Unsupervised Learning and Data Mining
Unsupervised Learning and Data Mining

Clustering
Supervised Learning

- Decision trees
- Artificial neural nets
- K-nearest neighbor
- Support vectors
- Linear regression
- Logistic regression
- ...
Supervised Learning

- F(x): true function (usually not known)
- D: training sample drawn from F(x)

```
57, M, 195, 0, 125, 95, 39, 25, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
78, M, 160, 1, 130, 100, 37, 40, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
69, F, 180, 0, 115, 85, 40, 22, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
18, M, 165, 0, 110, 80, 41, 30, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
54, F, 135, 0, 115, 95, 39, 35, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
84, F, 210, 1, 135, 105, 39, 24, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
89, F, 135, 0, 120, 95, 36, 28, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
49, M, 195, 0, 115, 85, 39, 32, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
40, M, 205, 0, 115, 90, 37, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
74, M, 250, 1, 130, 100, 38, 26, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
77, F, 140, 0, 125, 100, 40, 30, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
...```
Supervised Learning

- **F(x):** true function (usually not known)
- **D:** training sample drawn from F(x)

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 57 | M | 195 | 0 | 125 | 95 | 39 | 25 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 78 | M | 160 | 1 | 130 | 100 | 37 | 40 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 69 | F | 180 | 0 | 115 | 85 | 40 | 22 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | M | 165 | 0 | 110 | 80 | 41 | 30 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 54 | F | 135 | 0 | 115 | 95 | 39 | 35 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 71 | M | 160 | 1 | 130 | 105 | 38 | 20 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **G(x):** model learned from training sample D

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 71 | M | 160 | 1 | 130 | 105 | 38 | 20 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Goal:** $E<(F(x)-G(x))^2>$ is small (near zero) for future samples drawn from F(x)
Supervised Learning

Well Defined Goal:

Learn $G(x)$ that is a good approximation to $F(x)$ from training sample $D$

Know How to Measure Error:

Accuracy, RMSE, ROC, Cross Entropy, ...
Clustering ≠ Supervised Learning
Clustering

≡

Unsupervised Learning
## Supervised Learning

### Train Set:

<table>
<thead>
<tr>
<th>ID</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
<th>Blood Pressure</th>
<th>Heart Rate</th>
<th>Smoke</th>
<th>Alcohol</th>
<th>Exercise</th>
<th>BMI</th>
<th>Diabetes</th>
<th>Heart Disease</th>
<th>Stroke</th>
<th>Blood Pressure</th>
<th>Heart Rate</th>
<th>Smoke</th>
<th>Alcohol</th>
<th>Exercise</th>
<th>BMI</th>
<th>Diabetes</th>
<th>Heart Disease</th>
<th>Stroke</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>M</td>
<td>195</td>
<td>95</td>
<td>39</td>
<td>25</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>78</td>
<td>M</td>
<td>160</td>
<td>110</td>
<td>100</td>
<td>37</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>69</td>
<td>F</td>
<td>180</td>
<td>85</td>
<td>40</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>M</td>
<td>165</td>
<td>80</td>
<td>41</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>F</td>
<td>135</td>
<td>115</td>
<td>39</td>
<td>35</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>84</td>
<td>F</td>
<td>210</td>
<td>135</td>
<td>105</td>
<td>39</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>F</td>
<td>135</td>
<td>120</td>
<td>95</td>
<td>36</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>49</td>
<td>M</td>
<td>195</td>
<td>115</td>
<td>85</td>
<td>39</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>M</td>
<td>205</td>
<td>115</td>
<td>90</td>
<td>37</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>74</td>
<td>M</td>
<td>250</td>
<td>130</td>
<td>100</td>
<td>38</td>
<td>26</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>77</td>
<td>F</td>
<td>140</td>
<td>125</td>
<td>100</td>
<td>40</td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>

### Test Set:

<table>
<thead>
<tr>
<th>ID</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
<th>Blood Pressure</th>
<th>Heart Rate</th>
<th>Smoke</th>
<th>Alcohol</th>
<th>Exercise</th>
<th>BMI</th>
<th>Diabetes</th>
<th>Heart Disease</th>
<th>Stroke</th>
<th>Blood Pressure</th>
<th>Heart Rate</th>
<th>Smoke</th>
<th>Alcohol</th>
<th>Exercise</th>
<th>BMI</th>
<th>Diabetes</th>
<th>Heart Disease</th>
<th>Stroke</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>M</td>
<td>160</td>
<td>110</td>
<td>80</td>
<td>41</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Un-Supervised Learning

#### Train Set:

<table>
<thead>
<tr>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
<th>Cigarettes</th>
<th>Alcohol</th>
<th>Exercise</th>
<th>Sleep</th>
<th>Males</th>
