ from ’wt’
Linear model to predict ‘mpg’ from ‘wt’

```r
> fit = lm(formula = mpg ~ wt, data = mtcars_train)
> summary(fit)
```

Call:
`lm(formula = mpg ~ wt, data = mtcars_train)`

Residuals:
```
Min  1Q Median  3Q Max
-6.0130 -2.2147 -0.3713 1.4371 5.3692
```

Coefficients:
```
                    Estimate Std. Error t value Pr(>|t|)  
(Intercept)       38.8649    2.9878 13.008 3.30e-09 ***
wt                 -5.3792    0.8521 -6.313 1.91e-05 ***
---                  
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error: 3.299 on 14 degrees of freedom
Multiple R-squared: 0.74,  Adjusted R-squared: 0.7214
F-statistic: 39.85 on 1 and 14 DF,  p-value: 1.915e-05

**Fitted line (model):** $mpg = 38.8649 - 5.3792 \times wt + \epsilon$
Linear model to predict ‘mpg’ from ‘wt’

> fit = lm(formula = mpg ~ wt, data = mtcars_train)
> summary(fit)

Call:
  lm(formula = mpg ~ wt, data = mtcars_train)

Residuals:
   Min    1Q  Median    3Q   Max
-6.0130 -2.2147 -0.3713  1.4371  5.3692

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
(Intercept) 38.8649     2.9878 13.008 3.30e-09 ***
wt           -5.3792     0.8521  -6.313 1.91e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.299 on 14 degrees of freedom
Multiple R-squared:  0.74,   Adjusted R-squared:  0.7214
F-statistic: 39.85 on 1 and 14 DF,  p-value: 1.915e-05

Distribution of residuals of the fitted model
Linear model to predict ‘mpg’ from ‘wt’

> fit = lm(formula = mpg ~ wt, data = mtcars_train)
> summary(fit)

Call:
  lm(formula = mpg ~ wt, data = mtcars_train)

Residuals:
     Min      1Q  Median      3Q     Max
-6.0130 -2.2147 -0.3713  1.4371  5.3692

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  38.8649     2.9878   13.01  3.30e-09  ***
wt           -5.3792     0.8521    -6.31  1.91e-05  ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.299 on 14 degrees of freedom
Multiple R-squared:  0.74, Adjusted R-squared:  0.721

P-value of the hypothesis that the coefficient is non-zero
Training dataset (50% of all data)

<table>
<thead>
<tr>
<th>mpg</th>
<th>cyl</th>
<th>disp</th>
<th>hp</th>
<th>drat</th>
<th>wt</th>
<th>qsec</th>
<th>vs</th>
<th>am</th>
<th>gear</th>
<th>carb</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.0</td>
<td>6</td>
<td>160.0</td>
<td>110</td>
<td>3.90</td>
<td>2.875</td>
<td>17.02</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>21.4</td>
<td>6</td>
<td>258.0</td>
<td>110</td>
<td>3.08</td>
<td>3.215</td>
<td>19.44</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>18.7</td>
<td>8</td>
<td>360.0</td>
<td>175</td>
<td>3.15</td>
<td>3.440</td>
<td>17.02</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>18.1</td>
<td>6</td>
<td>225.0</td>
<td>105</td>
<td>2.76</td>
<td>3.460</td>
<td>20.22</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>24.4</td>
<td>4</td>
<td>146.7</td>
<td>62</td>
<td>3.69</td>
<td>3.190</td>
<td>20.00</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>22.8</td>
<td>4</td>
<td>140.8</td>
<td>95</td>
<td>3.92</td>
<td>3.150</td>
