American Sign Language Recognizer

Machine Learning Term Project - Holy Walkamolies

Problem Statement

To make a communication bridge for deaf/mute people to enable them to participate in conversations like everyone else

The sign language gestures are to be converted to text according to the American Sign Language Standard (ASL) using flex sensing gloves and position and angle sensors.

The project was implemented in two phases- number and character recognition

Research Survey

Sign Speak - Principal Component Analysis (PCA) was used to train the glove into recognizing various gestures, and later classify the gestures into alphabets in real time.

Talking Hands - Artificial neural networks are used to recognize the sensor data received from the sensor glove. These values are then categorized in 24 alphabets of English language and two punctuation symbols introduced by the author.

ASL detection using HMM - A Flock of Birds motion tracker are used to extract the features of ASL gestures. With the features extracted from the sensory data, we specify multi-dimensional states for ASL signs in the HMM processor.

Approach

- The dataset was recorded using the SDT sensory gloves -
  - Numbers for the first part
  - Character and words for the second part
- Data was processed and divided into training and test sets
- For number data set :
  - Created different classification models
  - Tested for accuracy
- For character data set:
  - Created classification models for individual characters
  - Used the models for predicting recorded words and tested for accuracy

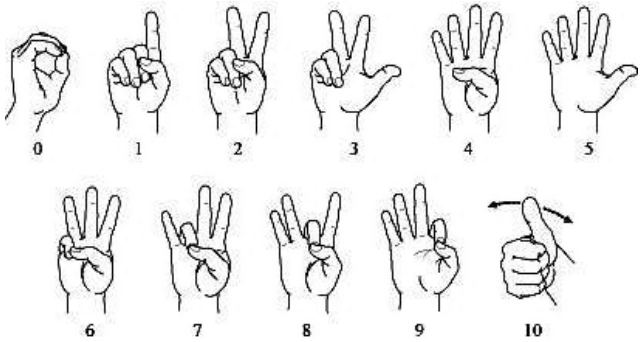
Methodology : Data collection and processing

- SDT Sensory Gloves provide flexion data from fourteen sensors
- Normalization of the readings:
  - The values for every orientation is according to a basic calibration on wearing the gloves for the first time - max and min sensor values were recorded for each sensor separately
  - The recorded values were normalized between 0 and 1 using the calibration results:

$$Value = (Value - min\_calibration)/ (max\_calibration)$$

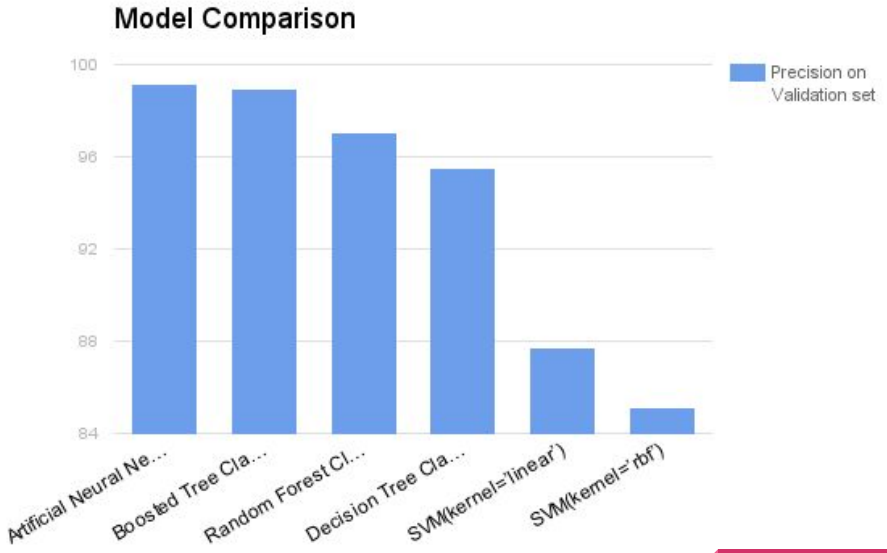
- The Data was recorded by every member of the team for each character/number and the readings were received in a '.csv' format file :
  - Input features for the Problem: Fourteen flexion sensor readings, hence **X** [dimension = 14]
  - Output : **Y :** Classification between 11 classes; i.e, digit values from 0 - 10.