<th>Females</th>
<th>Systolic</th>
<th>Diastolic</th>
<th>HDL</th>
<th>LDL</th>
<th>Triglycerides</th>
<th>Other</th>
<th>Outcome</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>M</td>
<td>195</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>78</td>
<td>M</td>
<td>160</td>
<td>1</td>
<td>130</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>69</td>
<td>F</td>
<td>180</td>
<td>0</td>
<td>115</td>
<td>85</td>
<td>40</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>M</td>
<td>165</td>
<td>0</td>
<td>110</td>
<td>80</td>
<td>41</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>54</td>
<td>F</td>
<td>135</td>
<td>0</td>
<td>115</td>
<td>95</td>
<td>39</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>F</td>
<td>210</td>
<td>1</td>
<td>135</td>
<td>105</td>
<td>39</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>F</td>
<td>135</td>
<td>0</td>
<td>120</td>
<td>95</td>
<td>36</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>74</td>
<td>M</td>
<td>250</td>
<td>1</td>
<td>130</td>
<td>100</td>
<td>38</td>
<td>26</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>84</td>
<td>F</td>
<td>140</td>
<td>0</td>
<td>125</td>
<td>100</td>
<td>40</td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Test Set:

<table>
<thead>
<tr>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
<th>Cigarettes</th>
<th>Alcohol</th>
<th>Exercise</th>
<th>Sleep</th>
<th>Males</th>
<th>Females</th>
<th>Systolic</th>
<th>Diastolic</th>
<th>HDL</th>
<th>LDL</th>
<th>Triglycerides</th>
<th>Other</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>M</td>
<td>160</td>
<td>1</td>
<td>130</td>
<td>105</td>
<td>38</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>?</td>
</tr>
</tbody>
</table>

...
Un-Supervised Learning

Train Set:

- 57, M, 195, 0, 125, 95, 39, 25, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
- 78, M, 160, 1, 130, 100, 37, 40, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
- 69, F, 180, 0, 115, 85, 40, 22, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
- 18, M, 165, 0, 110, 80, 41, 30, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
- 54, F, 135, 0, 115, 95, 39, 35, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0
- 84, F, 210, 1, 135, 105, 39, 24, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
- 89, F, 135, 0, 120, 95, 36, 28, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0
- 49, M, 195, 0, 115, 85, 39, 32, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
- 40, M, 205, 0, 115, 90, 37, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
- 74, M, 250, 1, 130, 100, 38, 26, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0
- 77, F, 140, 0, 125, 100, 40, 30, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Test Set:

- 71, M, 160, 1, 130, 105, 38, 20, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

…
## Un-Supervised Learning

### Data Set:

| Age | Height | Weight | Gender | Activity_1 | Activity_2 | Activity_3 | Activity_4 | Activity_5 | Activity_6 | Activity_7 | Activity_8 | Activity_9 | Activity_10 | Activity_11 | Activity_12 | Activity_13 | Activity_14 | Activity_15 | Activity_16 | Activity_17 | Activity_18 | Activity_19 | Activity_20 |
|-----|--------|--------|--------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 57  | M      | 195    | 95     | 5, 25      | 0, 1, 0   | 0, 0, 0    | 0, 1, 1    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 78  | M      | 160    | 1, 130  | 100, 37    | 40, 1, 0  | 0, 1, 1    | 1, 1, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 69  | F      | 180    | 0, 115  | 35, 40     | 22, 0, 0  | 0, 0, 0    | 0, 0, 0    | 0, 1, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 18  | M      | 165    | 0, 110  | 80, 41     | 30, 0, 0  | 0, 0, 0    | 0, 1, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 54  | F      | 135    | 0, 115  | 39, 35     | 30, 0, 0  | 0, 0, 0    | 0, 1, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 84  | F      | 210    | 1, 135  | 105, 39    | 24, 0, 0  | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 89  | F      | 135    | 0, 120  | 95, 36    | 28, 0, 0  | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 49  | M      | 195    | 0, 115  | 39, 32    | 30, 0, 0  | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 40  | M      | 205    | 0, 115  | 90, 37    | 18, 0, 0  | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 74  | M      | 250    | 1, 130  | 100, 38   | 26, 0, 0  | 0, 0, 0    | 0, 0, 0    | 0, 1, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |
| 77  | F      | 140    | 0, 125  | 100, 30   | 40, 0, 0  | 0, 0, 0    | 0, 0, 0    | 0, 1, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0    | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      | 0, 0, 0      |

...
**Supervised vs. Unsupervised Learning**

**Supervised**
- \( y = F(x) \): true function
- \( D \): labeled training set
- \( D: \{x_i, y_i\} \)
- \( y = G(x) \): model trained to predict labels \( D \)
- Goal:
  \[ E<(F(x) - G(x))^2> \approx 0 \]
- Well defined criteria: Accuracy, RMSE, ...

**Unsupervised**
- Generator: true model
- \( D \): unlabeled data sample
- \( D: \{x_i\} \)
- Learn
- Goal:
- Well defined criteria:
What to Learn/Discover?

- Statistical Summaries
- Generators
- Density Estimation
- Patterns/Rules
- Associations
- Clusters/Groups
- Exceptions/Outliers
- Changes in Patterns Over Time or Location
Goals and Performance Criteria?

- Statistical Summaries
- Generators
- Density Estimation
- Patterns/Rules
- Associations
- Clusters/Groups
- Exceptions/Outliers
- Changes in Patterns Over Time or Location
Clustering
Clustering

- **Given:**
  - Data Set D (training set)
  - Similarity/distance metric/information

- **Find:**
  - Partitioning of data
  - Groups of similar/close items
Similarity?

- Groups of similar customers
  - Similar demographics
  - Similar buying behavior
  - Similar health
- Similar products
  - Similar cost
  - Similar function
  - Similar store
  - ...
- Similarity usually is domain/problem specific
Types of Clustering

- **Partitioning**
  - K-means clustering
  - K-medoids clustering
  - EM (expectation maximization) clustering

- **Hierarchical**
  - Divisive clustering (top down)
  - Agglomerative clustering (bottom up)

- **Density-Based Methods**
  - Regions of dense points separated by sparser regions of relatively low density
Types of Clustering

- **Hard Clustering:**
  - Each object is in one and only one cluster

- **Soft Clustering:**
  - Each object has a probability of being in each cluster
Two Types of Data/Distance Info

- **N-dim vector space representation and distance metric**
  
  D1: 57, M, 195, 0, 125, 95, 39, 25, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  
  D2: 78, M, 160, 1, 130, 100, 37, 40, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  
  ...  
  
  Dn: 18, M, 165, 0, 110, 80, 41, 30, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

  **Distance (D1, D2) = ???.**

- **Pairwise distances between points (no N-dim space)**

  + **Similarity/dissimilarity matrix (upper or lower diagonal)**

  + **Distance:** 0 = near, $\infty$ = far

  + **Similarity:** 0 = far, $\infty$ = near
Agglomerative Clustering

- Put each item in its own cluster (641 singletons)
- Find all pairwise distances between clusters
- Merge the two closest clusters
- Repeat until everything is in one cluster

- Hierarchical clustering
- Yields a clustering with each possible # of clusters
- Greedy clustering: not optimal for any cluster size
Agglomerative Clustering of Proteins
Merging: Closest Clusters

- Nearest centroids
- Nearest medoids
- Nearest neighbors (shortest link)
- Nearest average distance (average link)
- Smallest greatest distance (maximum link)
- Domain specific similarity measure
  - word frequency, TFIDF, KL-divergence, ...
- Merge clusters that optimize criterion after merge
  - minimum mean_point_happiness
Mean Distance Between Clusters

$Mean\_Dist\ (c_1, c_2) = \frac{\sum_{i \in c_1} \sum_{j \in c_2} Dist(i, j)}{\binom{|c_1|}{2} \binom{|c_2|}{2}}$
Minimum Distance Between Clusters

$$\text{Min}\_\text{Dist} \ (c_1, c_2) = \text{MIN} \ (\text{Dist} (i, j))_{i \in c_1, j \in c_2}$$
Mean Internal Distance in Cluster

$$Mean \_Internal \_Dist (c) = \frac{\sum_{i \neq j \in c} Dist (i, j)}{\binom{|c|}{2}}$$
Mean Point Happiness

\[
\eta_{ij} = \begin{cases} 
1 & \text{when cluster}(i) = \text{cluster}(j) \\
0 & \text{when cluster}(i) \neq \text{cluster}(j)
\end{cases}
\]

\[
\text{Mean Happiness} = \frac{\prod_i \prod_j \eta_{ij} \cdot \text{Dist}(i, j)}{\prod_i \prod_j \eta_{ij}}
\]
Recursive Clusters
Recursive Clusters
Recursive Clusters
Mean Point Happiness
Mean Point Happiness
Recursive Clusters + Random Noise
Recursive Clusters + Random Noise
Clustering Proteins
Distance Between Helices

- Vector representation of protein data in 3-D space that gives x,y,z coordinates of each atom in helix

- Use a program developed by chemists (fortran) to convert 3-D atom coordinates into average atomic distances in angstroms between aligned helices

- \[641 \text{ helices} = \frac{641 \times 640}{2} = 205,120 \text{ pairwise distances}\]
Agglomerative Clustering of Proteins
Agglomerative Clustering of Proteins
Agglomerative Clustering of Proteins
Agglomerative Clustering of Proteins

Cluster Size During Agglomerative Clustering

Number of Clusters

Cluster Size

'wsd.641'
Agglomerative Clustering of Proteins

Cluster Purity vs. Cluster Size for PDB Structures

Number of PDB Structures in Cluster
Multidimensional Scaling of helix pairs by RMSD
Agglomerative Clustering

• Greedy clustering
  – once points are merged, never separated
  – suboptimal w.r.t. clustering criterion

• Combine greedy with iterative refinement
  – post processing
  – interleaved refinement
Agglomerative Clustering

- **Computational Cost**
  - $O(N^2)$ just to read/calculate pairwise distances
  - $N-1$ merges to build complete hierarchy
    - scan pairwise distances to find closest
    - calculate pairwise distances between clusters
    - fewer clusters to scan as clusters get larger
  - Overall $O(N^3)$ for simple implementations

- **Improvements**
  - sampling
  - dynamic sampling: add new points while merging
  - tricks for updating pairwise distances
K-Means Clustering

- Inputs: data set and k (number of clusters)
- Output: each point assigned to one of k clusters

- K-Means Algorithm:
  - Initialize the k-means
    - Assign from randomly selected points
    - Randomly or equally distributed in space
  - Assign each point to nearest mean
  - Update means from assigned points
  - Repeat until convergence
K-Means Clustering: Convergence

- **Squared-Error Criterion**

\[
\text{Squared Error} = \sum_{c} \sum_{i \in c} (\text{Dist}(i, \text{mean}(c)))^2
\]

- Converged when SE criterion stops changing
- Increasing K reduces SE - can’t determine K by finding minimum SE
- Instead, plot SE as function of K
K-Means Clustering

- Efficient
  - $K << N$, so assigning points is $O(K \cdot N) < O(N^2)$
  - updating means can be done during assignment
  - usually # of iterations $<< N$
  - Overall $O(N \cdot K \cdot \text{iterations})$ closer to $O(N)$ than $O(N^2)$

- Gets stuck in local minima
  - Sensitive to initialization

- Number of clusters must be pre-specified

- Requires vector space data to calculate means
Soft K-Means Clustering

- Instance of EM (Expectation Maximization)
- Like K-Means, except each point is assigned to each cluster with a probability
- Cluster means updated using weighted average
- Generalizes to Standard_Deviation/Covariance
- Works well if cluster models are known
Soft K-Means Clustering (EM)

- Initialize model parameters:
  + means
  + std_devs
  + ...

- Assign points probabilistically to each cluster

- Update cluster parameters from weighted points

- Repeat until convergence to local minimum
What do we do if we can’t calculate cluster means?

```
-- 1 2 3 4 5 6 7 8 9 10
1 - d d d d d d d d d d
2 - d d d d d d d d d d
3 - d d d d d d d d d d
4 - d d d d d d d d d d
5 - d d d d d d d d d d
6 - d d d d d d d d d d
7 - d d d d d d d d d d
8 - d d d d d d d d d d
9 - d d d d d d d d d d
```
K-Medoids Clustering

\[ \text{Medoid } (c) = \text{pt } \in c \text{ s.t. } \text{MIN} \left( \sum_{i \in c} \text{Dist}(i, pt) \right) \]
K-Medoids Clustering

- Inputs: data set and k (number of clusters)
- Output: each point assigned to one of k clusters
- Initialize k medoids
  - pick points randomly
- Pick medoid and non-medoid point at random
- Evaluate quality of swap
  - Mean point happiness
- Accept random swap if it improves cluster quality
Cost of K-Means Clustering

- n cases; d dimensions; k centers; i iterations
- compute distance each point to each center: O(n*d*k)
- assign each of n cases to closest center: O(n*k)
- update centers (means) from assigned points: O(n*d*k)
- repeat i times until convergence
- overall: O(n*d*k*i)
- much better than O(n^2)-O(n^3) for HAC
- sensitive to initialization - run many times
- usually don’t know k - run many times with different k
- requires many passes through data set
Graph-Based Clustering
Scaling Clustering to Big Databases

- K-means is still expensive: $O(n^d k^I)$
- Requires multiple passes through database
- Multiple scans may not be practical when:
  - database doesn’t fit in memory
  - database is very large:
    + $10^4$-$10^9$ (or more) records
    + $>10^2$ attributes
  - expensive join over distributed databases
Goals

- 1 scan of database
- early termination, on-line, anytime algorithm yields current best answer
Scale-Up Clustering?

- Large number of cases (big n)
- Large number of attributes (big d)
- Large number of clusters (big c)