<td>22.90</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>16.4</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>4.470</td>
<td>17.40</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>15.2</td>
<td>8</td>
<td>275.8</td>
<td>180</td>
<td>3.07</td>
<td>3.780</td>
<td>18.00</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10.4</td>
<td>8</td>
<td>460.0</td>
<td>215</td>
<td>3.00</td>
<td>5.424</td>
<td>17.82</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>14.7</td>
<td>8</td>
<td>440.0</td>
<td>230</td>
<td>3.23</td>
<td>5.345</td>
<td>17.42</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>32.4</td>
<td>4</td>
<td>78.7</td>
<td>66</td>
<td>4.08</td>
<td>2.200</td>
<td>19.47</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>33.9</td>
<td>4</td>
<td>71.1</td>
<td>65</td>
<td>4.22</td>
<td>1.835</td>
<td>19.90</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

We have used feature ‘wt’ to predict label ‘mpg’. Can we make better predictions by using more features? For example,

\[
mpg = -127.77 + 10.29 \text{ cyl} - 0.019 \text{ disp} + \ldots - 6.13 \text{ carb}
\]
Lecture 09: Machine learning 3

- Trade-off between true positives and false positives
- Scoring classifiers for TP/FP trade-off
- ROC curves
- Regression
- Simple linear regression
  - Multi-variate linear regression
  - Regularization
  - Fitting non-linear regression curves
Multi-variate linear regression
Multivariate linear regression

all the same, but instead of one feature, \( x \) is a \( k \)-dimensional vector

\[
x_i = (x_{i1}, x_{i2}, \ldots, x_{ik})
\]

the model is the linear combination of all features:

\[
\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k
\]

via the matrix representation: \( \hat{y} = X\beta \)

\[
\begin{pmatrix}
\hat{y}_1 \\
\vdots \\
\hat{y}_n
\end{pmatrix} =
\begin{pmatrix}
1 & x_{11} & \cdots & x_{1k} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{n1} & \cdots & x_{nk}
\end{pmatrix}
\begin{pmatrix}
\beta_0 \\
\vdots \\
\beta_k
\end{pmatrix}
\]

Adapted from slides by Anna Leontjeva
Multivariate linear regression

Recall from a simple regression a system of equations:

\[
\begin{align*}
\frac{\partial \text{MSE}}{\partial \alpha} &= 0 \\
\frac{\partial \text{MSE}}{\partial \beta} &= 0
\end{align*}
\]

\[\iff\]

\[
\begin{align*}
\alpha &= \bar{y} - \beta \bar{x} \\
\beta &= \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}
\end{align*}
\]

For multivariate regression MSE is defined:

\[
\text{MSE} = \frac{1}{N}(y - \hat{y})^T(y - \hat{y})
\]

\[
\begin{align*}
\frac{\partial \text{MSE}}{\partial \beta} &= -2y^T X + 2X^T X \beta
\end{align*}
\]

Adapted from slides by Anna Leontjeva
Multivariate linear regression

\[
\frac{\partial \text{MSE}}{\partial \beta} = -2y^T X + 2X^T X \beta
\]

\[
\frac{\partial \text{MSE}}{\partial \beta} = 0 \Rightarrow \beta = (X^T X)^{-1} X^T y
\]

complexity of matrix inverse is high: \( O(n^{2.373}) \)

in practice iterative methods are used (e.g. gradient descent)

Adapted from slides by Anna Leontjeva
Linear model to predict ‘mpg’

