CSE 255 – Lecture 10
Data Mining and Predictive Analytics

Homework and midterm recap
Assignment 1

CSE 190/255 (fa15) -- Assignment 1 -- Task 1 -- Helpfulness Prediction

Tue 20 Oct 2015

Public Leaderboard - CSE 190/255 (fa15) -- Assignment 1 -- Task 1 -- Helpfulness Prediction

This leaderboard is calculated on approximately 50% of the test data. The final results will be based on the other 50%, so the final standings may be different.

<table>
<thead>
<tr>
<th>#</th>
<th>ΔId</th>
<th>Team Name</th>
<th>Score</th>
<th>Entries</th>
<th>Last Submission UTC (Best - Last Submission)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Baseline</td>
<td>0.63663</td>
<td>1</td>
<td>Mon, 26 Oct 2015 23:40:54</td>
</tr>
<tr>
<td>2</td>
<td>new</td>
<td>kaizhou</td>
<td>0.63663</td>
<td>1</td>
<td>Tue, 27 Oct 2015 21:45:17</td>
</tr>
<tr>
<td>3</td>
<td>new</td>
<td>ShubhamSaini</td>
<td>0.63663</td>
<td>1</td>
<td>Wed, 28 Oct 2015 09:16:58</td>
</tr>
<tr>
<td>4</td>
<td>new</td>
<td>playground</td>
<td>0.63663</td>
<td>1</td>
<td>Wed, 28 Oct 2015 10:25:21</td>
</tr>
</tbody>
</table>

Download raw data
Midterm on Monday!

- 6:40 pm – 7:40 pm
- Closed book – but I’ll provide a similar level of basic info as in the last page of last quarter’s midterm
- I’ll run extra office hours from 9:30-11:30 tomorrow, though the TAs will have regular office hours on Friday
CSE 255 – Lecture 10
Data Mining and Predictive Analytics

Week 1 recap
Supervised versus unsupervised learning

**Learning** approaches attempt to model data in order to solve a problem.

**Unsupervised learning** approaches find patterns/relationships/structure in data, but are not optimized to solve a particular predictive task.

- E.g. PCA, community detection

**Supervised learning** aims to directly model the relationship between input and output variables, so that the output variables can be predicted accurately given the input.

- E.g. linear regression, logistic regression
Linear regression assumes a predictor of the form

\[ X \theta = y \]

(\text{or } Ax = b \text{ if you prefer})
Regression diagnostics

Mean-squared error (MSE)

$$\frac{1}{N} \| y - X \theta \|_2^2$$

$$= \frac{1}{N} \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2$$
Representing the month as a feature

How would you build a feature to represent the month?

\[ \text{rating} \sim \theta_0 + \theta \times \text{month} \]

\[ \text{jan} = 1 \quad \text{dec} = 12 \]
Representing the month as a feature

\[ \text{Jan} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

\[ \text{Nov} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

\[ \text{Dec} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

\[ (X'X)^{-1} \]
Occam’s razor

“Among competing hypotheses, the one with the fewest assumptions should be selected”

(image from personalspirituality.net)
Regularization is the process of penalizing model complexity during training.

\[
\arg \min_\theta = \frac{1}{N} \| y - X\theta \|^2_2 + \lambda \| \theta \|^2_2
\]

How much should we trade-off accuracy versus complexity?
A validation set is constructed to “tune” the model’s parameters

- Training set: used to optimize the model’s parameters
- Test set: used to report how well we expect the model to perform on unseen data
- Validation set: used to tune any model parameters that are not directly optimized
Regularization
A few “theorems” about training, validation, and test sets

- The training error **increases** as lambda **increases**
- The validation and test error are at least as large as the training error (assuming infinitely large random partitions)
- The validation/test error will usually have a “sweet spot” between under- and over-fitting
Classification

Will I **purchase** this product?  
(Yes)

Will I **click on** this ad?  
(No)
Classification

What animal appears in this image?

(mandarin duck)
What are the **categories** of the item being described?

