

Advanced Text Analysis for Business (IDS-566)

Lecture 4

Feb 16, 2018

Course Overview

- Instructor
 - Ehsan M. Ardehaly PhD, ehsan@uic.edu
 - Office hours: 4:45 - 5:45 pm F, BLC L270
 - Teacher assistant: 4:00 - 5:00 pm W, BLC L270
- Objectives:
 - Text mining
 - Applications for business decisions
 - Study of machine learning concepts
 - Design and implementation of text mining approaches

Assignments-2

- Grade: 20%
- Sentiment analysis
- Due date: 2/25/2018
- Submission:
 - Notebook (code + analysis) → PDF
 - Word document with code as an appendix → PDF

Agenda

Logistic regression

- MLE, cost function, regularization, multi-class

Gradient descent:

- Solving logistic regression

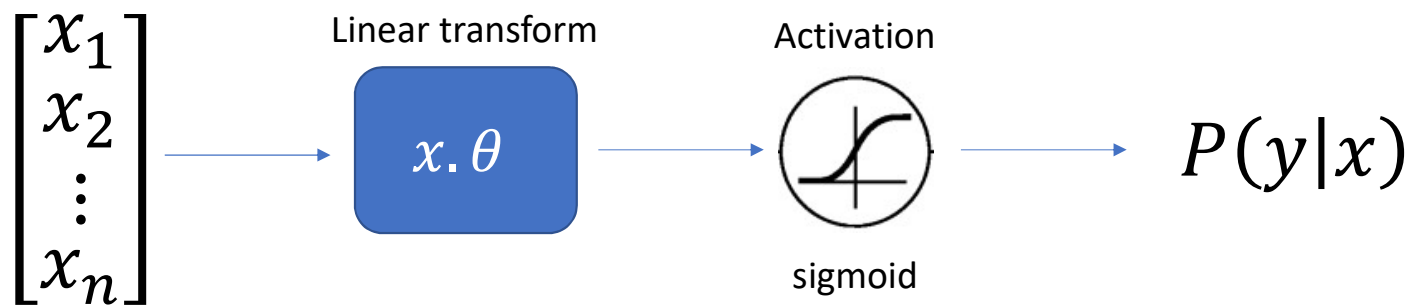
Metrics:

- Confusion matrix, precision, recall, F1

Applications:

- Sentiment analysis, demographic classification

Logistic Regression



Likelihood

- Conditional probability of i-th sample to be positive:

- $P(y_i = 1|X_i) = \sigma(X_i \cdot \theta)$ \leftarrow sigmoid function

- Conditional probability of i-th sample to be negative:

- $P(y_i = 0|X_i) = 1 - \sigma(X_i \cdot \theta)$

- Likelihood:

- $L(X; \theta) = \prod_i P(y_i = 1|X_i)^{y_i} P(y_i = 0|X_i)^{1-y_i}$

Likelihood

- $L(X; \theta) = \prod_i P(y_i = 1|X_i)^{y_i} P(y_i = 0|X_i)^{1-y_i}$
- Example:
- Predicted probabilities for positive samples: .9, .8, .2
- Predicted probabilities for negative samples: .6, .1
- $L = (.9^1 \times .1^0)(.8^1 \times .2^0)(.2^1 \times .8^0)(.6^0 \times .4^1)(.1^0 \times .9^1)$
- $L = .9 \times .8 \times .2 \times .4 \times .9 = 0.05184$

Maximum Likelihood Estimation (MLE)

- Find θ that maximizes likelihood:

- $L(X; \theta) = \prod_i P(y_i = 1|X_i)^{y_i} P(y_i = 0|X_i)^{1-y_i}$

- Or maximize log-likelihood:

- $l(X; \theta) = \sum_i (y_i \log P(y_i = 1|X_i) + (1 - y_i) \log P(y_i = 0|X_i))$

- $l(X; \theta) = \sum_i (y_i \log \sigma(X_i \cdot \theta) + (1 - y_i) \log(1 - \sigma(X_i \cdot \theta)))$