```r
> fit = lm(formula = mpg ~ ., data = mtcars_train)
> summary(fit)
...

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  -127.77341  28.31390 -4.513 0.006325 **
cyl           10.29870   1.95085  5.279 0.003247 **
disp          -0.01960   0.01237 -1.585 0.173871 
hp            -0.22132   0.03701 -6.001 0.001874 **
drat           24.96600   3.50497  7.123 0.000846 ***
w t            9.63213   2.47932  3.885 0.011583 *
qsec           0.27590   0.51255  0.538 0.613464 
vs             3.83936   1.84467  2.081 0.091907 .
am            1.71842   1.97014  0.872 0.422979 
gear          1.88721   1.93827  0.974 0.374945 
carb         -6.13806   1.16106 -5.287 0.003227 **
```

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.188 on 5 degrees of freedom
Multiple R-squared:  0.988,     Adjusted R-squared:  0.9639
F-statistic: 41.06 on 10 and 5 DF,  p-value: 0.0003599
Linear model to predict ‘mpg’

```r
> fit = lm(formula = mpg ~ ., data = mtcars_train)
> summary(fit)

... Predict ‘mpg’ from all other features

Coefficients:
            Estimate Std. Error t value  Pr(>|t|)  
(Intercept)  -127.77341   28.31390  -4.513 0.006325 **
cy  10.29870     1.95085   5.279 0.003247 **
disp -0.01960     0.01237  -1.585 0.173871   
hp  -0.22132     0.03701  -5.980 0.001874 **
drat  24.96600     3.50497   7.123 0.000846 ***
wt   9.63213     2.47932   3.885 0.011583 *
qsec  0.27590     0.51255   0.538 0.613464   
vs   3.83936     1.84467   2.081 0.091907 .  
am   1.71842     1.97014   0.872 0.422979   
egear  1.88721     1.93827   0.974 0.374945   
carb -6.13806     1.16106  -5.287 0.003227 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.188 on 5 degrees of freedom
Multiple R-squared:  0.988,  Adjusted R-squared:  0.9639
F-statistic: 41.06 on 10 and 5 DF,  p-value: 0.0003599
```
Linear model to predict ‘mpg’

\[
\text{fit} = \text{lm(formula = mpg ~ ., data = mtcars_train)}
\]

\[
\text{summary(fit)}
\]

...  

Coefficients:  

| Estimate | Std. Error | t value | Pr(>|t|) |  
|-----------|-------------|---------|----------|
| (Intercept) | -127.77341 | 28.31390 | -4.513 | 0.006325 ** |
| cyl | 10.29870 | 1.95085 | 5.279 | 0.003247 ** |
| disp | -0.01960 | 0.01237 | -1.585 | 0.173871 |
| hp | -0.22132 | 0.03701 | -5.980 | 0.001874 ** |
| drat | 24.96600 | 3.50497 | 7.123 | 0.000846 *** |
| wt | 9.63213 | 2.47932 | 3.885 | 0.011583 * |
| qsec | 0.27590 | 0.51255 | 0.538 | 0.613464 |
| vs | 3.83936 | 1.84467 | 2.081 | 0.091907 . |
| am | 1.71842 | 1.97014 | 0.872 | 0.422979 |
| gear | 1.88721 | 1.93827 | 0.974 | 0.374945 |
| carb | -6.13806 | 1.16106 | -5.287 | 0.003227 ** |

---

Signif. codes:  

0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Fitted line (model):  

\[ \text{mpg} = -127.773 + 10.298 \text{ cyl} - 0.019 \text{ disp} + \cdots - 6.138 \text{ carb} \]
Linear model to predict ‘mpg’

```r
> fit = lm(formula = mpg ~ ., data = mtcars_train)
> summary(fit)

... 

The coefficient for ‘drat’ is statistically significantly different from zero with confidence level 0.001

|   | Estimate | Std. Error | t value | Pr(>|t|) |
|---|----------|------------|---------|----------|
| (Intercept) | -127.77341 | 28.31390 | -4.513 | 0.006325 ** |
| cyl | 10.29870 | 1.95085 | 5.279 | 0.003247 ** |
| disp | -0.01960 | 0.01237 | -1.585 | 0.173871 |
| hp | -0.22132 | 0.03701 | -5.980 | 0.001874 ** |
| drat | 24.96600 | 3.50497 | 7.123 | 0.000846 *** |
| wt | 9.63213 | 2.47932 | 3.885 | 0.011583 * |
| qsec | 0.27590 | 0.51255 | 0.538 | 0.613464 |
| vs | 3.83936 | 1.84467 | 2.081 | 0.091907 . |
| am | 1.71842 | 1.97014 | 0.872 | 0.422979 |
| gear | 1.88721 | 1.93827 | 0.974 | 0.374945 |
| carb | -6.13806 | 1.16106 | -5.287 | 0.003227 ** |