Part I : ASL number recognition



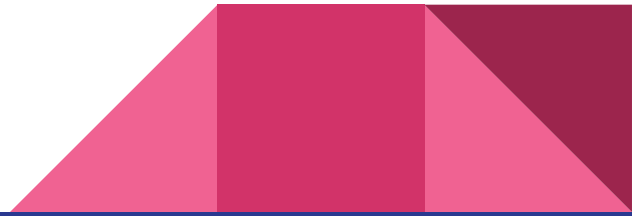
Results

Classifier Model	Precision
Artificial Neural Networks	99.18
Boosted Tree Classifier	98.95
Random Forest Classifier	97.02
Decision Tree Classifier	95.51
SVM(kernel='linear')	87.69
SVM(kernel='rbf')	85.15

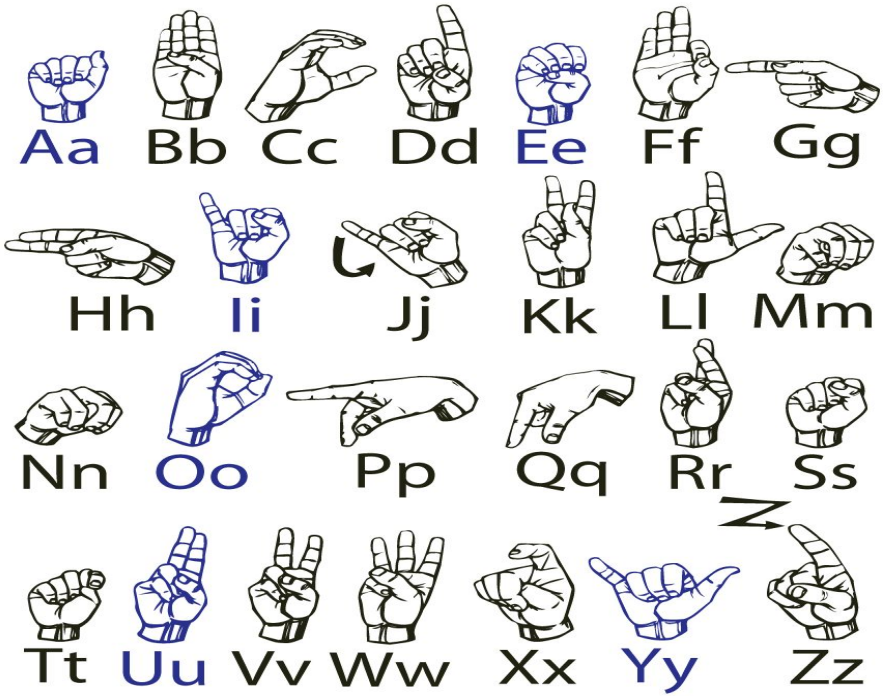


Observations

- The topology of the manifold of gestures is not suitable for RBF kernel
- The linear kernel is more suited than the RBF kernel for this manifold
- Decision trees work better than SVM-Kernel machines because of their ability to model nonlinear decision boundaries efficiently through a piecewise linear approximation
- A random forest (ensemble of decision trees) generalizes better than a single decision tree
- A deep learning artificial neural network performed the best through data-driven representation learning from a huge amount of data



