Lecture 08: Machine learning 3

Meelis Kull
meelis.kull@ut.ee
Autumn 2018
Previous Lecture 07 – 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
- Trade-off between true positives and false positives
- Scoring classifiers for TP/FP trade-off
- Constructing ROC curves
- Regression
  - Simple linear regression
  - Multi-variate linear regression
- Regularization
- Fitting non-linear regression curves
• **Trade-off between true positives and false positives**
• Scoring classifiers for TP/FP trade-off
• Constructing ROC curves
• Regression
• Simple linear regression
• Multi-variate linear regression
• Regularization
• Fitting non-linear regression curves
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>Positives (Pos)</strong></td>
<td><strong>Negatives (Neg)</strong></td>
<td><strong>Total</strong></td>
</tr>
<tr>
<td>False positives (FP) (Type I error)</td>
<td><strong>False positives (FP)</strong> (Type I error)</td>
<td>True negatives (TN)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Predicted positives (PPos)** | **Predicted negatives (PNeg)** |

<table>
<thead>
<tr>
<th>Pred +</th>
<th>Pred -</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ 10</td>
<td>10</td>
</tr>
<tr>
<td>- 30</td>
<td>50</td>
</tr>
</tbody>
</table>

**Example**

\[
\text{Accuracy} = \frac{TP + TN}{Total} \quad \text{Accuracy} = \frac{10 + 50}{100} = 0.60
\]

\[
\text{Precision} = \frac{TP}{PPos} \quad \text{Precision} = \frac{10}{40} = 0.25
\]

\[
\text{Recall} = \frac{TP}{Pos} \quad \text{Recall} = \frac{10}{20} = 0.50
\]
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>False negatives (FN) (Type II error)</td>
<td>Positives (Pos)</td>
<td>Negatives (Neg)</td>
<td>Total</td>
<td></td>
</tr>
<tr>
<td>False positives (FP) (Type I error)</td>
<td>True negatives (TN)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Predicted positives (PPos) | Predicted negatives (PNeg) | 40 | 60 | 100 |

Accuracy = (TP + TN)/Total

Precision = TP/PPos

TPR = Recall = TP/Pos

FPR = FP/Neg

Example

<table>
<thead>
<tr>
<th>Pred +</th>
<th>Pred -</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>40</td>
<td>60</td>
<td>100</td>
</tr>
</tbody>
</table>

Accuracy = (10 + 50)/100 = 0.60

Precision = 10/40 = 0.25

TPR = Recall = 10/20 = 0.50

FPR = 30/80 = 0.37
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
Consider the task of classifying opossums (positive class) and ondatras (negative class) based on photos. Suppose a deep neural network says “opossum” but it is actually an ondatra.

A. This is true positive
B. This is true negative
C. This is false positive
D. This is false negative
E. None of the above
F. Not sure
✓ Trade-off between true positives and false positives

• Scoring classifiers for TP/FP trade-off
• Constructing ROC curves
• Regression
• Simple linear regression
• Multi-variate linear regression
• Regularization
• Fitting non-linear regression curves
Scoring classifiers

- Most classification models can output scores in addition to labels
- KNN (K nearest neighbours):
  - Score = proportion of positive neighbours
- SVM (support vector machine):
  - Score = signed distance to the decision boundary
- DT (decision tree):
  - Score = proportion of positive instances in the decision leaf
- RF (random forest):
  - 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

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

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

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

Adapted from slides by Dmytro Fishman
True labels

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

TPR = TP/P
FPR = FP/N

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

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

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

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

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

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 = 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?