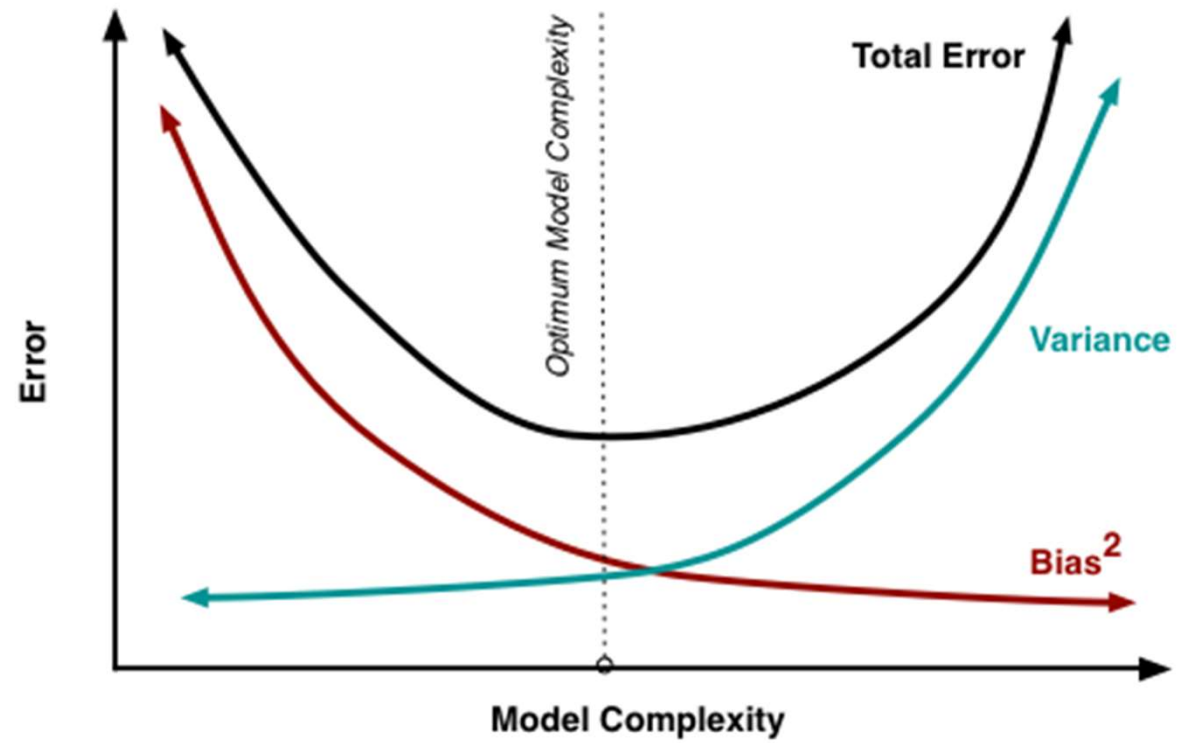
Cost function

- Maximize log-likelihood
- Or
- Minimize negative log-likelihood (cost function):
- $J(\theta) = -\sum_i (y_i \log \sigma(X_i \cdot \theta) + (1 - y_i) \log(1 - \sigma(X_i \cdot \theta)))$
- Example:
 - $L = .9 \times .8 \times .2 \times .4 \times .9 = 0.05184$
 - $l = \log .05184 = -2.96$
 - $J = 2.96$

Training logistic regression

- Creating the cost function:
 - Negative log likelihood
 - $J(\theta) = -l(\theta) = -\sum_i (y_i \log \sigma(X_i \cdot \theta) + (1 - y_i) \log(1 - \sigma(X_i \cdot \theta)))$
- Find θ which minimizes the cost function:
 - $\theta = \underset{\theta}{\operatorname{Argmin}} J(\theta)$

Bias vs. Variance



How to control the variance?

- Adding the regularization term.
- Penalize the cost function for high variation.
- No impact on hypothesis function.