(book, fiction, philosophical fiction)

From *Booklist*

Houellebecq's deeply philosophical novel is about an alienated young man searching for happiness in the computer age. Bored with the world and too weary to try to adapt to the foibles of friends and coworkers, he retreats into himself, descending into depression while attempting to analyze the passions of the people around him. Houellebecq uses his nameless narrator as a vehicle for extended exploration into the meanings and manifestations of love and desire in human interactions. Ironically, as the narrator attempts to define love in increasingly abstract terms, he becomes less and less capable of experiencing that which he is so desperate to understand. Intelligent and well written, the short novel is a thought-provoking inspection of a generation's confusion about all things sexual. Houellebecq captures precisely the cynical disillusionment of disaffected youth. *Bonnie Johnston --This text refers to an out of print or unavailable edition of this title.*
Linear regression assumes a predictor of the form $X\theta = y$.
Regression vs. classification

But how can we predict **binary** or **categorical** variables?

\[ f(\text{data}) \rightarrow \text{labels} \]

\{0, 1\}, \{\text{True, False}\}

\{1, \ldots, N\}
We’ll attempt to build **classifiers** that make decisions according to rules of the form

$$y = X_i \cdot \theta$$

$$y_i = \begin{cases} 
1 & \text{if } X_i \cdot \theta > 0 \\
0 & \text{otherwise}
\end{cases}$$
1. Naïve Bayes
Assumes an **independence** relationship between the features and the class label and “learns” a simple model by counting

2. Logistic regression
Adapts the **regression** approaches we saw last week to binary problems

3. Support Vector Machines
Learns to classify items by finding a hyperplane that separates them
(feature$_i$ $\perp feature$_j$ | label)

\[ p(feature$_i$, feature$_j$ | label) = p(feature$_i$ | label)p(feature$_j$ | label) \]
Naïve Bayes (2 slide summary)

\[
p(y | x) = \frac{p(y) p(x | y)}{p(x)}
\]

\[
p(y) \prod p(x_i | y) \geq 12
\]

\[
\frac{p(y) \prod p(x_i | y)}{p(y) \prod p(x_i | y)} \geq 12
\]
Double-counting: naïve Bayes vs Logistic Regression

**Q:** What would happen if we trained two regressors, and attempted to “naively” combine their parameters?

\begin{align*}
\text{no. of pages} &= \alpha + \beta_1 \cdot \delta(\text{mentions wizards}) \\
\text{no. of pages} &= \alpha + \beta_2 \cdot \delta(\text{mentions witches}) \\
\text{no. of pages} &= \alpha + \beta_1 \cdot \delta(\text{mentions wizards}) + \beta_2 \cdot \delta(\text{mentions witches})
\end{align*}
Logistic regression

**Sigmoid function:** \( \sigma(t) = \frac{1}{1 + e^{-t}} \)
Logistic regression

Training:

\[ X_i \cdot \theta \] should be maximized when \( y_i \) is positive and minimized when \( y_i \) is negative

\[
\arg \max \theta \prod_i \delta(y_i = 1)p_\theta(y_i | X_i) + \delta(y_i = 0)(1 - p_\theta(y_i | X_i))
\]

\( \delta(\text{arg}) = 1 \) if the argument is true, = 0 otherwise
Logistic regression

\[ \arg \max_{\theta} \prod_i \delta(y_i = 1)p_{\theta}(y_i | X_i) + \delta(y_i = 0)(1 - p_{\theta}(y_i | X_i)) \]

1. \( \log \ \text{obj} \)
2. subtract regularizer
3. \( \frac{\partial \text{obj}}{\partial \theta} \)
4. climb the hill
Q: Where would a logistic regressor place the decision boundary for these features?
Logistic regression