Adapted from slides by Dmytro Fishman
True labels

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

\[TPR = TP/P\]
\[FPR = FP/(FP + TN)\]

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

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

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

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?

\[\geq 0.7 \quad TPR = ?\]
\[FPR = ?\]
For 5 instances the actual classes are \((1,1,0,1,0)\) – where 1 is positive, 0 is negative; and the model predicts scores \((0.7,0.6,0.5,0.4,0.2)\). If we predict positive only for scores that are at least 0.7 then:

\[
TPR = \text{Recall} = \frac{TP}{Pos}
\]

A. TPR = 0
B. TPR = 1/5
C. TPR = 1/4
D. TPR = 1/3
E. TPR = 4/5
F. TPR = 1
G. TPR = 4/3
H. Not sure
For 5 instances the actual classes are (1, 1, 0, 1, 0) – where 1 is positive, 0 is negative; and the model predicts scores (0.7, 0.6, 0.5, 0.4, 0.2). If we predict positive only for scores that are at least 0.7 then:

\[ FPR = \frac{FP}{Neg} \]

A. FPR = 0
B. FPR = 1/5
C. FPR = 1/4
D. FPR = 1/3
E. FPR = 1/2
F. FPR = 4/5
G. FPR = 1
H. Not sure
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)

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?

$\geq 0.7 \quad TPR = \frac{1}{3} \quad FPR = \frac{0}{(0 + 2)}$

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

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

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

\[\geq 0.7\]
\[TPR = \frac{1}{3}\]
\[FPR = 0\]

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

Let’s plot this point on a graph

Adapted from slides by Dmytro Fishman
True labels:

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

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

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

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

We shall do this procedure for all possible thresholds.

Adapted from slides by Dmytro Fishman
True labels

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

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

\[\begin{align*}
\text{TPR} & \geq 0.7 \quad \text{TPR} = \frac{1}{3} \quad \text{FPR} = 0 \\
\text{TPR} & \geq 0.6 \quad \text{TPR} = ? \quad \text{FPR} = ?
\end{align*}\]

How about TPR and FPR?

Adapted from slides by Dmytro Fishman
True labels

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

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

\[FPR = \frac{FP}{FP + TN}\]

\[\geq 0.7\quad TPR = \frac{1}{3}\quad FPR = 0\]

\[\geq 0.6\quad TPR = \frac{2}{3}\quad FPR = 0\]
True labels

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

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

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

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

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

\( \geq 0.5 \quad TPR = ? \quad FPR = ? \)

Adapted from slides by Dmytro Fishman
True labels

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

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

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

Oops, this is a false positive!

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)

>= 0.7  TPR = 1/3  FPR = 0

>= 0.6  TPR = 2/3  FPR = 0

>= 0.5  TPR = 2/3  FPR = 1/2

Adapted from slides by Dmytro Fishman
True labels

\{(1, 1, 0, 1, 0)\}

\[ TP = TP/P \]
\[ FPR = FP/(FP + TN) \]

\[ \geq 0.7 \quad TPR = 1/3 \quad FPR = 0 \]
\[ \geq 0.6 \quad TPR = 2/3 \quad FPR = 0 \]
\[ \geq 0.5 \quad TPR = 2/3 \quad FPR = 1/2 \]

And so on…

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 = 2/3  FPR = 1/2

>= 0.4  TPR = 3/3  FPR = 1/2

>= 0.2  TPR = 3/3  FPR = 2/2

Adapted from slides by Dmytro Fishman
True labels

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

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

This curve is called the Receiver Operating Characteristic (ROC)

Adapted from slides by Dmytro Fishman

Meelis Kull - Autumn 2018 - LTAT.02.002 – Intro to Data Science - Lecture 08
True labels

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

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

This curve is called the Receiver Operating Characteristic (ROC)

AUC = area under curve

AUC = 0.83

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)

AUC of 0.5 means random guess

AUC of 1 means perfect classification

(0.7, 0.6, 0.5, 0.4, 0.2)

AUC = 1

Adapted from slides by Dmytro Fishman
A random classifier ($p=0.5$) can be made better than random by inverting its predictions.

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

• ROC = Receiver Operating Characteristic
• Developed in the context of radar data analysis in World War II
ROC curves

• ROC = Receiver Operating Characteristic
• Developed in the context of radar data analysis in World War II
✓ Trade-off between true positives and false positives

✓ Scoring classifiers for TP/FP trade-off

• **Constructing ROC curves**

• Regression

• Simple linear regression

• Multi-variate linear regression

• Regularization

• Fitting non-linear regression curves
True labels

(1, 1, 0, 1, 0)

(0.7, 0.6, 0.5, 0.4, 0.2)

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
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
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**

\((0.7, 0.6, 0.5, 0.4, 0.2)\)
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.
This is called Receiver Operating Characteristic (ROC)
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)

AUC = 0.83

\[
\text{AUC} = \text{area under the (ROC) curve}
\]
✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ Constructing ROC curves

• **Regression**
  • Simple linear regression
  • Multi-variate linear regression
  • Regularization
  • Fitting non-linear regression curves
## 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>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>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

### Features

<table>
<thead>
<tr>
<th>No</th>
<th>Yes</th>
<th>Myope</th>
<th>No</th>
<th>Normal</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<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>Hyperm</td>
<td>No</td>
<td>Reduced</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hyperm</td>
<td>No</td>
<td>Normal</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Training data

<table>
<thead>
<tr>
<th>No</th>
<th>No</th>
<th>Hyperm</th>
<th>Yes</th>
<th>Reduced</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>Hyperm</td>
<td>Yes</td>
<td>Normal</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hyperm</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>
</tbody>
</table>

### Test data

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
<th>Myope</th>
<th>Yes</th>
<th>Normal</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hyperm</td>
<td>No</td>
<td>Reduced</td>
<td>???</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hyperm</td>
<td>No</td>
<td>Normal</td>
<td>???</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Hyperm</td>
<td>Yes</td>
<td>Reduced</td>
<td>???</td>
</tr>
</tbody>
</table>
Classification

Features

Categorical Label

Training data
(labels known)

Test data
(must predict)

- Features
- Categorical Label
- Training data
- (labels known)
- Test data
- (must predict)

Features

Categorical Label

Training data
(labels known)

Test data
(must predict)

- Features
- Categorical Label
- Training data
- (labels known)
- Test data
- (must predict)
Classification

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

Features

Numeric Label

Training data

Test data

(labels known)