L2 regularization

- Penalize the magnitude of model parameters:
 - $\|\theta\|_2^2 = \sum_{i=1}^n \theta_i^2$
 - $J(\theta) = -l(\theta) + \lambda \|\theta\|_2^2$
- Regularization strength:
 - Higher $\lambda \rightarrow$ Low variation
 - Lower $\lambda \rightarrow$ High variation

L1 regularization

- Penalize the absolute deviation of parameters:
 - $\|\theta\|_1 = \sum_{i=1}^n |\theta_i|$
 - $J(\theta) = -l(\theta) + \lambda \|\theta\|_1$
- Regularization strength:
 - Higher $\lambda \rightarrow$ Low variation
 - Lower $\lambda \rightarrow$ High variation

Elastic net regularization

- Take advantage of both regularization
- Combine both L1 and L2:
 - $J(\theta) = -l(\theta) + \alpha\|\theta\|_1 + \beta\|\theta\|_2^2$
- Hard to tune.

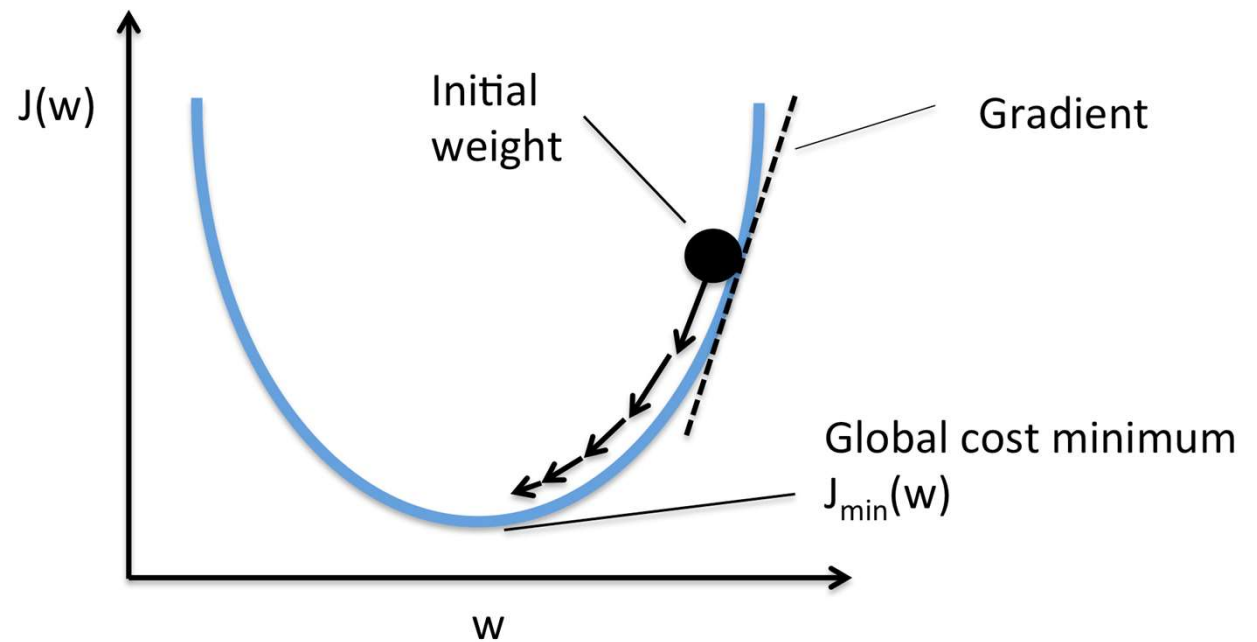
L1 vs L2

- L1:
 - Shrinks less important features.
 - Good for feature selection.
- L2:
 - Often have better result.