• Logistic regressors don’t optimize the number of “mistakes”
• No special attention is paid to the “difficult” instances – every instance influences the model
• But “easy” instances can affect the model (and in a bad way!)
• How can we develop a classifier that optimizes the number of mislabeled examples?
Support Vector Machines

\[ \theta x - \alpha = 1 \]
\[ \theta x - \alpha = 0 \]
\[ \theta x - \alpha = -1 \]

\[ \arg \min_{\theta, \alpha} \frac{1}{2} \| \theta \|_2^2 \]

such that
\[ \forall i : y_i (\theta \cdot X_i - \alpha) \geq 1 \]
The classifiers we’ve seen in Week 2 all attempt to make decisions by associating weights (theta) with features (x) and classifying according to

$$y_i = \begin{cases} 
1 & \text{if } X_i \cdot \theta > 0 \\
0 & \text{otherwise}
\end{cases}$$
Summary

• **Naïve Bayes**

  • Probabilistic model (fits $p(label|data)$)
  • Makes a conditional independence assumption of the form $(feature_i \perp \perp feature_j|label)$ allowing us to define the model by computing $p(feature_i|label)$ for each feature
  • Simple to compute just by counting

• **Logistic Regression**

  • Fixes the “double counting” problem present in naïve Bayes

• **SVMs**

  • Non-probabilistic: optimizes the classification error rather than the likelihood
Which classifier is best?

1. When data are highly imbalanced

   If there are far fewer positive examples than negative examples we may want to assign additional weight to negative instances (or vice versa)

   e.g. will I purchase a product? If I purchase 0.00001% of products, then a classifier which just predicts “no” everywhere is 99.99999% accurate, but not very useful.
Which classifier is best?

2. When mistakes are more costly in one direction
   False positives are nuisances but false negatives are disastrous (or vice versa)

   e.g. which of these bags contains a weapon?
Which classifier is best?

3. When we only care about the “most confident” predictions
e.g. does a relevant result appear among the first page of results?
Evaluating classifiers

decision boundary

negative positive
Evaluating classifiers

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

- **True Positive**: Correctly predicted true
- **False Positive**: Incorrectly predicted true
- **False Negative**: Incorrectly predicted false
- **True Negative**: Correctly predicted false

**Classification accuracy**

\[
\text{Classification accuracy} = \frac{\text{correct predictions}}{\#\text{predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}
\]

**Error rate**

\[
\text{Error rate} = \frac{\text{incorrect predictions}}{\#\text{predictions}} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}
\]
• Linear classification – know what the different classifiers are and when you should use each of them. What are the advantages/disadvantages of each.
• Know how to evaluate classifiers – what should you do when you care more about false positives than false negatives etc.
Why dimensionality reduction?

Goal: take **high-dimensional** data, and describe it compactly using a small number of dimensions.

Assumption: Data lies (approximately) on some **low-dimensional** manifold

(a few dimensions of opinions, a small number of topics, or a small number of communities)
Principal Component Analysis

$\mathbf{X}$

$\mathbf{Y}$

$\varphi$

rotate

$\varphi^T$

un-rotate

discard lowest-variance dimensions

$\mathbf{Y}, 0$
Principal Component Analysis

Construct such vectors from 100,000 patches from real images and run PCA:
Principal Component Analysis

• We want to find a low-dimensional representation that best compresses or “summarizes” our data
• To do this we’d like to keep the dimensions with the highest variance (we proved this), and discard dimensions with lower variance. Essentially we’d like to capture the aspects of the data that are “hardest” to predict, while discard the parts that are “easy” to predict
• This can be done by taking the eigenvectors of the covariance matrix (we didn’t prove this, but it’s right there in the slides)
Q: What would PCA do with this data?
A: Not much, variance is about equal in all dimensions.
But: The data are highly **clustered**

Idea: can we compactly describe the data in terms of **cluster memberships**?
K-means Clustering

1. Input is still a matrix of features:

\[
X = \begin{pmatrix}
5 & 3 & \cdots & 1 \\
4 & 2 & 1 \\
3 & 1 & 3 \\
2 & 2 & 4 \\
1 & 5 & 2 \\
\vdots & \vdots & \vdots \\
1 & 2 & \cdots & 1
\end{pmatrix}
\]