(must predict)
The image contains a text document about the `mtcars` dataset, which is a well-known dataset in data science education. The document includes a code snippet to load the dataset and a table summarizing the dataset's structure and contents. The table shows the variables and their values for various car models, including their miles per gallon (mpg), cylinders (cyl), displacement (disp), horsepower (hp), drat (rear axle ratio), weight (wt), 100mph quarter mile (qsec), number of forward gears (gear), and carburetors (carb). The data is used as an example in an introduction to data science lecture.
Example: Mtcars dataset

```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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 230</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 280</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 28C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 440</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 454</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lincoln Continental</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chrysler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chrysler 300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fiat 128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fiat 1500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Honda Civic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toyota Corolla</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toyota Corona</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dodge Challenger</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMC Javelin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMC Javelin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camaro</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pontiac Firebird</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pontiac Firebird</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fiat X19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Porsche 914</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Porsche 914</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lotus Europa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lotus Europa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ford Pantera L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ferrari Dino</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ferrari Dino</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maserati Bora</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maserati Bora</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volvo 142E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- **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

<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>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>
<td>4</td>
</tr>
<tr>
<td>Mazda RX4 Wag</td>
<td>21.0</td>
<td>6</td>
<td>160.0</td>
<td>3.80</td>
<td>2.875</td>
<td>17.02</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Datsun 710</td>
<td>22.8</td>
<td>4</td>
<td>108</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>
<td>1</td>
</tr>
<tr>
<td>Hornet 4 Drive</td>
<td>21.4</td>
<td>6</td>
<td>258.0</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>
<td>3</td>
</tr>
<tr>
<td>Hornet Sportabout</td>
<td>18.7</td>
<td>8</td>
<td>360.0</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>
<td>2</td>
</tr>
<tr>
<td>Valiant</td>
<td>18.1</td>
<td>6</td>
<td>225.0</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>
<td>1</td>
</tr>
<tr>
<td>Duster 360</td>
<td>14.3</td>
<td>8</td>
<td>360.0</td>
<td>2.44</td>
<td>3.570</td>
<td>15.84</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Merc 240D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 230</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 280</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 496</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 430</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merc 450</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mercury 98</td>
<td>18.7</td>
<td>8</td>
<td>360.0</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>
<td>2</td>
</tr>
<tr>
<td>Lincoln Continental</td>
<td>14.3</td>
<td>8</td>
<td>360.0</td>
<td>2.44</td>
<td>3.570</td>
<td>15.84</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Chrysler Imperial</td>
<td>14.7</td>
<td>8</td>
<td>440.0</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>
<td>4</td>
</tr>
<tr>
<td>Chrysler 300</td>
<td>32.4</td>
<td>4</td>
<td>78.7</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>
<td>1</td>
</tr>
<tr>
<td>Honda Civic</td>
<td>30.4</td>
<td>4</td>
<td>75.7</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>
<td>2</td>
</tr>
<tr>
<td>Toyota Corolla</td>
<td>33.9</td>
<td>4</td>
<td>71.1</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>
<td>1</td>
</tr>
<tr>
<td>Toyota Corona</td>
<td>21.5</td>
<td>4</td>
<td>120.1</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>
<td>1</td>
</tr>
<tr>
<td>Dodge Challenger</td>
<td>15.5</td>
<td>8</td>
<td>318.0</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>
<td>2</td>
</tr>
<tr>
<td>AMC Javelin</td>
<td>15.2</td>
<td>8</td>
<td>304.0</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>
<td>2</td>
</tr>
<tr>
<td>Camaro Z28</td>
<td>13.3</td>
<td>8</td>
<td>350.0</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>
<td>4</td>
</tr>
<tr>
<td>Pontiac Firebird</td>
<td>19.2</td>
<td>8</td>
<td>400.0</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>
<td>2</td>
</tr>
<tr>
<td>Fiat X1-9</td>
<td>27.3</td>
<td>4</td>
<td>79.0</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>
<td>1</td>
</tr>
<tr>
<td>Porsche 914-2</td>
<td>26.0</td>
<td>4</td>
<td>120.3</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>
<td>2</td>
</tr>
<tr>
<td>Lotus Europa</td>
<td>30.4</td>
<td>4</td>
<td>95.1</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>
<td>2</td>
</tr>
<tr>
<td>Ford Pantera L</td>
<td>15.8</td>
<td>8</td>
<td>351.0</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>
<td>4</td>
</tr>
<tr>
<td>Ferrari Dino</td>
<td>19.7</td>
<td>6</td>
<td>145.0</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>
