Sprint #2
Instrument Recognition Software

Members: Aleks, Aner, Axel, Cuong, Joe and Thomas

BRD and Management plan updates:

- Our team did not find it necessary to add any new information to the **Business Requirement Document** as our project is focused more on researching sound recognition technology
- For the **Management Plan** we updated the **Sprint Board** and the **Burndown Chart**

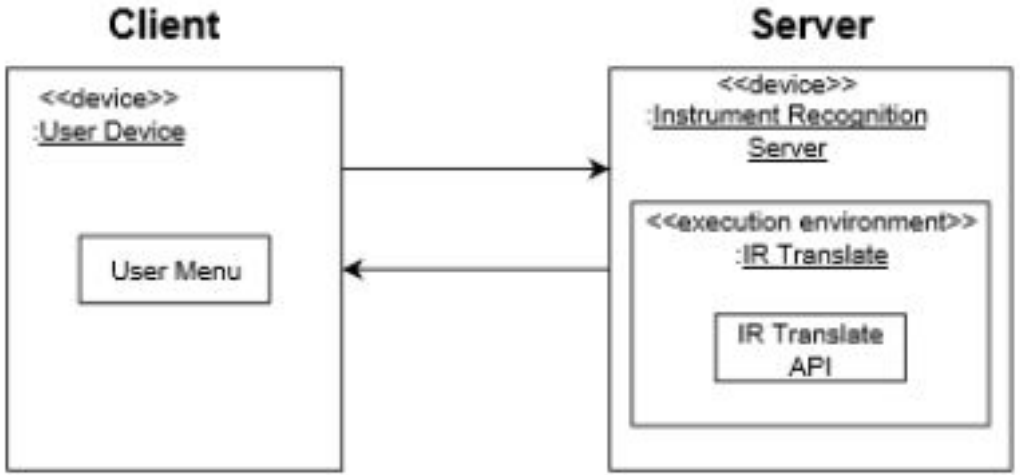
Frontend

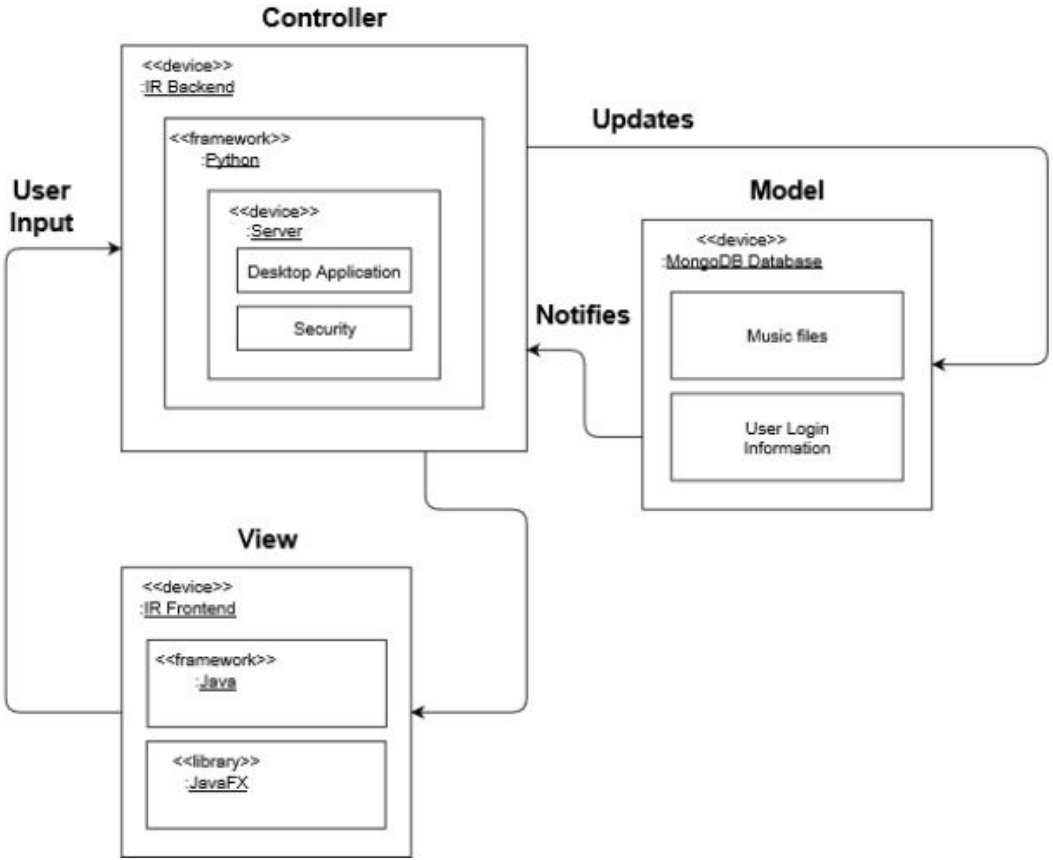
- Javafx for GUI (building using JavaFX SDK)
  - Desktop App
  - Mobile App (potentially)

Backend

- Python for the server
- MongoDB (non-relational) database
  - Connect using pymongo
  - Store usernames and hashed passwords

Instrument Recognition Architectural Design

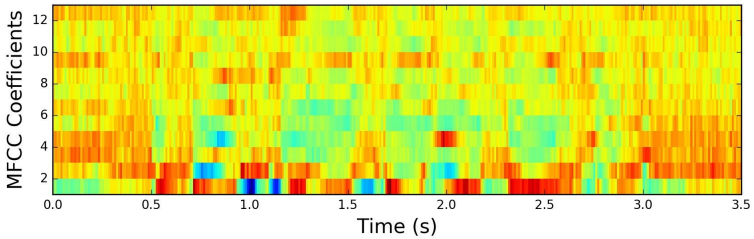



Controller
<<device>>
:IR Backend
<<framework>>
:Python
<<device>>
:Server
Desktop Application
Security
Updates
Model
<<device>>
:MongoDB Database
Music files
User Login Information
Notifies
User Input
View
<<device>>
:IR Frontend
<<framework>>
:Java
<<library>>
:JavaFX

Data Processing

Features:

- ★ Mel-Frequency Cepstral Coefficients (MFCCs)
  - Commonly used as features in speech recognition systems, such as the systems which can automatically recognize numbers spoken into a telephone.
  - Increasingly finding uses in music information retrieval applications such as genre classification, audio similarity measures, etc.
  - Can be used to analyze the Timbre space (the uniqueness of an instrument from another).



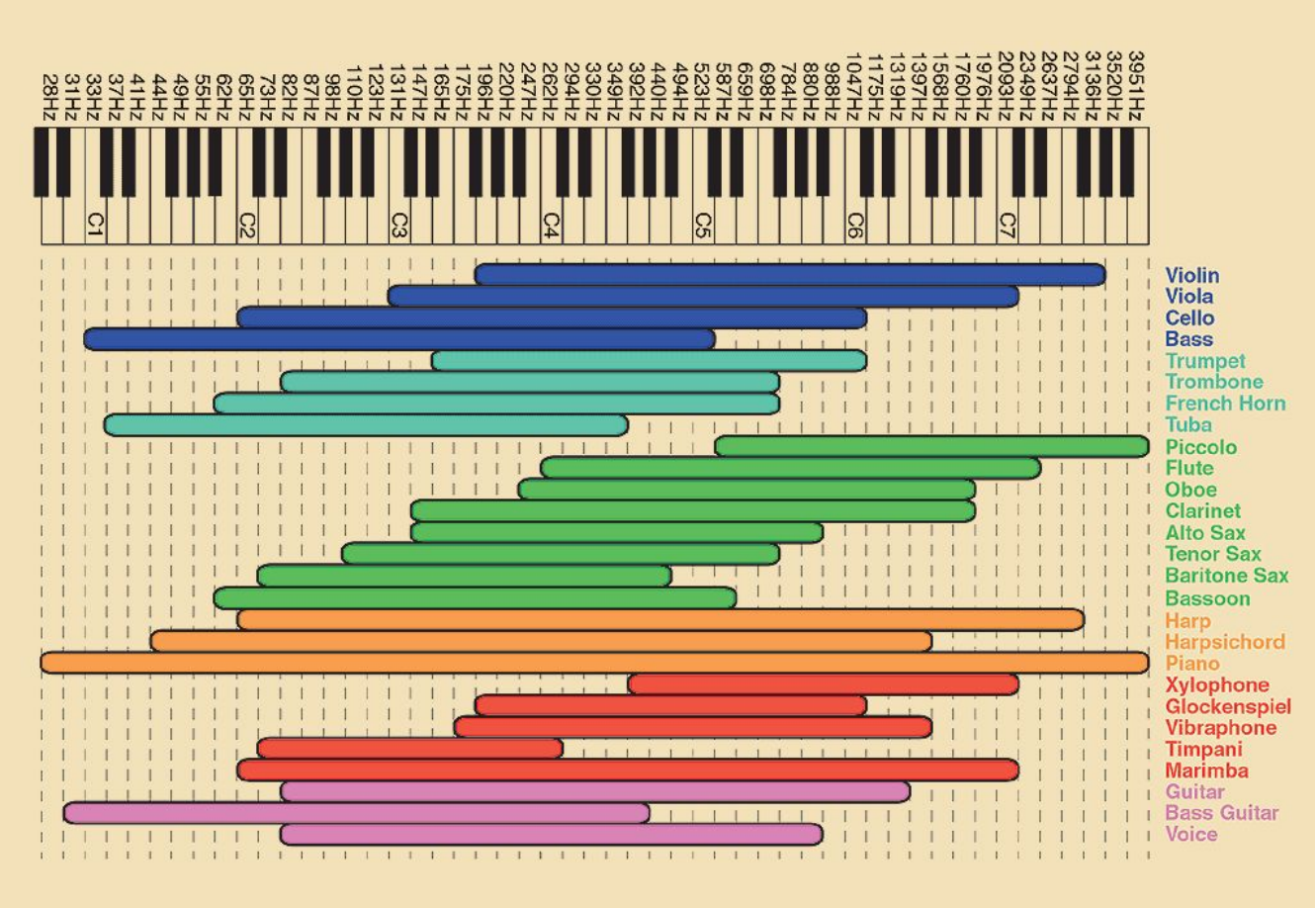
Data Processing

Features:

- ★ Tone
  - The difference in frequencies

Frequency Band	Hz
Low bass	20-40
Mid bass	40-80
Upper bass	80-160
Lower midrage	160-320
Middle midrange	320-640
Upper midrange	640-1280
Lower treble	1280-2560
Middle treble	2560-5120
Upper treble	5120-10200
Top octave	10200-20400

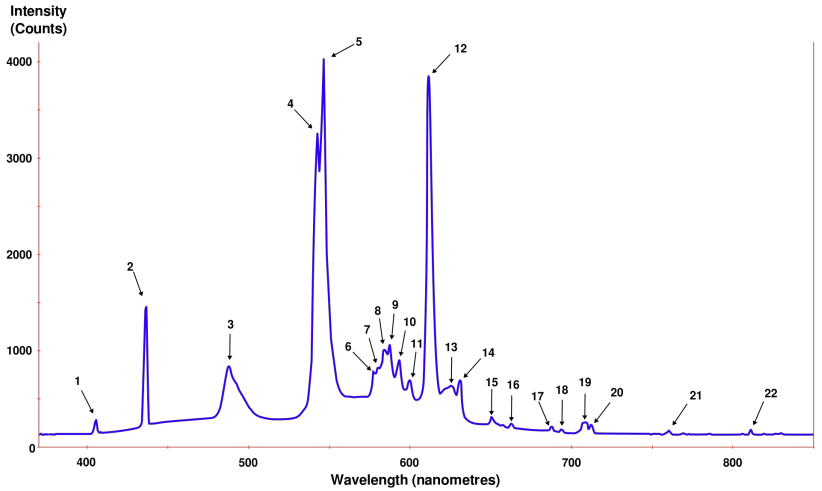
Equalization Chart



Data Processing

Features:

- ★ Power Spectral Density
  - The power of each frequency in a signal
  - Shows at which frequencies variations are strong and at which frequencies variations are weak
  - Energy per frequency(width)
  - Can obtain specific frequency range by integrating PSD within that frequency range