2. Output is a list of cluster "centroids":

\[
\text{centroids} = \begin{pmatrix}
1.1 & 2.1 \\
3.5 & 1.8 \\
0.2 & 0.1 \\
3.0 & -0.3
\end{pmatrix}
\]

3. From this we can describe each point in X by its cluster membership:

\[
f = [0,0,1,0]
\quad\quad
f = [0,0,0,1]
\]

\[
Y = (1, 2, 4, 3, 4, 2, 4, 2, 2, 3, 3, 2, 1, 1, 3, \ldots, 2)
\]
K-means Clustering

Greedy algorithm:

1. Initialize $C$ (e.g. at random)
2. Do
3. Assign each $X_i$ to its nearest centroid
4. Update each centroid to be the mean of points assigned to it
5. While (assignments change between iterations)

(also: reinitialize clusters at random should they become empty)
**Q:** What if our clusters are **hierarchical**?

**A:** We’d like a representation that encodes that points have *some features* in common but not others.
Hierarchical (agglomerative) clustering works by gradually fusing clusters whose points are closest together.

Assign every point to its own cluster:
Clusters = [[1],[2],[3],[4],[5],[6],…,[N]]

While len(Clusters) > 1:
  Compute the center of each cluster
  Combine the two clusters with the nearest centers
1. Connected components

Define communities in terms of sets of nodes which are reachable from each other

- If \( a \) and \( b \) belong to a **strongly connected component** then there must be a path from \( a \rightarrow b \) and a path from \( b \rightarrow a \)
- A **weakly connected component** is a set of nodes that **would be** strongly connected, if the graph were undirected
2. Graph cuts

What is the **Ratio Cut** cost of the following two cuts?

\[
\text{Ratio Cut}(\ldots) = \frac{1}{2} \left( \frac{3}{33} + \frac{3}{1} \right) = 1.54545
\]

\[
\text{Ratio Cut}(\ldots) = \frac{1}{2} \left( \frac{9}{16} + \frac{9}{18} \right) = 0.53125
\]
3. Clique percolation

- Clique percolation searches for “cliques” in the network of a certain size (K). Initially each of these cliques is considered to be its own community.
- If two communities share a (K-1) clique in common, they are merged into a single community.
- This process repeats until no more communities can be merged.

1. Given a clique size K
2. Initialize every K-clique as its own community
3. While (two communities I and J have a (K-1)-clique in common):
4. Merge I and J into a single community
Week 3

- Clustering & Community detection – understand the basics of the different algorithms
  - Given some features, know when to apply PCA vs. K-means vs. hierarchical clustering
  - Given some networks, know when to apply clique percolation vs. graph cuts vs. connected components
Midterm on Monday!

- Similar in format to last quarter’s midterm
- Somewhat harder since (1) this is a graduate class, and (2) since you have a practice exam to work from
- Worth a little bit less (30% vs 25%)
CSE 255 – Lecture 10
Data Mining and Predictive Analytics

Last quarter’s midterm
Section 1: Regression

Q1: Restaurants & ratings (10 marks)

Suppose we collected the following data about restaurants from *Yelp!*

<table>
<thead>
<tr>
<th>Name</th>
<th>Average Rating</th>
<th>Takes reservations?</th>
<th>Take-out?</th>
<th>Price</th>
<th>Good for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oceana Coastal Kitchen</td>
<td>4.5</td>
<td>Yes</td>
<td>No</td>
<td>$$$</td>
<td>Breakfast</td>
</tr>
<tr>
<td>Beyer Deli</td>
<td>5.0</td>
<td>No</td>
<td>Yes</td>
<td>$</td>
<td>Lunch</td>
</tr>
<tr>
<td>Werewolf</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Brunch</td>
</tr>
<tr>
<td>C Level</td>
<td>4.0</td>
<td>No</td>
<td>Yes</td>
<td>$$</td>
<td>Lunch, Dinner</td>
</tr>
<tr>
<td>Cucina Urban</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Dinner</td>
</tr>
</tbody>
</table>

and that from this data we want to estimate

\[
\text{av. rating} \approx \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]
\]
Last quarter’s midterm

<table>
<thead>
<tr>
<th>Name</th>
<th>Average Rating</th>
<th>Takes reservations?</th>
<th>Take-out?</th>
<th>Price</th>
<th>Good for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oceana Coastal Kitchen</td>
<td>4.5</td>
<td>Yes</td>
<td>No</td>
<td>$$$</td>
<td>Breakfast</td>
</tr>
<tr>
<td>Beyer Deli</td>
<td>5.0</td>
<td>No</td>
<td>Yes</td>
<td>$</td>
<td>Lunch</td>
</tr>
<tr>
<td>Werewolf</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Brunch</td>
</tr>
<tr>
<td>C Level</td>
<td>4.0</td>
<td>No</td>
<td>Yes</td>
<td>$$</td>
<td>Lunch, Dinner</td>
</tr>
<tr>
<td>Cucina Urban</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Dinner</td>
</tr>
</tbody>
</table>

and that from this data we want to estimate

\[ \text{av. rating} \approx \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}] \]

1. What is the average rating across all restaurants (1 mark)? \( A: \boxed{4.5} \)

2. What is the Mean Squared Error of the predictor that just predicts the average rating for all items (1 mark)? \( A: \boxed{\left( \frac{1}{5} \cdot \frac{1}{2} + \frac{1}{5} \cdot \frac{1}{2} \right) / 5 = \frac{1}{10}} \)

3. Suppose we’d like to write down the above expression for the rating in the form \( y \approx X\theta \). Complete the following equation to do so:

\[
\begin{bmatrix}
4.5 \\
5 \\
4.5 \\
4.5
\end{bmatrix}
\approx
\begin{bmatrix}
1 & 1 & 0 & 3
\end{bmatrix}
\theta
\]

(1 mark)

4. In the expression \( y \approx X\theta \), which term encodes the labels, which term encodes the features, and which term encodes the parameters (1 mark)? \( \text{labels: } y \text{ features: } X \text{ parameters: } \Theta \)
Last quarter’s midterm

<table>
<thead>
<tr>
<th>Name</th>
<th>Average Rating</th>
<th>Takes reservations?</th>
<th>Take-out?</th>
<th>Price</th>
<th>Good for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oceana Coastal Kitchen</td>
<td>4.5</td>
<td>Yes</td>
<td>No</td>
<td>$$$</td>
<td>Breakfast</td>
</tr>
<tr>
<td>Beyer Deli</td>
<td>5.0</td>
<td>No</td>
<td>Yes</td>
<td>$</td>
<td>Lunch</td>
</tr>
<tr>
<td>Werewolf</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Brunch</td>
</tr>
<tr>
<td>C Level</td>
<td>4.0</td>
<td>No</td>
<td>Yes</td>
<td>$$</td>
<td>Lunch, Dinner</td>
</tr>
<tr>
<td>Cucina Urban</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Dinner</td>
</tr>
</tbody>
</table>

and that from this data we want to estimate

\[
\text{av. rating} \simeq \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]
\]

5. Suppose that after fitting our model for the rating we obtain \(\theta = [7, 0.5, -1, -1]^T\). What is the interpretation of \(\theta_0 = 7\) in this expression (1 mark)?

A:

6. What is the interpretation of \(\theta_3 = -1\) (1 mark)?

A: \text{Price up by } 1$, \text{ calm down by } 1
Last quarter’s midterm

<table>
<thead>
<tr>
<th>Name</th>
<th>Average Rating</th>
<th>Takes reservations?</th>
<th>Take-out?</th>
<th>Price</th>
<th>Good for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oceana Coastal Kitchen</td>
<td>4.5</td>
<td>Yes</td>
<td>No</td>
<td>$$$</td>
<td>Breakfast</td>
</tr>
<tr>
<td>Beyer Deli</td>
<td>5.0</td>
<td>No</td>
<td>Yes</td>
<td>$</td>
<td>Lunch</td>
</tr>
<tr>
<td>Werewolf</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Brunch</td>
</tr>
<tr>
<td>C Level</td>
<td>4.0</td>
<td>No</td>
<td>Yes</td>
<td>$$</td>
<td>Lunch, Dinner</td>
</tr>
<tr>
<td>Cucina Urban</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Dinner</td>
</tr>
</tbody>
</table>

and that from this data we want to estimate

\[
\text{av. rating} \simeq \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]
\]

