Machine Learning Introduction – 2

EE698V - Machine Learning for Signal Processing

Vipul Arora

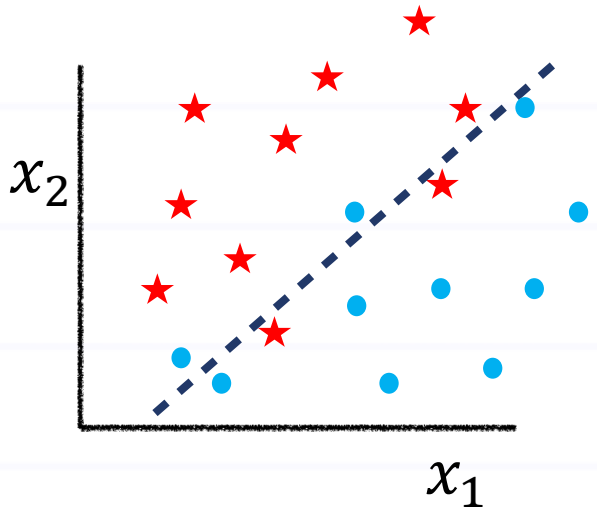
Machine Learning

- Training set
  - Inputs, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\}$
  - Targets, $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots\}$
- Test set
  - Inputs, $\{\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \ldots\}$
  - Unseen targets, $\{\boldsymbol{y}_1^t, \boldsymbol{y}_2^t, \ldots\}$

Evaluation Measures for Classification

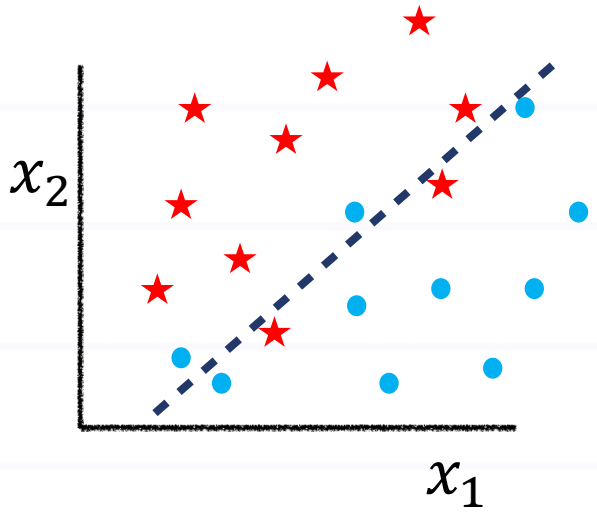
Confusion Matrix

ACTUAL \ PREDICTED	NEGATIVE	POSITIVE
NEGATIVE	True Neg.	False Pos.
POSITIVE	False Neg.	True Pos.



Accuracy

- $Accuracy = \frac{correct}{all} = \frac{TP+TN}{TP+TN+FP+FN}$

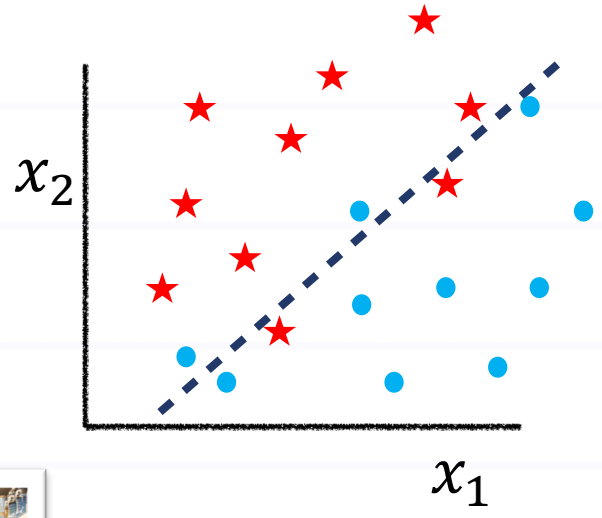


Problems with Accuracy

- E.g.
  - actually 990 R, 10 L
  - predicted all R

Precision and Recall

- $P = \frac{TP}{TP+FP}$



- $R = \frac{TP}{TP+FN}$

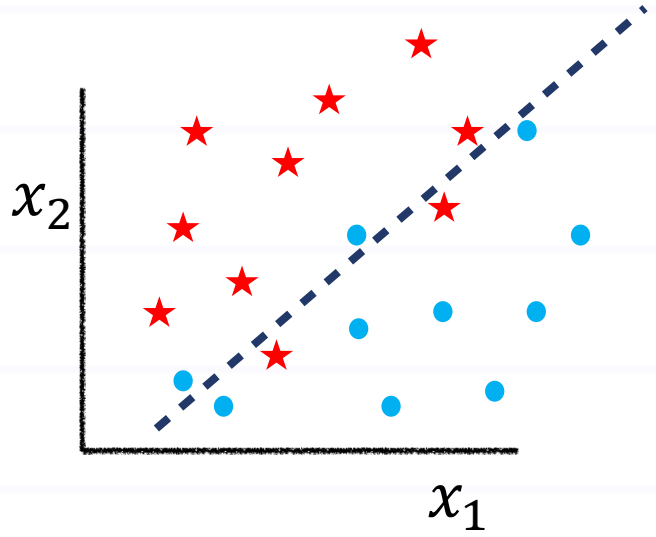
F-score

- $F = harmonic\ mean(P, R) = \left(\frac{P^{-1}+R^{-1}}{2}\right)^{-1} = \frac{2PR}{P+R}$

ROC curve

- Change in Precision vs Recall

  w.r.t. a threshold, e.g. c



Practice problems

- Give some example tasks where we need high precision, but recall can be somewhat compromised
- Give some example tasks where we need high recall, but precision can be somewhat compromised

Typical Steps

1. Fix Goal

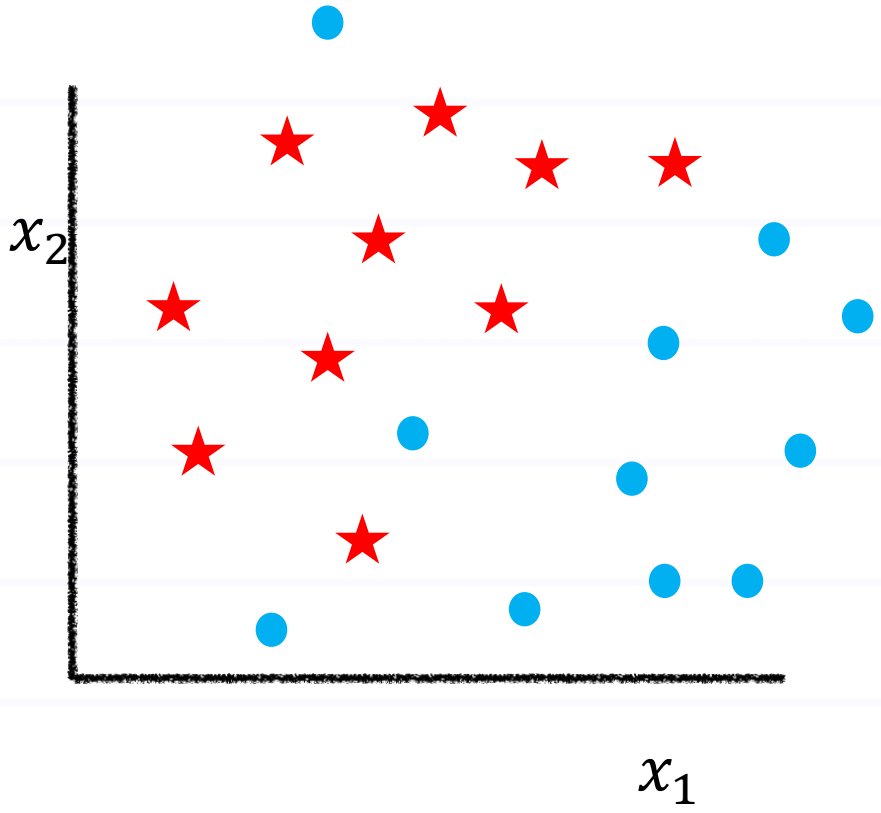
- Classification of flowers into 3 classes

2. Data Collection

- Collect relevant data
- Data variability – select scope of generalization
- Least effort desirable

2. Data Collection

- Noise: errors in measurement due
  - measuring device
  - background phenomena
  - data size artifacts
  - …
- Outlier: data point that differs significantly from the normal trend



3. Data Annotation/Labeling

- Data denoising
- Make conventions for data labeling
  - boundary cases
  - background noise
- Train the team of annotators
- Continuous annotation
- Monitor Annotation Quality
  - Re-train annotators
  - Tune annotation conventions

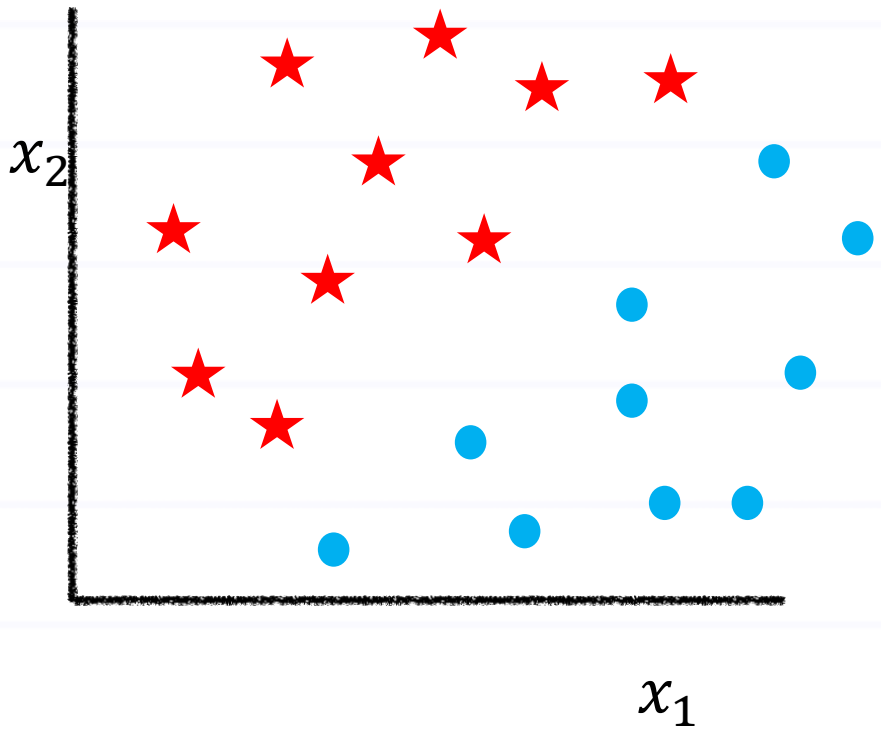
4. Feature Design/Extraction

(height, diameter)

- Find relevant features
  - Informative
  - Invariant
  - Compact

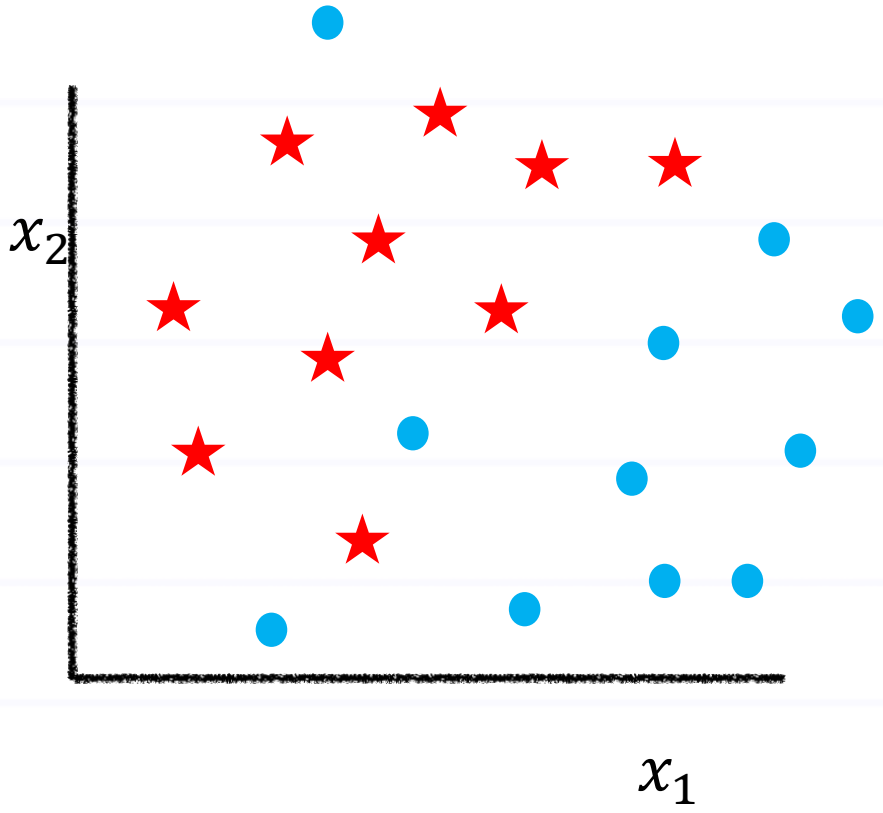
5. Model Selection and Training

- Select what model to use
  - high performance
  - computationally efficient
- Use training data to find optimal values of model parameters (m,c)



6. Handling noise and outliers

- Overfitting
- Validation



7. Testing

- Performance on test data
- Variability

8. Deployment

- Release to beta users
- Release in market

9. Debugging and Model Improvement

- Gather more data for training
- Error analysis
- Go back to step 3

References

- "Pattern Recognition and Machine Learning", C.M. Bishop, 2nd Edition, Springer, 2011.
  - **Section 1.0**