Part II : ASL character recognition

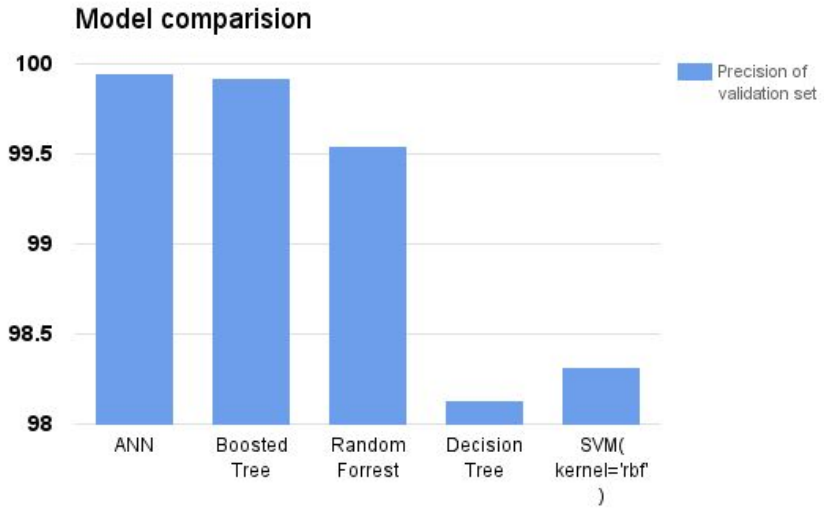


Methodology

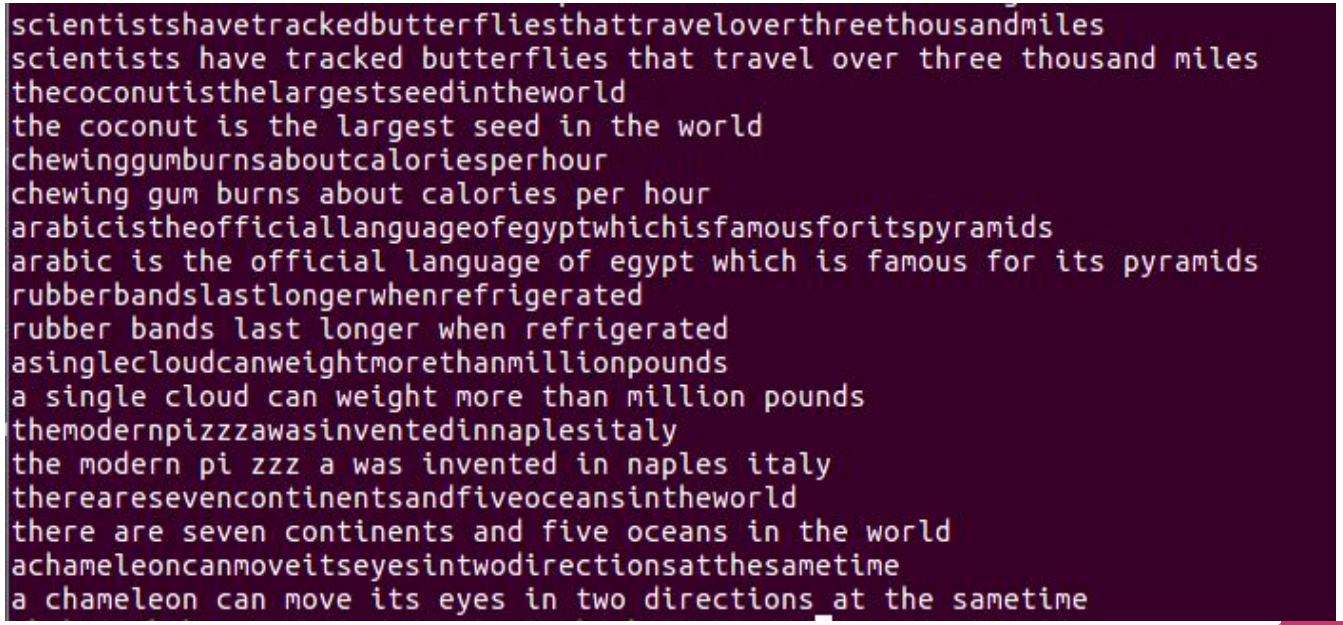
- Data collection and processing was similar to first part
- The same prediction models were used to train the character set recorded
- In order to test the algorithm, we followed four steps:
  - Implemented the models on a dataset of sentences which were also recorded manually
  - A continuous stream of characters was obtained
  - The unique characters were separated from the string
  - Then with the help of a dictionary, we modelled the string distribution to obtain the individual words

Results

Classifier Model	Precision
Artificial Neural Networks	99.95
Boosted Tree Classifier	99.92
Random Forest Classifier	99.54
Decision Tree Classifier	98.13
SVM(kernel='rbf')	98.32