9. Suppose you wanted to incorporate the ‘Good for’ field (the last column of the above table) into your model. How would you represent the features in order to do so? Answer this by writing down the model you would use:

\[
\text{av. rating} \simeq \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}] + \text{A:}
\]

and by completing the feature matrix using your representation:

\[
X = \begin{bmatrix}
1 & 1 & 0 & 3 \\
\end{bmatrix}
\]

(2 marks)
Q2: Training, testing, & model selection (6 marks)

Suppose we are training regressors to minimize the regularized Mean Squared Error

\[
\sum_{(x,y) \in \text{train}} \frac{1}{|\text{train}|} (y - x \cdot \theta)^2 + \lambda \|	heta\|^2.
\]

10. Suppose that we fit some model for \( \lambda \in \{0.01, 0.1, 1, 10, 100, 1000\} \) and obtain the following performance on the training and validation sets:

Which value of \( \lambda \) would you select based on the results above (1 mark)? \[ \lambda = \]

11. Answer the following questions about training, validation, and test sets:

(a) What is the role of a validation set (1 mark)?

A:

(b) How does the training error typically vary with \( \lambda \) (1 mark)?

A:

(c) What is meant by under/over fitting? Which values of \( \lambda \) in the above figure correspond to maximum over/under fitting (1 mark)?

A:
12. Further suppose that we consider two different feature representations (model X and model Y), and two different regularization parameters ($\lambda = 1$ and $\lambda = 10$) and obtain the following results on the training and validation sets:

<table>
<thead>
<tr>
<th>model</th>
<th>training error</th>
<th>validation error</th>
</tr>
</thead>
<tbody>
<tr>
<td>model X, $\lambda = 1$</td>
<td>23.34</td>
<td>?</td>
</tr>
<tr>
<td>model X, $\lambda = 10$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>model Y, $\lambda = 1$</td>
<td>?</td>
<td>18.32</td>
</tr>
<tr>
<td>model Y, $\lambda = 10$</td>
<td>25.98</td>
<td>?</td>
</tr>
</tbody>
</table>

(‘?’ indicates an unknown value).

Assuming that our training/validation/test sets are large, independent samples, is the above information enough to determine which model and which value of $\lambda$ we would expect to yield the best performance on the test set? If so, which model and which value of $\lambda$ would you expect to perform best and why? Explain your answer (2 marks).

A:
Q3: Fantasy novels (6 marks)

Suppose we have a database consisting of the following books:

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Circle of Sorcerers</td>
<td>Fantasy</td>
<td>True</td>
</tr>
<tr>
<td>Knights: The Eye of Divinity</td>
<td>Fantasy</td>
<td>True</td>
</tr>
<tr>
<td>Superman/Batman: Sorcerer Kings</td>
<td>Graphic Novel</td>
<td></td>
</tr>
<tr>
<td>In the Blood</td>
<td>Mystery</td>
<td></td>
</tr>
<tr>
<td>Remains of the Day</td>
<td>Literature &amp; Fiction</td>
<td></td>
</tr>
<tr>
<td>Blood Song</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td>Flame Moon</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td>The Book of the Sword: A History of Daggers</td>
<td>History</td>
<td></td>
</tr>
<tr>
<td>A Storm of Swords</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td>The Storm Book</td>
<td>Children's</td>
<td></td>
</tr>
</tbody>
</table>

Further, suppose we are given the following classifier to classify Fantasy vs. non-Fantasy books:

```python
if (Title contains 'Sorcerer' or 'Blood' or 'Knights' or 'Moon' or 'Storm'):
    return True
else:
    return False
```

13. What are the predictions made by this classifier? Write your answers in the last column of the table above (1 mark).

14. Of these predictions, what is the number of true positives, true negatives, false positives, and false negatives (1 mark)?

