Idea: Estimate probabilities $\hat{P}(y|x)$ and use those for prediction

**Probability Classifier**

Input $x$: Sentence from review
- Estimate class probability $\hat{P}(y = +1|x)$
- If $\hat{P}(y = +1|x) > 0.5$:
  - $\hat{y} = +1$
- Else:
  - $\hat{y} = -1$

**Notes:**
- Estimating the probability improves **interpretability**
Interpreting Score

\[ \text{Score}(x_i) = w^T h(x_i) \]

- \( \hat{y}_i = -1 \):
  - Very sure
  - \( P(y_i = +1 | x_i) = 0 \)

- \( \hat{y}_i = 0 \):
  - Not sure if \( \hat{y}_i = -1 \) or \( \hat{y}_i = +1 \)
  - \( P(y_i = +1 | x_i) = 0.5 \)

- \( \hat{y}_i = +1 \):
  - Very sure
  - \( P(y_i = +1 | x_i) = 1 \)
\[ \hat{P}(y = +1|x, \hat{w}) = \text{sigmoid} \left( \hat{w}^T h(x) \right) = \frac{1}{1 + e^{-\hat{w}^T h(x)}} \]
Naïve Bayes
Idea: Naïve Bayes

$x = \text{“The sushi & everything else was awesome!”} $  \\
P \left( y = +1 \mid x = \text{“The sushi & everything else was awesome!”} \right) \text{?}  \\
P \left( y = -1 \mid x = \text{“The sushi & everything else was awesome!”} \right) \text{?}$

Idea: Select the class that is the most likely!

Bayes Rule:

\[
P(y = +1 \mid x) = \frac{P(x \mid y = +1) P(y = +1)}{P(x)}
\]

Example

\[
P \left( \text{“The sushi & everything else was awesome!”} \mid y = +1 \right) \cdot P(y = +1)
\]

\[
P \left( \text{“The sushi & everything else was awesome!”} \mid y = +1 \right)
\]

Since we’re just trying to find out which class has the greater probability, we can discard the divisor.
Naïve Assumption

**Idea:** Select the class with the highest probability!

**Problem:** We have not seen the sentence before.

**Assumption:** Words are independent from each other.

\[ x = "The sushi & everything else was awesome!" \]

\[
P("The sushi & everything else was awesome!" | y = +1) \cdot P(y = +1) \\
= P(\text{The} | y = +1) \cdot P(\text{sushi} | y = +1) \cdot P(\& | y = +1) \\
\quad \cdot P(\text{everything} | y = +1) \cdot P(\text{else} | y = +1) \cdot P(\text{was} | y = +1) \\
\quad \cdot P(\text{awesome} | y = +1)
\]
How do we compute something like

\[ P(y = +1)? = \frac{\# \text{ pos reviews}}{\# \text{ reviews}} \]

How do we compute something like

\[ P(\text{"awesome" } | y = +1)? = \frac{\# \text{ of times see "awesome" in pos reviews}}{\# \text{ words in all positive reviews}} \]
If a feature is missing in a class everything becomes zero.

\[ P(\text{"The sushi & everything else was awesome!" } | y = +1) \]
\[ = P(\text{The } | y = +1) \times P(\text{sushi } | y = +1) \times P(\& | y = +1) \]
\[ \times P(\text{everything } | y = +1) \times P(\text{else } | y = +1) \times P(\text{was } | y = +1) \]
\[ \times P(\text{awesome } | y = +1) \]

Solutions?

- Take the log (product becomes a sum).
  - Generally define \( \log(0) = 0 \) in these contexts
- Laplacian Smoothing (adding a constant to avoid multiplying by zero)

\[ \frac{0}{100} \rightarrow \frac{1}{100} \]
Compare Models

Logistic Regression:

\[ P(y = +1 | x, w) = \frac{1}{1 + e^{-w^T h(x)}} \]

Naïve Bayes:

\[ P(y | x_1, x_2, ..., x_d) = \prod_{j=1}^{d} P(x_j | y) \cdot P(y) \]
Compare Models

**Generative**: defines a model for generating $x$ (e.g. Naïve Bayes)

**Discriminative**: only cares about defining and optimizing a decision boundary (e.g. Logistic Regression)
How do we make decisions?

A line might not always support our decisions.

XOR
(Exclusive Or)
What makes a loan risky?

I want to buy a new house!

Credit History ★★★★★
Income ★★★
Term ★★★★★★
Personal Info ★★★
Credit history explained

Did I pay previous loans on time?

**Example:** excellent, good, or fair
What’s my income?

Example:
$80K per year
Loan terms

How soon do I need to pay the loan?

Example: 3 years, 5 years, ...
Personal information

Age, reason for the loan, marital status,...

Example: Home loan for a married couple
Intelligent application

Loan Applications

Intelligent loan application review system

Safe ✓

Risky ×

Risky ×
Classifier review

Loan Application

Classifier MODEL

Input: $x_i$

Output: $\hat{y}$
Predicted class

$\hat{y}_i = +1$
Safe

$\hat{y}_i = -1$
Risky
**Setup**

Data (N observations, 3 features)

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>risky</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>3 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
</tbody>
</table>

Evaluation: classification error

Many possible decisions: number of trees grows exponentially!
Decision Trees

- **Branch/Internal node**: splits into possible values of a feature
- **Leaf node**: final decision (the class value)
Growing Trees

• Grow the trees using a greedy approach
• What do we need?

Ls which features are “good”
Ls when to stop growing tree
Loan status: Safe  Risky

Root
6  3

# of Safe loans

# of Risky loans

N = 9 examples
Decision stump: 1 level

Loan status:
Safe Risky

Split on Credit

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>risky</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>3 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
</tbody>
</table>

Subset of data with Credit = excellent
Subset of data with Credit = fair
Subset of data with Credit = poor
making predictions

Decision Stump: Tree w/ 1 internal node

For each leaf node, set $\hat{y} = \text{majority value}$

Loan status:
Safe Risky

Root
6 3

credit?

excellent
2 0
Safe

fair
3 1
Safe

poor
1 2
Risky
How do we select the best feature?

* Select the split with lowest classification error

**Choice 1: Split on Credit**

- **Loan status:** Safe  Risky
- **Root:** 6 3
- **Credit?**
  - excellent: 2 0
  - fair: 3 1
  - poor: 1 2

**Choice 2: Split on Term**

- **Loan status:** Safe  Risky
- **Root:** 6 3
- **Term?**
  - 3 years
  - 5 years
Calculate the node values.

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>risky</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>3 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
</tbody>
</table>

Choice 2: Split on Term

Loan status:
Safe  Risky

Root
6 3

Term?

3 years
4 1

5 years
2 2
How do we select the best feature?

Select the split with lowest classification error

---

**Choice 1: Split on Credit**

<table>
<thead>
<tr>
<th>Loan status:</th>
<th>Safe</th>
<th>Risky</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>6 3</td>
<td></td>
</tr>
</tbody>
</table>

- Credit?
  - excellent 2 0
  - fair 3 1
  - poor 1 2

**Choice 2: Split on Term**

<table>
<thead>
<tr>
<th>Loan status:</th>
<th>Safe</th>
<th>Risky</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>6 3</td>
<td></td>
</tr>
</tbody>
</table>

- Term?
  - 3 years 4 1
  - 5 years 2 2
How do we measure effectiveness of a split?

**Loan status:**
- Safe
- Risky

**Root**
- 6 3

**Credit?**

- **excellent**
  - 2 0
  - **SAFE**

- **fair**
  - 3 1
  - SAFE

- **poor**
  - 1 2
  - **RISKY**

**Idea:** Calculate classification error of this decision stump

**Error**

\[
\text{Error} = \frac{\text{# mistakes}}{\text{# data points}}
\]
Calculating classification error

Step 1: \( \hat{y} = \text{class of majority of data in node} \)

Step 2: Calculate classification error of predicting \( \hat{y} \) for this data

Loan status: Safe  Risky

<table>
<thead>
<tr>
<th>Tree</th>
<th>Classification error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(root)</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Root

6 3

Safe

6 correct 3 mistakes

\( \hat{y} = \text{majority class} \)
Choice 1: Split on Credit history?