<td>6</td>
</tr>
<tr>
<td>Maserati Bora</td>
<td>15.0</td>
<td>8</td>
<td>301.0</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>
<td>8</td>
</tr>
<tr>
<td>Volvo 142E</td>
<td>21.4</td>
<td>4</td>
<td>121.0</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>
<td>2</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>Make</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>
Correlations of features and label

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’
How to predict ‘mpg’ from ‘wt’?

![Graph showing the relationship between mpg and wt.](image)
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?
Which value would you predict?

A. A
B. B
C. C
D. D
E. E
F. Not sure
✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ Constructing ROC curves
✓ Regression
  • **Simple linear regression**
  • Multi-variate linear regression
  • Regularization
  • Fitting non-linear regression curves
Simple linear regression

\[ mpg = \alpha + \beta \, wt + \epsilon \]

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

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

Intercept: $\alpha = 38.86$

Slope: $\beta = -5.38$

Residual at this particular training instance: $\epsilon = -6.01$
Which regression line is best?
Which line would you use for prediction?

A. A
B. B
C. C
D. Not sure
Which regression line is best?

We want to minimize residuals!
Which regression line is best?

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:

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

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

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \quad \text{and} \quad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i
\]

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

Fitted line (model): \( mpg = 38.8649 - 5.3792 \, wt + \epsilon \)

Intercept: \( \alpha = 38.86 \)

Slope: \( \beta = -5.38 \)
### Training dataset (50% of all data)

<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 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>
</tbody>
</table>

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

\[
mpg = -127.77 + 10.29 \text{ cyl} - 0.019 \text{ disp} + \cdots - 6.13 \text{ carb}
\]
✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ Constructing ROC curves
✓ Regression
✓ Simple linear regression
  • **Multi-variate linear regression**
  • Regularization
  • Fitting non-linear regression curves
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: \( \hat{y} = X\beta \)

\[
\begin{pmatrix}
\hat{y}_1 \\
:\ \\
\hat{y}_n
\end{pmatrix} =
\begin{pmatrix}
1 & x_{11} & \ldots & x_{1k} \\
:\ & : & \ldots & : \\
1 & x_{n1} & \ldots & x_{np}
\end{pmatrix} \times
\begin{pmatrix}
\beta_0 \\
:\ \\
\beta_k
\end{pmatrix}
\]

Adapted from slides by Anna Leontjeva
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:

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

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

Adapted from slides by Anna Leontjeva
Multivariate linear regression

$$\frac{\partial MSE}{\partial \beta} = -2y^TX + 2X^TX\beta$$

$$\frac{\partial MSE}{\partial \beta} = 0 \Rightarrow \beta = (X^TX)^{-1}X^Ty$$

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’

Fitted hyperplane (linear model):

\[
mpg = -127.773 + 10.298 \, cyl - 0.019 \, disp - 0.221 \, hp \\
+ 24.966 \, drat + 9.632 \, wt + 0.276 \, qsec + 3.839 \, vs + 1.718 \, am \\
+ 1.887 \, gear - 6.138 \, carb
\]
Overfitting...

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

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

- **Multivariate linear regression model:**
  - \( mpg = -127.773 + 10.298 \text{ cyl} + \cdots - 6.138 \text{ carb} \)
  - 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
✓ Trade-off between true positives and false positives
✓ Scoring classifiers for TP/FP trade-off
✓ Constructing ROC curves
✓ Regression
✓ Simple linear regression
✓ Multi-variate linear regression
• **Regularization**
• Fitting non-linear regression curves
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’

(Intercept) 28.2891704
cyl .
disp .
hp -0.0295088
drat 1.3486882
wt -2.0854970
qsec .
vs .
am .
gear .
carb -0.5106284
### Lasso regression model to predict ‘mpg’

<table>
<thead>
<tr>
<th>Feature</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>28.2891704</td>
</tr>
<tr>
<td>cyl</td>
<td>.</td>
</tr>
<tr>
<td>disp</td>
<td>.</td>
</tr>
<tr>
<td>hp</td>
<td>-0.0295088</td>
</tr>
<tr>
<td>drat</td>
<td>1.3486882</td>
</tr>
<tr>
<td>wt</td>
<td>-2.0854970</td>
</tr>
<tr>
<td>qsec</td>
<td>.</td>
</tr>
<tr>
<td>vs</td>
<td>.</td>
</tr>
<tr>
<td>am</td>
<td>.</td>
</tr>
<tr>
<td>gear</td>
<td>.</td>
</tr>
<tr>
<td>carb</td>
<td>-0.5106284</td>
</tr>
</tbody>
</table>

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

(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’

\[
\begin{align*}
\text{(Intercept)} & : 28.30709226 \\
cyl & : -0.02957381 \\
disp & : 1.34533317 \\
hp & : 1.34533317 \\
drat & : -0.02957381 \\
w t & : 1.34533317 \\
qsec & : -2.08692289 \\
vs & : -2.08692289 \\
am & : -2.08692289 \\
gear & : 0.50729963 \\
carb & : 0.50729963
\end{align*}
\]
Ridge regression model to predict ‘mpg’

(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?

 mpg

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

• Fitting non-linear regression curves
Multivariate linear regression

Linear model requires parameters to be linear, not features!

This is linear model

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

This is linear model

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

This is not linear model

\[ y = \beta_0 + \beta_1 x_1 + \beta_2^2 x_2 \]
Fitted model:

\[ mpg = 58.066 - 16.660 \, wt + 1.531 \, wt^2 \]
## 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>6.125943</td>
<td>13.63225</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>

- **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>0.44081109</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>

- **Linear in ‘wt’**
- **Quadratic 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=yes, 0=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
Linear regression

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