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.188 on 5 degrees of freedom
Multiple R-squared:  0.988,  Adjusted R-squared:  0.9639
F-statistic: 41.06 on 10 and 5 DF,  p-value: 0.0003599
## Linear model to predict `mpg`

```r
> fit = lm(formula = mpg ~ ., data = mtcars_train)
> summary(fit)

...  

Coefficients:  

|                | Estimate | Std. Error | t value | Pr(>|t|) |
|----------------|----------|------------|---------|----------|
| (Intercept)    | -127.77341 | 28.31390  | -4.513  | 0.006325 ** |
| cyl            | 10.29870  | 1.95085   | 5.279   | 0.003247 ** |
| disp           | -0.01960  | 0.01237   | -1.585  | 0.173871 |
| hp             | -0.22132  | 0.03701   | -5.980  | 0.001874 ** |
| drat           | 24.96600  | 3.50497   | 7.123   | 0.000846 *** |
| wt             | 9.63213   | 2.47932   | 3.885   | 0.011583 *  
| qsec           | 0.27590   | 0.51255   | 0.538   | 0.613464 |
| vs             | 3.83936   | 1.84467   | 2.081   | 0.091907 .
| am             | 1.71842   | 1.97014   | 0.872   | 0.422979 |
| gear           | 1.88721   | 1.93827   | 0.974   | 0.374945 |
| carb           | -6.13806  | 1.16106   | -5.287  | 0.003227 ** |

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.188 on 5 degrees of freedom  
Multiple R-squared: 0.988,    Adjusted R-squared:  0.9639  
F-statistic: 41.06 on 10 and 5 DF,  p-value: 0.0003599

*The coefficient for ‘am’ is NOT statistically significantly different from zero*
Linear model to predict ‘mpg’

```r
> fit = lm(formula = mpg ~ ., data = mtcars_train)
> summary(fit)

... 

Coefficients: 

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -127.77341 | 28.31390 | -4.513  0.006325 ** |
| cyl       | 10.29870   | 1.95085  | 5.279  0.003247 ** |
| disp      | -0.01960   | 0.01237  | -1.585 0.173871 |
| hp        | -0.22132   | 0.03701  | -5.980 0.001874 ** |
| drat      | 24.96600   | 3.50497  | 7.123  0.000846 *** |
| wt        | 9.63213    | 2.47932  | 3.885  0.011583 * |
| qsec      | 0.27590    | 0.51255  | 0.538  0.613464 |
| vs        | 3.83936    | 1.84467  | 2.081  0.091907 .|
| am        | 1.71842    | 1.97014  | 0.872  0.422979 |
| carb      | -6.13806   | 1.16106  | -5.287 0.003227 ** |

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.188 on 5 degrees of freedom
Multiple R-squared: 0.988, Adjusted R-squared: 0.9639
F-statistic: 41.06 on 10 and 5 DF, p-value: 0.0003599
```

The coefficient for ‘am’ is NOT statistically significantly different from zero

By this the regression learning algorithm reports that it is NOT confident that all features have an effect at all on the target variable ‘mpg’
Overfitting...

- **Constant model:**
  - \( mpg = 20.737 \)
  - trainMSE = 36.63859
  - testMSE = 34.57625

- **Simple linear model:**
  - \( mpg = 38.8649 - 5.3792 \cdot wt \)
  - trainMSE = 9.525724
  - testMSE = 12.18288

- **Multivariate linear regression model:**
  - \( mpg = -127.773 + 10.298 \cdot cyl + \cdots - 6.138 \cdot carb \)
  - trainMSE = 0.4408109
  - testMSE = 337.9995
Overfitting…

• **Constant model:**
  - $mpg = 20.737$
  - trainMSE = 36.63859
  - testMSE = 34.57625

• **Simple linear model:**
  - $mpg = 38.8649 - 5.3792 \cdot wt$
  - trainMSE = 9.525724
  - testMSE = 12.18288

• **Multivariate linear regression model:**
  - $mpg = -127.773 + 10.298 \cdot cyl + \cdots - 6.138 \cdot carb$
  - trainMSE = 0.4408109
  - testMSE = 337.9995

  **Overfitting badly… Too complex model!**
Why overfitting?

• We have 16 training instances
• We have 10 features, hence 11 parameters in the linear model
• Learning 11 parameters from 16 instances is very hard
• Noise level is too high to learn a good predictor
• Ordinary least squares method does not ‘notice’ this and starts fitting noise as well
Lecture 09: Machine learning 3

✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ ROC curves
✓ Regression
✓ Simple linear regression
✓ Multi-variate linear regression
• Regularization
• Fitting non-linear regression curves
Regularization
Solution: Regularization

• We want to achieve two goals:
  – Minimize error on training data
  – Minimize model complexity

• Regularization minimizes:
  \[ \text{Error} + \lambda \text{Complexity} \]

  where \( \lambda \) is the regularization parameter

• Higher \( \lambda \) favors simpler models,
  lower \( \lambda \) favors more accurate models
Regularization in linear regression

• Intuition:
  - Complexity = many weights with high absolute value

• Common complexity measures:
  - \( \| \beta \|_1 = \sum_{i=1}^{m} |\beta_i| \) (Lasso regression)
  - \( \| \beta \|_2^2 = \sum_{i=1}^{m} \beta_i^2 \) (Ridge regression)

• Lasso regression results in many coefficients being zero, thus performing feature selection

• Ridge regression tends to keep all coefficients but decreases them to smaller numbers
Regularization on Mtcars dataset

• Let us first try lasso regression and ridge regression with $\lambda = 1$
Lasso regression model to predict ‘mpg’

```r
> library(glmnet)
> fit = glmnet(as.matrix(mtcars_train[,-1]), mtcars_train[,1], alpha=1, lambda=1)
> coef(fit)
11 x 1 sparse Matrix of class "dgCMatrix"

           s0
(Intercept) 28.2891704
cyl .
disp .
hp -0.0295088
drat 1.3486882
wt -2.0854970
qsec .
vs .
am .
gear .
carb -0.5106284
```
library(glmnet)
fit = glmnet(as.matrix(mtcars_train[, -1]), mtcars_train[, 1], alpha=1, lambda=1)
coef(fit)
11 x 1 sparse Matrix of class "dgCMatrix"

s0
(Intercept)  28.2891704
cyl .
disp .
hp  -0.0295088
drat  1.3486882
wt  -2.0854970
qsec .
vs .
am .
gear .
carb  -0.5106284

**glmnet** is a command that can do lasso regression (alpha=1) and ridge regression (alpha=0)
Lasso regression model to predict ‘mpg’

```r
> library(glmnet)
> fit = glmnet(as.matrix(mtcars_train[,,-1]), mtcars_train[,1], alpha=1, lambda=1)
> coef(fit)
11 x 1 sparse Matrix of class "dgCMatrix"

s0
(Intercept) 28.2891704
cyl .
disp .
hp -0.0295088
drat 1.3486882
wt -2.0854970
qsec .
vs .
am .
 gear .
carb -0.5106284
```

Some features have been excluded from the model (coefficient = 0)
Ridge regression model to predict ‘mpg’

```r
> library(glmnet)
> fit = glmnet(as.matrix(mtcars_train[, -1]), mtcars_train[, 1], alpha=0, lambda=1)
> coef(fit)
11 x 1 sparse Matrix of class "dgCMatrix"
  s0
(Intercept) 21.908779196
cyl     -0.177159879
disp    -0.002701168
hp      -0.021681177
drat     2.978921208
wt       -1.197334809
qsec    0.150467388
vs      -0.292731083
am       1.629256517
gear   -0.751923514
carb    -1.474598851
```
Regularization on Mtcars dataset

• Let us first try lasso regression and ridge regression with $\lambda = 1$

• Next let us find the best $\lambda$ with cross-validation (tune $\lambda$)
Lasso regression model to predict ‘mpg’