Gradient descent algorithm

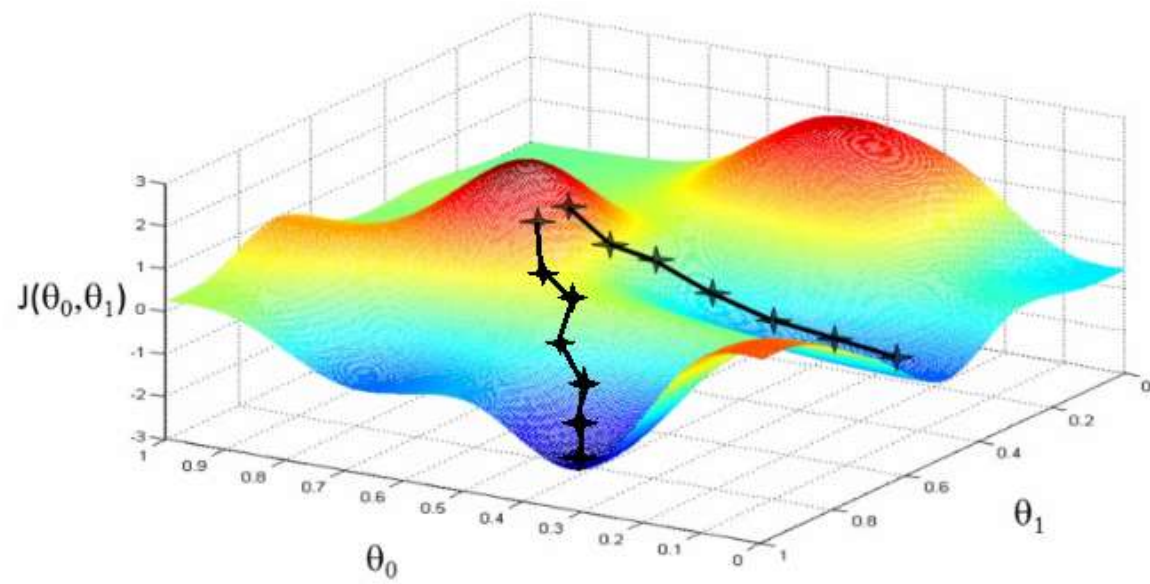
- An iterative algorithm
- Finding the minimum of a function:
 - Local minimum \leftarrow non-convex function
 - Global minimum \leftarrow convex function
- Starts with a random initialization.
- Takes step to the negative of the gradient.

Convex-GD



https://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/

Non-convex
GD



<http://blog.datumbox.com/tuning-the-learning-rate-in-gradient-descent/>

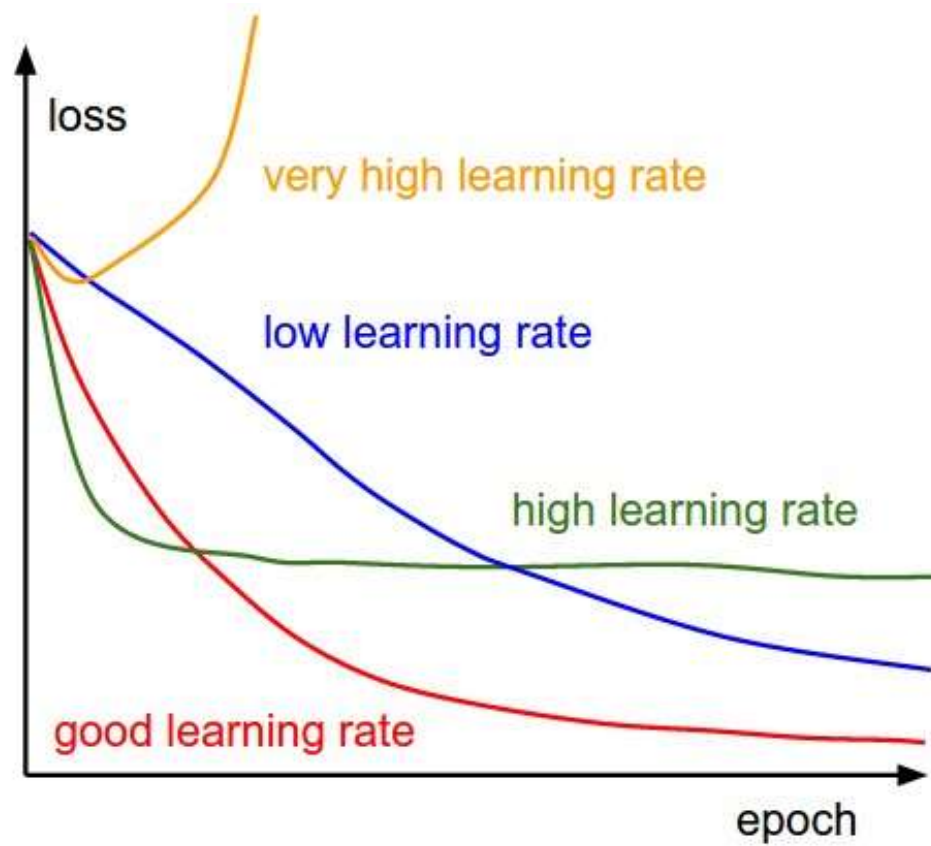
Gradient Descent

- Input:
 - Lost function: $J(\theta)$
 - Learning rate: η
- Initialize with random small parameters: θ_0
- Descent to local minimum:
 - $\theta_{n+1} = \theta_n - \eta \nabla J(\theta_n)$
 - Stopping condition

Gradient

- Gradient is a vector (with size of parameters)
- Partial deviation:
 - $\nabla J_i = \frac{\partial J}{\partial \theta_i}$

Learning rate



Gradient descent for Logistic regression

- $J(\theta) = -\sum_i (y_i \log \sigma(X_i \cdot \theta) + (1 - y_i) \log(1 - \sigma(X_i \cdot \theta)))$
- Gradient:
 - $\nabla J(\theta) = \sum_i X_i (\sigma(X_i \cdot \theta) - y_i)$
- Gradient descent:
 - $\theta_{n+1} = \theta_n - \eta \sum_i X_i (\sigma(X_i \cdot \theta_n) - y_i)$

Multi-class logistic regression

- Parameters for each class: (m classes)

- $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(m)}$

- Hypothesis:

- Softmax function

- $$P(y_i = k | X_i) = \frac{\exp(X_i \cdot \theta^{(k)})}{\sum_{j=1}^m \exp(X_i \cdot \theta^{(j)})}$$

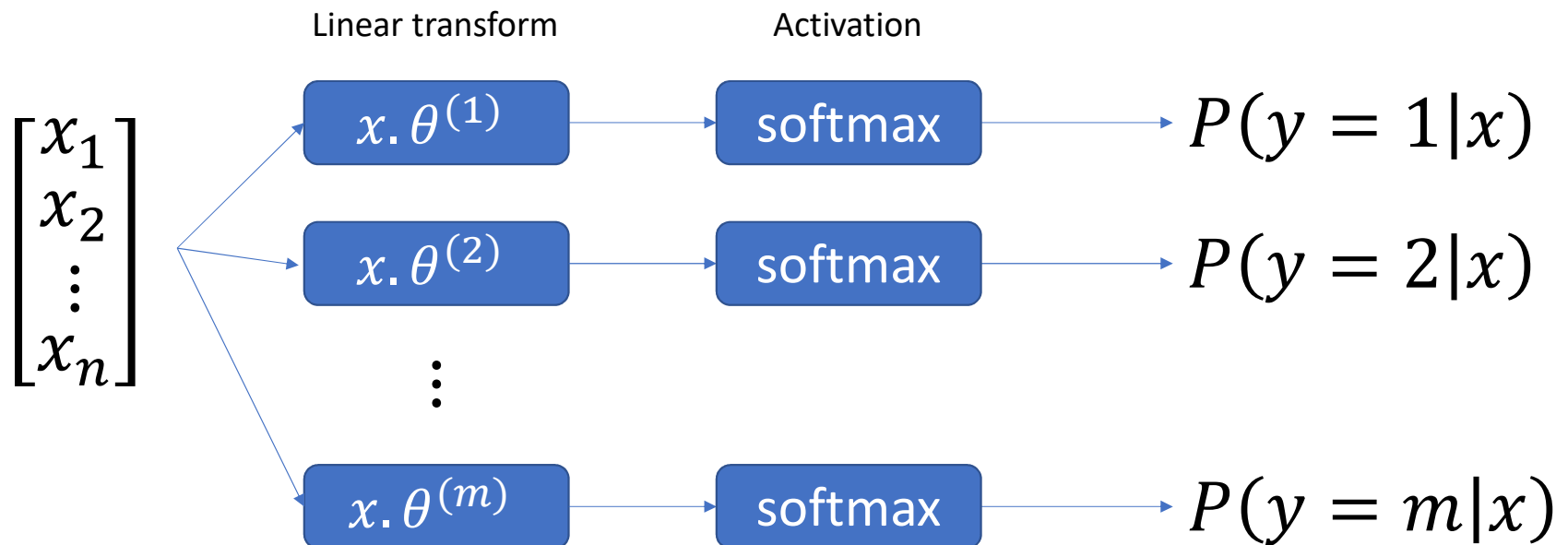
Multi-class logistic regression

- Softmax hypothesis:

