Predictive Analysis of Text: Concepts, Features, and Instances

Jaime Arguello
jarguell@email.unc.edu

August 27, 2014
Objective: developing and evaluating computer programs that automatically detect a particular concept in natural language text
Predictive Analysis
basic ingredients

1. **Training data**: a set of positive and negative examples of the concept we want to automatically recognize

2. **Representation**: a set of features that we believe are useful in recognizing the desired concept

3. **Learning algorithm**: a computer program that uses the training data to learn a predictive model of the concept
Predictive Analysis
basic ingredients

4. **Model**: a function that describes a predictive relationship between feature values and the presence/absence of the concept

5. **Test data**: a set of previously unseen examples used to estimate the model’s effectiveness

6. **Performance metrics**: a set of statistics used to measure the predictive effectiveness of the model
Predictive Analysis
training and testing

labeled examples

machine learning algorithm

model

new, unlabeled examples

predictions

training

model

testing
## Predictive Analysis

concept, instances, and features

<table>
<thead>
<tr>
<th>instances</th>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>color</td>
<td>size</td>
</tr>
<tr>
<td>red</td>
<td>red</td>
<td>big</td>
</tr>
<tr>
<td>green</td>
<td>green</td>
<td>big</td>
</tr>
<tr>
<td>blue</td>
<td>blue</td>
<td>small</td>
</tr>
<tr>
<td>blue</td>
<td>blue</td>
<td>small</td>
</tr>
<tr>
<td>red</td>
<td>red</td>
<td>big</td>
</tr>
</tbody>
</table>
Predictive Analysis
training and testing

training

machine learning algorithm

model

labeled examples

new, unlabeled examples

color size sides equal sides ... label
red big 3 no ... yes
green big 3 yes ... yes
blue small inf yes ... no
blue small 4 yes ... no
... ... ... ... ...
red big 3 yes ... yes

color size sides equal sides ... label
red big 3 no ... ???
green big 3 yes ... ???
blue small inf yes ... ???
blue small 4 yes ... ???...
... ... ... ...
red big 3 yes ... ???

color size sides equal sides ... label
red big 3 no yes
green big 3 yes yes
blue small inf yes no
blue small 4 yes no
... ...
red big 3 yes yes

predictions
Predictive Analysis
questions

• Is a particular concept appropriate for predictive analysis?
• What should the unit of analysis be?
• How should I divide the data into training and test sets?
• What is a good feature representation for this task?
• What type of learning algorithm should I use?
• How should I evaluate my model’s performance?
Learning algorithms can recognize some concepts better than others.

What are some properties of concepts that are easier to recognize?
Predictive Analysis

- Option 1: can a human recognize the concept?
Predictive Analysis

concepts

• **Option 1:** can a human recognize the concept?

• **Option 2:** can two or more humans recognize the concept independently and do they agree?
Option 1: can a human recognize the concept?

Option 2: can two or more humans recognize the concept independently and do they agree?

Option 2 is better.

In fact, models are sometimes evaluated as an independent assessor.

How does the model’s performance compare to the performance of one assessor with respect to another?

- One assessor produces the “ground truth” and the other produces the “predictions”
Predictive Analysis
measures agreement: percent agreement

- **Percent agreement:** percentage of instances for which both assessors agree that the concept occurs or does not occur

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>no</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

\[
\frac{A + B}{A + B + C + D}
\]
Predictive Analysis
measures agreement: percent agreement

- **Percent agreement**: percentage of instances for which both assessors agree that the concept occurs or does not occur

\[
\frac{(A + D)}{(A + B + C + D)}
\]
Predictive Analysis measures agreement: percent agreement

- **Percent agreement**: percentage of instances for which both assessors agree that the concept occurs or does not occur

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>no</td>
<td>15</td>
<td>75</td>
</tr>
</tbody>
</table>

% agreement = ???
### Predictive Analysis

measures agreement: percent agreement

- **Percent agreement:** percentage of instances for which both assessors agree that the concept occurs or does not occur.