<table>
<thead>
<tr>
<th></th>
<th>true positive</th>
<th>true negative</th>
<th>false positive</th>
<th>false negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15. What are the true positive rate (hint: TP / (TP + FN)), true negative rate, and balanced error rate (1 mark)?

<table>
<thead>
<tr>
<th>true positive rate</th>
<th>true negative rate</th>
<th>balanced error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\frac{2}{2+3} = \frac{2}{5} = \frac{1}{2} \left(1 + \frac{2}{5}\right)
\]

16. In class we saw three approaches to classification: naïve Bayes, logistic regression, and support-vector machines. Describe one benefit of each approach compared to the other two (3 marks).

- naïve Bayes:
  -

- logistic regression:
  -

- SVM:
Section 3: Communities & clustering

Q4: Algorithms for community detection, dimensionality reduction, and clustering

Recall three algorithms we saw in class to detect communities: connected components, ratio cut, and clique percolation (pseudocode is given as Algorithms 1, 2, and 3 at the end of the test).

17. Identify the communities that would be produced on the graphs below using these three algorithms. Circle the communities directly in the space below (some more graphs are provided on the final page in case you need to re-write your answer):

| Connected components (1 mark) | Ratio cut (2 communities) (1 mark) | Clique percolation ($k = 3$) (1 mark) | Clique percolation ($k = 2$) (1 mark) |
18. Suppose we are given the following 2-dimensional data $X$, and wish to cluster it so as to minimize the reconstruction error ($\sum_{x \in X} \|x - \bar{x}\|^2$). Separate the points into three clusters such that the reconstruction error (when replacing each point by its cluster centroid) would be minimized. Draw the clusters directly in the space below (1 mark):

![Cluster Diagram]

19. By replacing each point with one of the three centroids above, we have effectively ‘compressed’ the data, since each (2-d) point is replaced by a (1-d) integer. Another way to compress the data would be to perform Principal Component Analysis, and discard the lowest variance dimension, which would also result in a 1-d representation of the data. Out of these two possible compressed representations, which one would result in the lower reconstruction error on the above data, and why (1 mark)?

A:

20. In class we saw hierarchical clustering, an algorithm that works by successively joining clusters whose centroids are nearest. Psuedocode is given in Algorithm 4 over the page.

Suppose you are given the following set of points:

![Hierarchy Diagram]

<table>
<thead>
<tr>
<th>Step</th>
<th>Clusters merged</th>
<th>List of clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(initialization)</td>
<td>{a}, {b}, {c}, {d}, {e}, {f}, {g}</td>
</tr>
<tr>
<td>1</td>
<td>{a} merges with {b}</td>
<td>{a, b}, {c}, {d}, {e}, {f}, {g}</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>{a, b, c, d, e, f, g}</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If we were to perform hierarchical clustering on this data, in what order would the clusters be joined? Answer this question by completing the table above (2 marks).
No reduction after degree 1 (HW1/wk1)

\[ \text{train} = X[:, 25000] \]
\[ \text{test} = X[25000:] \]
Train vs. lambda (Classification, HW1/wk2)
PCA reconstruction error
PCA reconstruction error
PCA reconstruction error
CSE 255 – Lecture 10
Data Mining and Predictive Analytics

Misc. questions
Representing the **day** as a feature

How would you build a feature to represent the time of **day**?
Representing the day as a feature

How would you build a feature to represent the time of day?
• Suppose we have a linear regression model to predict college GPA
• One of the features of this model encodes whether a student owns a car
• The fitted model looks like:

\[ y = ... - 0.4[\text{owns a car}] + ... \]

Conclusion: “The GPA of the average student who owns a car is 0.4 lower than that of the average student”

Q: is this conclusion reasonable?