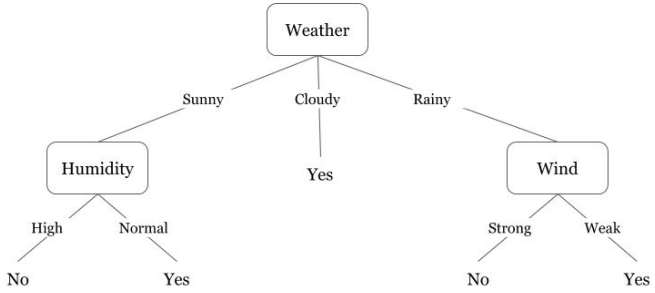
Data Processing

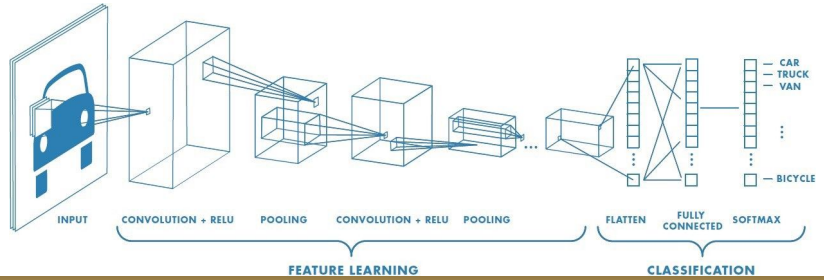
- ★ What is PSD good for?
  - From our current research we noticed that every instrument produces a relatively unique shape when graphed.
  - Also the image is not based on time is based on power and frequency.
  - This information on how all these power is distributed could be later used in filtering.
  - Where you choose to focus on a bandwidth that your signal of interest is.

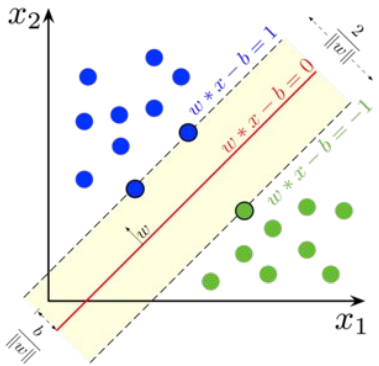
Candidate Models

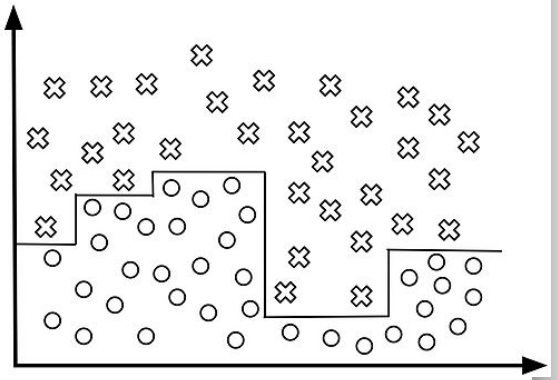
Label focused machine learning:

- Decision Tree
- Random Forest
- SVM
- Convolutional Neural Network

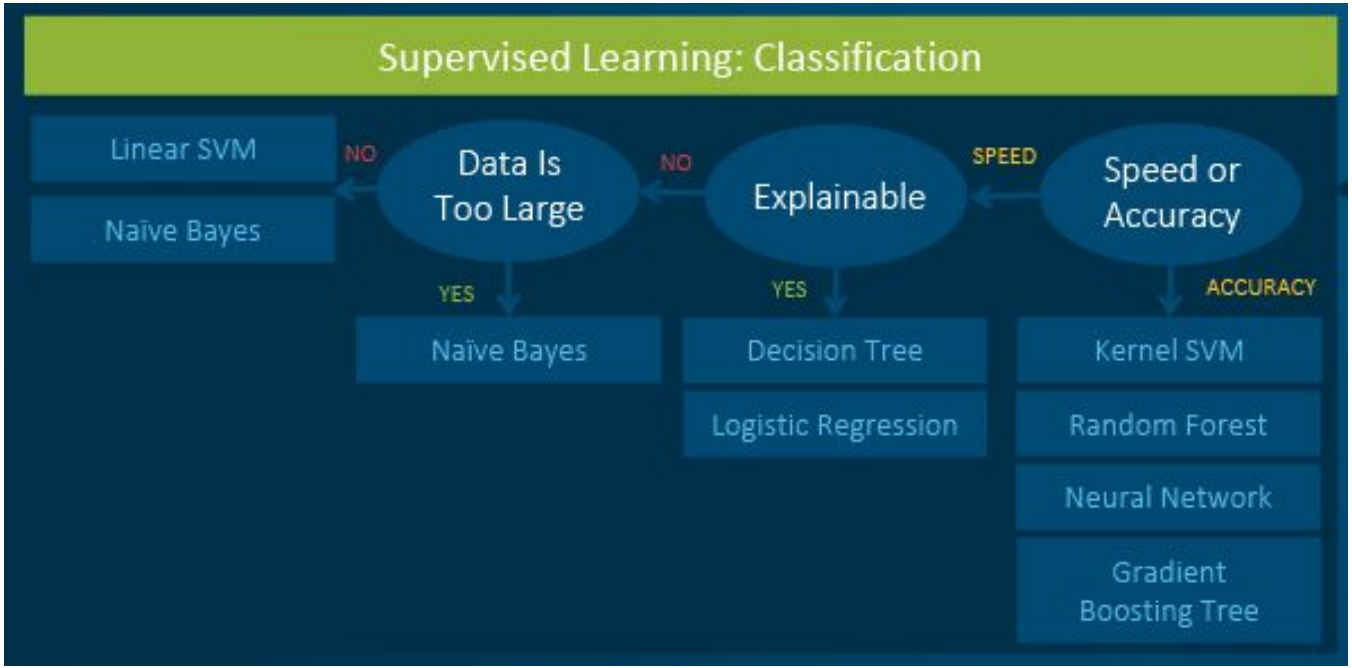








Candidate Models (cont.)



Progress and Plans:

Current Developments:

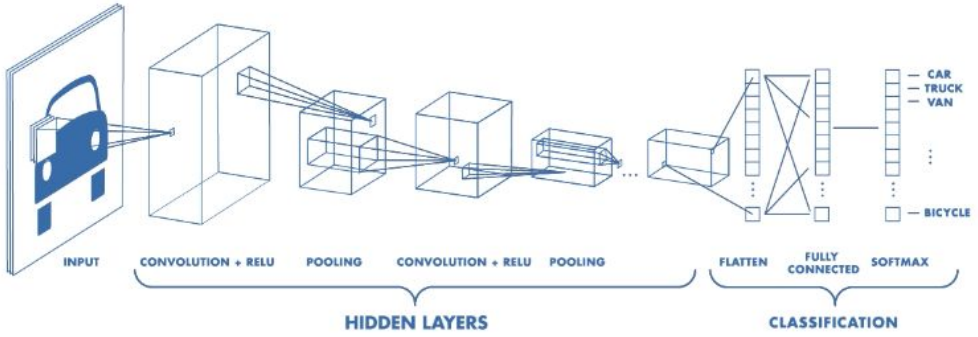
- Decision Tree (Regression)
  - Classification
  - Fast
  - Justifiable

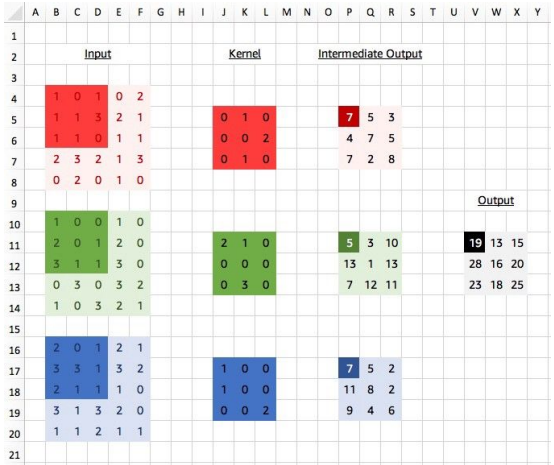
Future Plans and Rational:

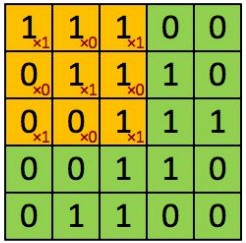
- Convolutional Neural Network
  - Image processing
  - Pattern recognition
- Naive Bayes
  - Large data

What is CNN?

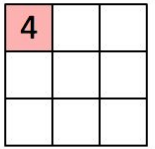
- Built of layers that transform data
- **Layers:** composed of filters (feature matrix) that indicate desired patterns in the data
- Condensed using a dot product
- **Pooling**: most relevant data
- **reLU** (Normalization): removes negatives
- **Drop out**: random disabling
- **Softmax**: creates probabilities for all labels







Image



Convolved Feature

Sprint Goals

Planned:

- Research
- Setting up the frontend, backend and database
- Understand Power Spectral Density
- Determine Candidate Models to use

Completed:

- Understanding CNN
- Research
- Setting up the frontend

Burndown Chart

Didn’t quite have enough time to fully accomplish our goals in this sprint (idk what i’m saying someone fix this)