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>no</td>
<td>15</td>
<td>75</td>
<td>90</td>
</tr>
</tbody>
</table>

$$%\text{ agreement} = \frac{(5 + 75)}{100} = 80\%$$
Predictive Analysis
measures agreement: percent agreement

- **Problem**: percent agreement does not account for agreement due to random chance.
- How can we compute the expected agreement due to random chance?
  - **Option 1**: assume unbiased assessors
  - **Option 2**: assume biased assessors
### Predictive Analysis

**kappa agreement: chance-corrected % agreement**

- **Option 1:** unbiased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>yes</strong></td>
<td>??</td>
<td>??</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>??</td>
<td>??</td>
</tr>
</tbody>
</table>

**50**

**50**
Predictive Analysis
kappa agreement: chance-corrected % agreement

- Option 1: unbiased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>no</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>
Predictive Analysis
kappa agreement: chance-corrected % agreement

- **Option 1:** unbiased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>no</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

random chance % agreement = ???
Predictive Analysis
kappa agreement: chance-corrected % agreement

• Option 1: unbiased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>25</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>no</td>
<td>25</td>
<td>25</td>
<td>50</td>
</tr>
</tbody>
</table>

random chance % agreement = \( \frac{25 + 25}{100} = 50\% \)
Predictive Analysis
kappa agreement: chance-corrected % agreement

• Kappa agreement: percent agreement after correcting for the expected agreement due to random chance

\[ \kappa = \frac{P(a) - P(e)}{1 - P(e)} \]

• \( P(a) = \) percent of observed agreement
• \( P(e) = \) percent of agreement due to random chance
### Predictive Analysis

**Kappa agreement:** percent agreement after correcting for the expected agreement due to unbiased chance

- **Kappa agreement** is a measure of agreement between raters or judges beyond what would be expected by chance.

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>no</td>
<td>15</td>
<td>75</td>
</tr>
</tbody>
</table>

- Total: 20 yes, 80 no

- **Expected agreement** ($P(e)$) is the probability of agreement expected by chance.

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>no</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

- Total: 50 yes, 50 no

#### Calculations

- **Observed agreement** ($P(a)$):

  $$P(a) = \frac{5 + 75}{100} = 0.80$$

- **Expected agreement** ($P(e)$):

  $$P(e) = \frac{25 + 25}{100} = 0.50$$

- **Kappa ($\kappa$)**:

  $$\kappa = \frac{P(a) - P(e)}{1 - P(e)} = \frac{0.80 - 0.50}{1 - 0.50} = 0.60$$
**Predictive Analysis**

**kappa agreement: chance-corrected % agreement**

- **Option 2**: biased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>yes</strong></td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>15</td>
<td>75</td>
</tr>
</tbody>
</table>

**biased chance % agreement = ???**
Predictive Analysis

kappa agreement: chance-corrected % agreement

- Kappa agreement: percent agreement after correcting for the expected agreement due to biased chance

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>no</td>
<td>15</td>
<td>75</td>
</tr>
</tbody>
</table>

\[
P(a) = \frac{5 + 75}{100} = 0.80
\]
\[
P(e) = \left(\frac{10}{100} \times \frac{20}{100}\right) + \left(\frac{90}{100} \times \frac{80}{100}\right) = 0.74
\]
\[
\kappa = \frac{P(a) - P(e)}{1 - P(e)} = \frac{0.80 - 0.74}{1 - 0.74} = 0.23
\]
Predictive Analysis

data annotation process

• **INPUT:** unlabeled data, annotators, coding manual

• **OUTPUT:** labeled data

1. using the latest coding manual, have all annotators label some previously unseen portion of the data (~10%)

2. measure inter-annotator agreement (Kappa)

3. **IF** agreement < X, **THEN:**
   - refine coding manual using disagreements to resolve inconsistencies and clarify definitions
   - return to 1

**ELSE**

- have annotators label the remainder of the data independently and **EXIT**
What is good (Kappa) agreement?

It depends on who you ask

According to Landis and Koch, 1977:

- 0.81 - 1.00: almost perfect
- 0.61 - 0.70: substantial
- 0.41 - 0.60: moderate
- 0.21 - 0.40: fair
- 0.00 - 0.20: slight
- < 0.00: no agreement
Predictive Analysis

questions

• Is a particular concept appropriate for predictive analysis?
• What should the unit of analysis be?
• How should I divide the data into training and test sets?
• What is a good feature representation for this task?
• What type of learning algorithm should I use?
• How should I evaluate my model’s performance?
For many text-mining applications, turning the data into instances for training and testing is fairly straightforward.

- **Easy case:** instances are self-contained, independent units of analysis
  - text classification: instances = documents
  - opinion mining: instances = product reviews
  - bias detection: instances = political blog posts
  - emotion detection: instances = support group posts
## Text Classification
predicting health-related documents

<table>
<thead>
<tr>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td>label</td>
</tr>
<tr>
<td>w_2</td>
<td>health</td>
</tr>
<tr>
<td>w_3</td>
<td>other</td>
</tr>
<tr>
<td>...</td>
<td>other</td>
</tr>
<tr>
<td>w_n</td>
<td>other</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>instances</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>health</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>health</td>
<td></td>
</tr>
</tbody>
</table>
Opinion Mining
predicting positive/negative movie reviews

<table>
<thead>
<tr>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td>w_2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## Bias Detection
predicting liberal/conservative blog posts

<table>
<thead>
<tr>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>instances</td>
<td>label</td>
</tr>
<tr>
<td>w_1</td>
<td>w_2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

liberal
conservative
conservative
conservative
conservative
...
Predictive Analysis
turning data into (training and test) instances

- A not-so-easy case: relational data
- The concept to be learned is a relation between pairs of objects
Predictive Analysis

example of relational data: Brother(X,Y)

(example borrowed and modified from Witten et al. textbook)
Predictive Analysis
example of relational data: Brother($X,Y$)

<table>
<thead>
<tr>
<th>name_1</th>
<th>gender_1</th>
<th>mother_1</th>
<th>father_1</th>
<th>name_2</th>
<th>gender_2</th>
<th>mother_2</th>
<th>father_2</th>
<th>brother</th>
</tr>
</thead>
<tbody>
<tr>
<td>steven</td>
<td>male</td>
<td>peggy</td>
<td>peter</td>
<td>graham</td>
<td>male</td>
<td>peggy</td>
<td>peter</td>
<td>yes</td>
</tr>
<tr>
<td>ian</td>
<td>male</td>
<td>grace</td>
<td>ray</td>
<td>brian</td>
<td>male</td>
<td>grace</td>
<td>ray</td>
<td>yes</td>
</tr>
<tr>
<td>anna</td>
<td>female</td>
<td>pam</td>
<td>ian</td>
<td>nikki</td>
<td>female</td>
<td>pam</td>
<td>ian</td>
<td>no</td>
</tr>
<tr>
<td>pippa</td>
<td>female</td>
<td>grace</td>
<td>ray</td>
<td>brian</td>
<td>male</td>
<td>grace</td>
<td>ray</td>
<td>no</td>
</tr>
<tr>
<td>steven</td>
<td>male</td>
<td>peggy</td>
<td>peter</td>
<td>brian</td>
<td>male</td>
<td>grace</td>
<td>ray</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>anna</td>
<td>female</td>
<td>pam</td>
<td>ian</td>
<td>brian</td>
<td>male</td>
<td>grace</td>
<td>ray</td>
<td>no</td>
</tr>
</tbody>
</table>

**features**

**concept**

Predictive Analysis example of relational data: Brother($X,Y$)
Predictive Analysis
turning data into (training and test) instances