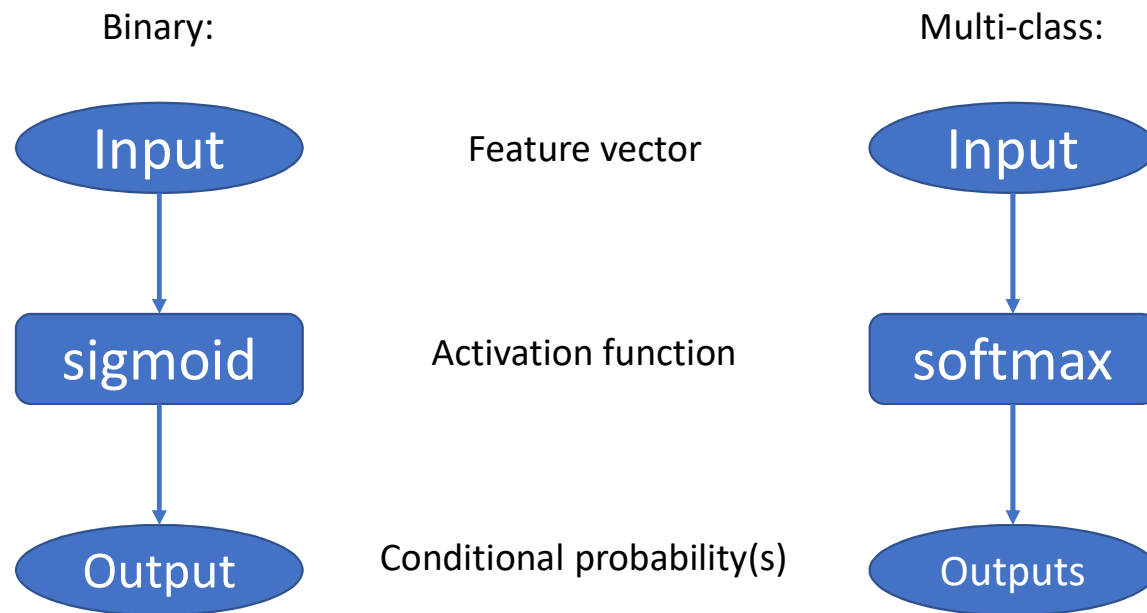
$$\bullet P(y_i = k | X_i) = \frac{\exp(X_i \cdot \theta^{(k)})}{\sum_{j=1}^m \exp(X_i \cdot \theta^{(j)})}$$

- How to solve:
 - Create the lost function as negative log likelihood.
 - Use gradient descent algorithm

Multi-class logistic regression



Binary vs. Multi-class



Analyzing classifier result

- Confusion matrix
- Metrics
- Error analysis

Confusion Matrix

	Predicted: No	Predicted: Yes
Actual: No	300	80
Actual: Yes	20	600

Confusion Matrix

	Predicted: No	Predicted: Yes
Actual: No	TN = 300	FP = 80
Actual: Yes	FN = 20	TP = 600

Accuracy

	Predicted: No	Predicted: Yes
Actual: No	TN = 300	FP = 80
Actual: Yes	FN = 20	TP = 600

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} = \frac{300 + 600}{300 + 600 + 20 + 80} = .9$$

Accuracy: Rate of corrected prediction

Precision

	Predicted: No	Predicted: Yes
Actual: No	TN = 300	FP = 80
Actual: Yes	FN = 20	TP = 600

$$Precision = \frac{TP}{TP + FP} = \frac{600}{600 + 80} = .882$$

Precision: How many selected documents are relevant?

Recall

	Predicted: No	Predicted: Yes
Actual: No	TN = 300	FP = 80
Actual: Yes	FN = 20	TP = 600

$$Recall = \frac{TP}{TP + FN} = \frac{600}{600 + 20} = .968$$

Recall: how many relevant documents are selected?