Does a split on Credit reduce classification error below 0.33?

Loan status:
Safe  Risky

Credit?

excellent  2  0

fair  3  1

poor  1  2

Choice 1: Split on Credit
Split on Credit: Classification error

**Choice 1: Split on Credit**

**Loan status:**
- Safe
- Risky

**Root:**
- 6 3

**Credit?**

- **excellent**
  - 2 0
  - Safe
  - 0 mistakes

- **fair**
  - 3 1
  - Safe
  - 1 mistake

- **poor**
  - 1 2
  - Risky
  - 1 mistake

**Error calculation:**

\[
\text{Error} = \frac{0+1+1}{9} = \frac{2}{9}
\]

\[
= 0.22
\]

**Tree Classification error**

<table>
<thead>
<tr>
<th>Tree</th>
<th>Classification error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(root)</td>
<td>0.33</td>
</tr>
<tr>
<td>Split on credit</td>
<td>0.22</td>
</tr>
</tbody>
</table>
Choice 2: Split on Term

Loan status: Safe Risky

Root
6 3

Term?

3 years
4 1
Safe

5 years
2 2
Risky
Evaluating the split on Term

Choice 2: Split on Term

Loan status: Safe Risky

Root 6 3

Term?

3 years 4 1

Safe

1 mistake

5 years 2 2

Risky

2 mistakes

Error = \( \frac{1+2}{9} = \frac{3}{9} = 0.33 \)

<table>
<thead>
<tr>
<th>Tree</th>
<th>Classification error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(root)</td>
<td>0.33</td>
</tr>
<tr>
<td>Split on credit</td>
<td>0.22</td>
</tr>
<tr>
<td>Split on term</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Choice 1 vs Choice 2: Comparing split on credit vs term

<table>
<thead>
<tr>
<th>Tree</th>
<th>Classification error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(root)</td>
<td>0.33</td>
</tr>
<tr>
<td>split on credit</td>
<td>0.22</td>
</tr>
<tr>
<td>split on loan term</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Choice 1: Split on Credit

Choice 2: Split on Term
• Given a subset of data M (a node in a tree)

• For each remaining feature $h_i(x)$:
  1. Split data of M according to feature $h_i(x)$
  2. Compute classification error of split

• Chose feature $h^*(x)$ with lowest classification error
Greedy Algorithm

- If classification for data at current node is perfect (classification error = 0):
  - Stop
- Else:
  - repeat split selection with next stump
Decision stump: 1 level

Loan status: Safe Risky

Split on Credit

Credit?

Subset of data with Credit = excellent
excellent 2 0

Subset of data with Credit = fair
fair 3 1

Subset of data with Credit = poor
poor 1 2
Stopping

- Stop if all points are in one class

**Loan status:**
- Safe
- Risky

**Root**
- 6
- 3

Credit?

- **excellent**
  - 2
  - 0
- **fair**
  - 3
  - 1
- **poor**
  - 2
  - 1

Safe

Leaf node

All data points are Safe nothing else to do with this subset of data
Tree learning = Recursive stump learning

Loan status: Safe Risky

Root
6 3

Credit?

excellent
2 0
Safe

fair
3 1
Build decision stump with subset of data where Credit = fair

poor
2 1
Build decision stump with subset of data where Credit = poor
Loan status: Safe Risky

Credit?
- excellent
  - 2 0
  - Safe
- fair
  - 3 1
- poor
  - 1 2

Term?
- 3 years
  - 2 0
  - Safe
- 5 years
  - 1 1

Income?
- high
  - 0 2
  - Risky
- Low
  - 1 0
  - Safe

Build another stump these data points
## Real valued features

<table>
<thead>
<tr>
<th>Income</th>
<th>Credit</th>
<th>Term</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>$105 K</td>
<td>excellent</td>
<td>3 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$112 K</td>
<td>good</td>
<td>5 yrs</td>
<td>Risky</td>
</tr>
<tr>
<td>$73 K</td>
<td>fair</td>
<td>3 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$69 K</td>
<td>excellent</td>
<td>5 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$217 K</td>
<td>excellent</td>
<td>3 yrs</td>
<td>Risky</td>
</tr>
<tr>
<td>$120 K</td>
<td>good</td>
<td>5 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$64 K</td>
<td>fair</td>
<td>3 yrs</td>
<td>Risky</td>
</tr>
<tr>
<td>$340 K</td>
<td>excellent</td>
<td>5 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$60 K</td>
<td>good</td>
<td>3 yrs</td>
<td>Risky</td>
</tr>
</tbody>
</table>
Threshold split

Loan status:
Safe Risky

Split on Income

Income?

< $60K
8 13

>= $60K
14 5

Subset of data with
Income >= $60K
Best threshold?

Infinite possible values of $t$

- $\text{Income} < t^*$: Predict = Risky
- $\text{Income} = t^*$: Safe
- $\text{Income} \geq t^*$: Predict = Safe

Income

- $\text{Risky}=7$, $\text{Safe}=2$
- $\text{Risky}=1$, $\text{Safe}=5$

Income

- $\$$10K
- $\$$120K
Threshold between points

Same classification error for any threshold split between $v_A$ and $v_B$
Only need to consider mid-points

Finite number of splits to consider

Income

$10K

$120K

Safe

Risky
Threshold split selection algorithm

- **Step 1:** Sort the values of a feature \( h_j(x) \):
  
  Let \( \{v_1, v_2, v_3, \ldots, v_N\} \) denote sorted values

- **Step 2:**
  - For \( i = 1 \ldots N-1 \)
    - Consider split \( t_i = (v_i + v_{i+1}) / 2 \)
    - Compute classification error for threshold split \( h_j(x) \geq t_i \)
    - Chose the \( t^* \) with the lowest classification error
Visualizing the threshold split

Threshold split is the line $\text{Age} = 38$
Split on Age $\geq 38$

![Diagram showing a scatter plot with age and income axes. The plot is divided into two regions: one for age < 38 and one for age $\geq 38$. There are symbols indicating data points, with a legend indicating that points in the younger age group are predicted to be safe, and points in the older age group are predicted to be risky. Income levels are marked at $0K, $40K, $80K.](image-url)
Each split partitions the 2-D space
Depth 1: Split on $x[1]$
Depth 2

```
Y values
- +

Root
18 13

x[1]

x[1] < -0.07
13 3

x[1] >= -0.07
4 11

x[1]

x[1] < -1.66
7 0

x[1] >= -1.66
6 3

x[2] < 1.55
1 11

x[2] >= 1.55
3 0
```
Threshold split caveat

For threshold splits, same feature can be used multiple times
Decision boundaries

- Decision boundaries can be complex!
Overfitting

- Deep decision trees are prone to overfitting
  - Decision boundaries are interpretable but not stable
  - Small change in the dataset leads to big difference in the outcome

- Overcoming Overfitting:
  - Early stopping
    - Fixed length depth
    - Stop if error does not considerably decrease
  - Pruning
    - Grow full length trees
    - Prune nodes to balance a complexity penalty
Predicting probabilities

Loan status:
Safe Risky

Credit?

excellent
9 2
Safe

fair
6 9
Risky

poor
3 1
Safe

P(y = Safe | x) = \frac{3}{3 + 1} = 0.75
Recap

What you can do now:

- Define a decision tree classifier
- Interpret the output of a decision trees
- Learn a decision tree classifier using greedy algorithm
- Traverse a decision tree to make predictions
  - Majority class predictions