- A not-so-easy case: relational data
- Each instance should correspond to an object pair (which may or may not share the relation of interest)
- May require features that characterize properties of the pair
Predictive Analysis
example of relational data: Brother(X,Y)

<table>
<thead>
<tr>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>name_1</td>
<td>gender_1</td>
</tr>
<tr>
<td>steven</td>
<td>male</td>
</tr>
<tr>
<td>ian</td>
<td>male</td>
</tr>
<tr>
<td>anna</td>
<td>female</td>
</tr>
<tr>
<td>pippa</td>
<td>female</td>
</tr>
<tr>
<td>steven</td>
<td>male</td>
</tr>
<tr>
<td>anna</td>
<td>female</td>
</tr>
</tbody>
</table>

(can we think of a better feature representation?)
### Predictive Analysis

**example of relational data: Brother\((X,Y)\)**

<table>
<thead>
<tr>
<th>instances</th>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>gender_1</td>
<td>gender_2</td>
</tr>
<tr>
<td>male</td>
<td>male</td>
<td>male</td>
</tr>
<tr>
<td>male</td>
<td>male</td>
<td>male</td>
</tr>
<tr>
<td>female</td>
<td>female</td>
<td>female</td>
</tr>
<tr>
<td>female</td>
<td>male</td>
<td>male</td>
</tr>
<tr>
<td>male</td>
<td>male</td>
<td>male</td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
<tr>
<td>female</td>
<td>male</td>
<td>male</td>
</tr>
</tbody>
</table>
Predictive Analysis
turning data into (training and test) instances

• A not-so-easy case: relational data
• There is still an issue that we’re not capturing! Any ideas?
• Hint: In this case, should the predicted labels really be independent?
Predictive Analysis
turning data into (training and test) instances

\[
\begin{align*}
\text{Brother}(A,B) &= \text{yes} \\
\text{Brother}(B,C) &= \text{yes} \\
\text{Brother}(A,C) &= \text{no}
\end{align*}
\]
• In this case, what we would really want is:
  ▸ a method that does *joint* prediction on the test set
  ▸ a method whose joint predictions satisfy a set of known properties about the data as a whole (e.g., transitivity)
Predictive Analysis
turning data into (training and test) instances

- There are learning algorithms that incorporate relational constraints between predictions
- However, they are beyond the scope of this class
- We’ll be covering algorithms that make independent predictions on instances
- That said, many algorithms output prediction confidence values
- Heuristics can be used to disfavor inconsistencies
Examples of relational data in text-mining:

- **information extraction**: predicting that a word-sequence belongs to a particular class (e.g., person, location)
- **topic segmentation**: segmenting discourse into topically coherent chunks
Predictive Analysis

topic segmentation example
Predictive Analysis

topic segmentation example: instances
Predictive Analysis

topic segmentation example: independent instances?
Predictive Analysis

topic segmentation example: independent instances?
Predictive Analysis
questions

• Is a particular concept appropriate for predictive analysis?
• What should the unit of analysis be?
• How should I divide the data into training and test sets?
• What is a good feature representation for this task?
• What type of learning algorithm should I use?
• How should I evaluate my model’s performance?
Predictive Analysis
training and test data

• We want our model to “learn” to recognize a concept
• So, what does it mean to learn?
The machine learning definition of learning:

A machine learns with respect to a particular task T, performance metric P, and experience E, if the system improves its performance P at task T following experience E. -- Tom Mitchell
We want our model to improve its generalization performance!

That is, its performance on previously unseen data!

Generalize: to derive or induce a general conception or principle from particulars. -- Merriam-Webster

In order to test generalization performance, the training and test data cannot be the same.

Why?
Training data + Representation
what could possibly go wrong?
• While we don’t want to test on training data, models usually perform the best when the training and test set are derived from the same “probability distribution”.

• What does that mean?
Predictive Analysis
training and test data

Data

Training Data

Test Data

positive instances

negative instances
Predictive Analysis
training and test data

• Is this a good partitioning? Why or why not?
Predictive Analysis
training and test data

Data

Random Sample
Training Data

Random Sample
Test Data

positive instances

negative instances
Predictive Analysis
training and test data

• On average, random sampling should produce comparable data for training and testing

Data

Training Data

Test Data

positive instances

negative instances
• Models usually perform the best when the training and test set have:
  ‣ a similar proportion of positive and negative examples
  ‣ a similar co-occurrence of feature-values and each target class value
Predictive Analysis
training and test data

• **Caution:** in some situations, partitioning the data randomly might inflate performance in an unrealistic way!

• How the data is split into training and test sets determines what we can claim about generalization performance

• The appropriate split between training and test sets is usually determined on a case-by-case basis
Spam detection: should the training and test sets contain email messages from the same sender, same recipient, and/or same timeframe?

Topic segmentation: should the training and test sets contain potential boundaries from the same discourse?

Opinion mining for movie reviews: should the training and test sets contain reviews for the same movie?

Sentiment analysis: should the training and test sets contain blog posts from the same discussion thread?
Predictive Analysis
questions

• Is a particular concept appropriate for predictive analysis?

• What should the unit of analysis be?

• How should I divide the data into training and test sets?

• What is a good feature representation for this task?

• What type of learning algorithm should I use?

• How should I evaluate my model’s performance?
Predictive Analysis
three types of classifiers

• Linear classifiers
• Decision tree classifiers
• Instance-based classifiers
All types of classifiers learn to make predictions based on the input feature values.

However, different types of classifiers combine the input feature values in different ways.

Chapter 3 in the book refers to a trained model as knowledge representation.
Predictive Analysis
linear classifiers: perceptron algorithm

\[ y = \begin{cases} 
1 & \text{if } w_0 + \sum_{j=1}^{n} w_j x_j > 0 \\
0 & \text{otherwise}
\end{cases} \]
Predictive Analysis
linear classifiers: perceptron algorithm

\[ y = \begin{cases} 1 & \text{if } w_0 + \sum_{j=1}^{n} w_j x_j > 0 \\ 0 & \text{otherwise} \end{cases} \]

parameters learned by the model

predicted value (e.g., 1 = positive, 0 = negative)
Predictive Analysis
linear classifiers: perceptron algorithm

<table>
<thead>
<tr>
<th>f_1</th>
<th>f_2</th>
<th>f_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.0</td>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>w_0</th>
<th>w_1</th>
<th>w_2</th>
<th>w_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>-5.0</td>
<td>2.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

output = 2.0 + (0.50 \times -5.0) + (1.0 \times 2.0) + (0.2 \times 1.0)

output = 1.7

output prediction = positive
Predictive Analysis
linear classifiers: perceptron algorithm

(two-feature example borrowed from Witten et al. textbook)
Predictive Analysis
linear classifiers: perceptron algorithm

• Would a linear classifier do well on positive (black) and negative (white) data that looks like this?
Predictive Analysis
three types of classifiers

• Linear classifiers
• Decision tree classifiers
• Instance-based classifiers
Predictive Analysis
example of decision tree classifier: \( \text{Brother}(X,Y) \)

same parents

- yes
- no

gender_1

- male
- female

- male
- female

gender_2

- yes
- no
• Draw a decision tree that would perform perfectly on this training data!
Predictive Analysis
three types of classifiers

- Linear classifiers
- Decision tree classifiers
- Instance-based classifiers
Predictive Analysis
instance-based classifiers

- predict the class associated with the most similar training examples
Predictive Analysis
instance-based classifiers

- predict the class associated with the most similar training examples
Predictive Analysis
instance-based classifiers

- **Assumption**: instances with similar feature values should have a similar label
- Given a test instance, predict the label associated with its nearest neighbors
- There are many different similarity metrics for computing distance between training/test instances
- There are many ways of combining labels from multiple training instances
Predictive Analysis questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
- How should I evaluate my model’s performance?