F1

	Predicted: No	Predicted: Yes
Actual: No	TN = 300	FP = 80
Actual: Yes	FN = 20	TP = 600

$$F1 = 2 \frac{Prec. Recall}{Prec + Recall} = 2 \frac{.882 \times .968}{.882 + .968} = .923$$

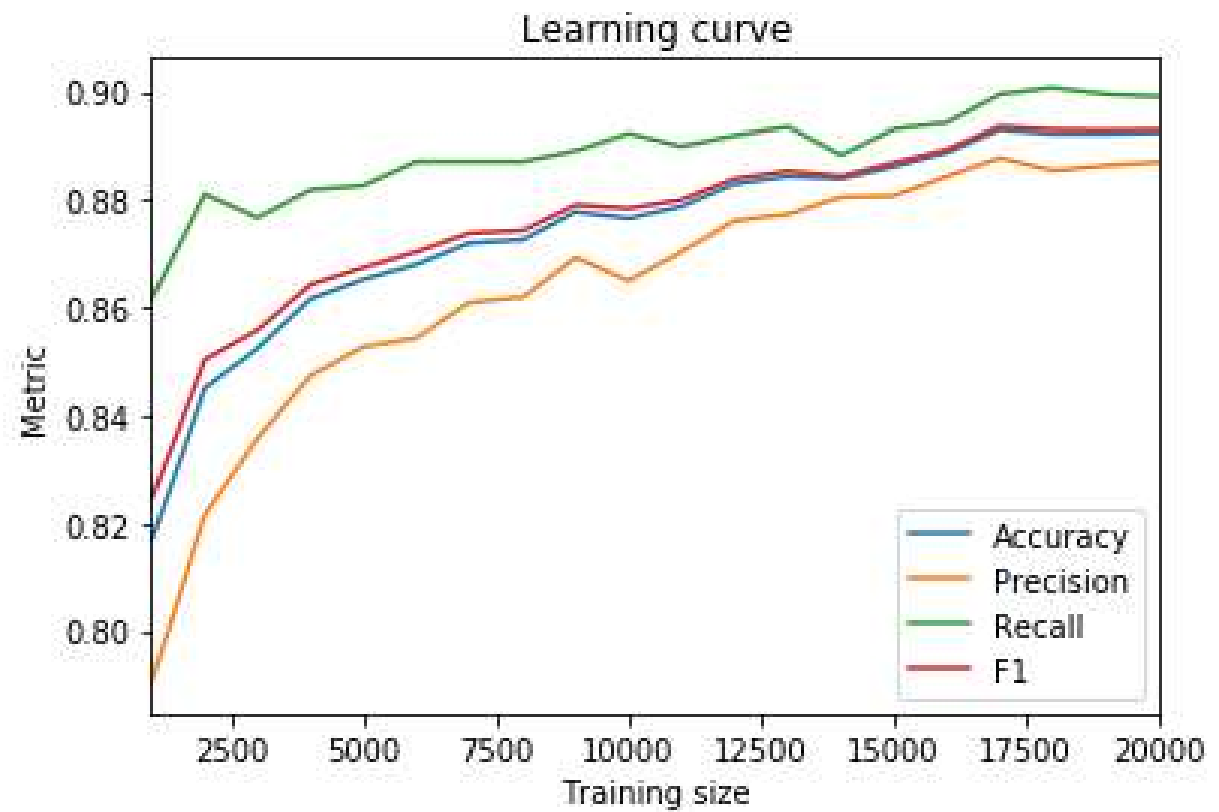
F1: The harmonic average of precision and recall

False Positive Rate (FPR)

	Predicted: No	Predicted: Yes
Actual: No	TN = 300	FP = 80
Actual: Yes	FN = 20	TP = 600

$$FPR = \frac{FP}{FP + TN} = \frac{80}{80 + 300} = .211$$

Learning Curve



Sentiment analysis models

Using

Lexicon

- Requires a dictionary

Using

Text classification

- Requires training data

Demographic classification

- Classifying demographic attributes from textual activities:
 - Gender
 - Race
 - Age
 - Income

Applications

- Marketing
- Public health
- Opinion mining