```r
> library(glmnet)
> lambdas = 10^seq(4,-4,by=-0.1)
> fit =
  cv.glmnet(as.matrix(mtcars_train[,-1]),mtcars_train[,1],alpha=1,lambda=lambdas)
> coef(fit)
11 x 1 sparse Matrix of class "dgCMatrix"

     s0
(Intercept)  28.30709226
cyl          .
disp         .
hp     -0.02957381
drat       1.34533317
wt      -2.08692289
qsec        .
vs          .
am         .
gear        .
carb    -0.50729963
```
Ridge regression model to predict ‘mpg’

```r
> library(glmnet)
> lambdas = 10^seq(4,-4,by=-0.1)
> fit =
cv.glmnet(as.matrix(mtcars_train[,1]),mtcars_train[,1],alpha=0,lambda=lambdas)
> coef(fit)
11 x 1 sparse Matrix of class "dgCMatrix"
  s0
(Intercept) 21.567310854
cyl   -0.307376760
disp  -0.004765816
hp   -0.010997820
drat   0.976418690
wt   -0.764100211
qsec   0.129659556
vs     0.720776200
am    0.954542760
gear  0.346222196
carb -0.617503068
```
## Comparison of methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Training MSE</th>
<th>Test MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>36.63859</td>
<td>34.57625</td>
</tr>
<tr>
<td>Simple linear with ‘wt’</td>
<td>9.525724</td>
<td>12.18288</td>
</tr>
<tr>
<td>Multivariate linear</td>
<td>0.4408109</td>
<td>337.9995</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>6.745241</td>
<td>7.494272</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 1$</td>
<td>4.587832</td>
<td>11.24904</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>6.74551</td>
<td>7.492263</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 12.6$</td>
<td>8.538854</td>
<td>6.505381</td>
</tr>
</tbody>
</table>
Can we improve the fit?
How to fit a non-linear curve?
Lecture 09: Machine learning 3

✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ ROC curves
✓ Regression
✓ Simple linear regression
✓ Multi-variate linear regression
✓ Regularization
• Fitting non-linear regression curves
Fitting non-linear regression curves
Multivariate linear regression

Linear model requires parameters to be linear, not features!

This is linear model

\[ y = \beta_0 + \beta_1 x_1^2 + \beta_2 x_1 + \beta_3 x_2 \]

\[ x' = \phi(x) \]

This is linear model

\[ y = \beta_0 + \beta_1 x_1^7 + \beta_2 x_1^3 + \beta_3 x_1 + \beta_4 x_2^2 \]

\[ x^z, \sqrt{x}, \log(x) \ldots \]

This is not linear model

\[ y = \beta_0 + \beta_1 x_1 + \beta_2^2 x_2 \]
Quadratic model to predict ‘mpg’ from ‘wt’

```r
> fit = lm(formula = mpg ~ wt + I(wt^2), data = mtcars_train)
> summary(fit)

Call:
  lm(formula = mpg ~ wt + I(wt^2), data = mtcars_train)

Residuals:
     Min      1Q  Median      3Q     Max
-4.8331 -1.9087  0.3109  1.9749  3.9054

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)    58.0658    7.5685    7.672 3.53e-06 ***
wt           -16.6601    4.2593   -3.911 0.00179  **
I(wt^2)           1.5305    0.5698    2.686 0.01869 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.746 on 13 degrees of freedom
Multiple R-squared:  0.8328,    Adjusted R-squared:  0.8071
F-statistic: 32.38 on 2 and 13 DF,  p-value: 8.934e-06
```
## Comparison of methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Training MSE</th>
<th>Test MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>36.63859</td>
<td>34.57625</td>
</tr>
<tr>
<td>Simple linear with ‘wt’</td>
<td>9.525724</td>
<td>12.18288</td>
</tr>
<tr>
<td>Multivariate linear</td>
<td>0.4408109</td>
<td>337.9995</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>6.745241</td>
<td>7.494272</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 1$</td>
<td>4.587832</td>
<td>11.24904</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>6.74551</td>
<td>7.492263</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 12.6$</td>
<td>8.538854</td>
<td>6.505381</td>
</tr>
<tr>
<td>Quadratic with ‘wt’</td>
<td>6.125943</td>
<td>13.63225</td>
</tr>
</tbody>
</table>
## Comparison of methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Training MSE</th>
<th>Test MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>36.63859</td>
<td>34.57625</td>
</tr>
<tr>
<td>Simple linear with ‘wt’</td>
<td>9.525724</td>
<td>12.18288</td>
</tr>
<tr>
<td>Multivariate linear</td>
<td>0.4408109</td>
<td>337.9995</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>6.74551</td>
<td>7.492263</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 1$</td>
<td>4.587832</td>
<td>11.24904</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>8.538854</td>
<td>6.505381</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 12.6$</td>
<td>6.125943</td>
<td>13.63225</td>
</tr>
<tr>
<td>Quadratic with ‘wt’</td>
<td>6.125943</td>
<td>13.63225</td>
</tr>
</tbody>
</table>

- **Linear in ‘wt’**
- **Quadratic in ‘wt’**
Comparison of methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Training MSE</th>
<th>Test MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>36.63859</td>
<td>34.57625</td>
</tr>
<tr>
<td>Simple linear with ‘wt’</td>
<td>9.525724</td>
<td>12.18288</td>
</tr>
<tr>
<td>Multivariate linear</td>
<td>0.4408109</td>
<td>337.9995</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>6.74551</td>
<td>7.492263</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 1$</td>
<td>4.587832</td>
<td>11.24904</td>
</tr>
<tr>
<td>Lasso regression $\lambda = 1$</td>
<td>6.74551</td>
<td>7.492263</td>
</tr>
<tr>
<td>Ridge regression $\lambda = 12.6$</td>
<td>8.538854</td>
<td>6.505381</td>
</tr>
<tr>
<td>Quadratic with ‘wt’</td>
<td>6.125943</td>
<td>13.63225</td>
</tr>
</tbody>
</table>

Quadratic in ‘wt’

Linear in ‘wt’

Quadratic is overfitting!
Linear regression with categorical features?

- A categorical feature can be represented as many binary features:
  - Grade \{A,B,C,D,E,F\} can be transformed into:
    - gradeIsA (1/0) \([1=\text{yes}, \, 0=\text{no}]\)
    - gradeIsB (1/0)
    - ...
    - gradeIsF (1/0)
- Then can run linear regression as with numeric features
- A method dedicated to regression with categorical features:
  - ANOVA (ANalysis Of VAriance)
Linear regression works well, if:

- the relationship between $x$ and $y$ is linear
- $y$ distributed normally at each value of $x$
- no heteroscedasticity (variance is systematically changing)
- independence and normality of errors
- lack of multicollinearity (non-correlated features)

Adapted from slides by Anna Leontjeva
In practice...

- In practice, these conditions are often violated
- However, even then linear methods are often very competitive with other methods
Interpolation vs extrapolation

- Interpolation is when applying regression in the region where there are training data (inter = between training instances)
- Extrapolation is when applying regression outside the region of training data (extra = away from training instances)
- Generally, extrapolation leads to very high errors, because all training data are only on one side from test data, not around it
Lecture 09: Machine learning 3

✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ ROC curves
✓ Regression
✓ Simple linear regression
✓ Multi-variate linear regression
✓ Regularization
✓ Fitting non-linear regression curves
Linear regression