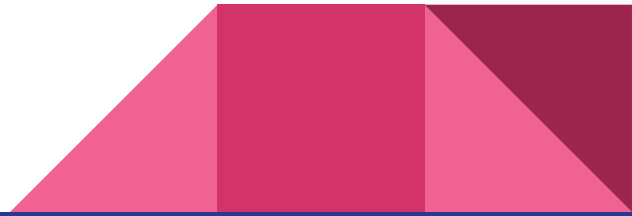


Sentence Recognition



Conclusion and Future Work

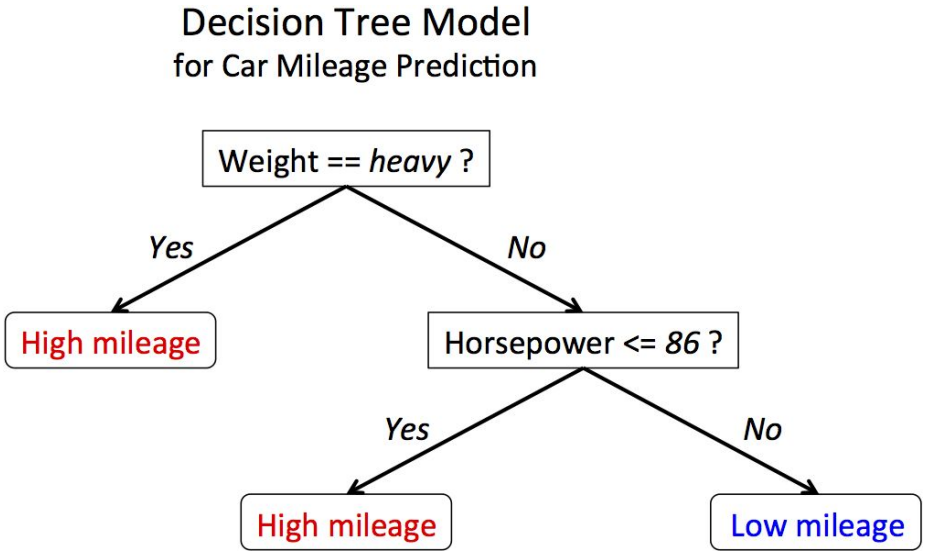
- Existing model gives very good results.
- However our existing character stream to sentence conversion can be improved by making a model (using LSTM etc) for that as well.
- Training of characters and then forming words is a bit slow and tough to implement in real time but the best possible option given the lack of data.
- Need to take a lot of data sample for sentences and words and train on that using ANN, GBT, HMM and other algorithms.



Thank You

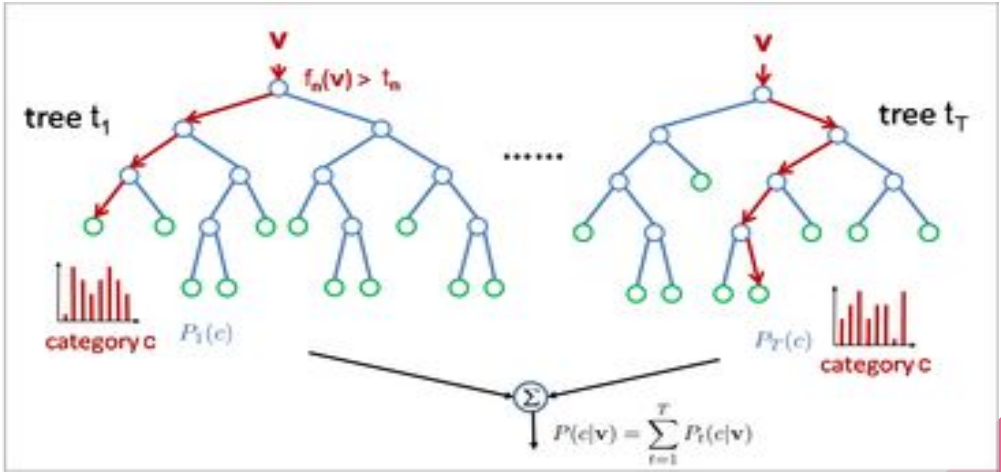
Prediction Models

- **Decision Trees** :
  - Maps observations about an item to conclusions about the item's target value.



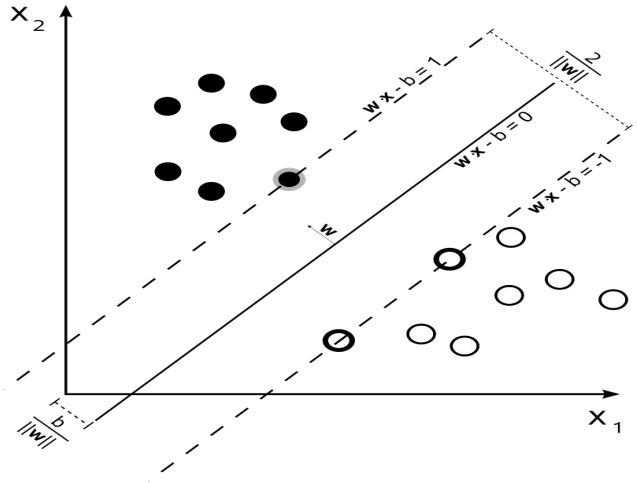
Prediction Models

- **Boosted Trees** :
  - Builds the model in a stage-wise fashion and generalizes them by allowing optimization of an arbitrary differentiable loss function.
- **Random Forest :**
  - Constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



Prediction Models

- **Support Vector Machines:**
  - Assigns new examples into one category or the other.
  - Representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.



Prediction Models

- **Artificial Neural Networks :**
  - Computational model based on the structure and functions of biological neural networks.
  - Nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled