MACHINE LEARNING REFERENCE ANDROID APP PHASE 1

APPLICATION NAME - yet to be decided

SDLC

Preliminary Analysis

App Objective -

- A tool for amateur machine learning engineers who have just started in ML, to visualize and play around with the different algorithms at their own pace.
- A reference app for expereinced ML engineers who quickly want to check the implementation detail of a particular algorithm. Rather than googling on their computer and wasting the screen space they can keep this app on their phones which is otherwise lying idle at work to refer the algos.

Problem Statement -
Make an open source android app which visualizes ML algorithms using interactive graphs and lets the user play interactively with different settings in the algorithms on different pre defined datasets.

System analysis and requirements definition

Project goals, functions & operations -

- describe each algorithm and ML concept in a simple language following the framework of what, why, how and when along with a brief history of that concept.
- supply atleast 3 different datasets on which the algorithm can be applied, also supply atleast 5 examples of datasets, usecases and applications of a particular algo.
- using 2D and 3D graphs, visualize the algorithm on a particular dataset, allow the dataset to be choosen by the user from the 3 supplied data sets.
- allow the user to play with the graph by tweaking the settings of a particular algo.
- all the content including the dataset, algo defintion, texts, images, graphs and other application resources to be stored online.
- if the data is stored online due to design decisions, allow the content to be made offline by the user on click of a button, subsequently allow deletion of the offline content from the user's app on click of a button
- allow the user to create bookmarks for a particular section of the app like algo or introduction section.
- make a bookmarks section which can be accessed by the user in minimal click(s) which lists all the bookmarked content by the user.
- allow navigation from one section to other by a navigation pane or directly navigate to the home using a single click
- make a about section of the app
- make a contribute section in the app where the user can sign up if he wants to contribute in the content, development or marketing of the app.
- If a keyword related to ML is used in section and we have that keyword in our app, hyperlink that keyword so that the user can click on the keyword and reach the section where it is explained.

Proposed sections in the app for First Phase -

- **Introduction**
- **Supervised Learning**
- **Unsupervised Learning**
- **Data sets**
- **General topics**
- **Resources**

INTRODUCTION -
- Definition of machine learning including what, why, how, when and a brief history.
- Types of machine learning
- General machine learning ideas, concepts and keywords

SUPERVISED LEARNING
1. Linear regression with one variable
2. Linear regression with multiple variables
3. Gradient descent
4. Normal equation
5. Logistic regression
6. Feature scaling & mean normalization
7. Classification
8. One vs. All
9. Regularization
10. Neural Networks - Non linear hypothesis
11. Backpropagation algorithm
12. Gradient Descent
13. Random initialization
14. Support vector machine
15. Kernels

UNSUPERVISED LEARNING
- K - means algorithm
- Principal component analysis
- Dimensionality reduction
- Anomaly detection
- Recommender systems

DATA SETS
- 3 data sets to be used (yet to be decided)

GENERAL TOPICS
- How to select the best ML model ?
- Bias vs. Variance
- Reguralization & Bias / Variance trade off
- Learning curves
- Error analysis / metrics
- Skewed data
- Validation / testing / training data
- Comparison between different ML libraries

* The above structure of the sections are inspired by the Andrew NG's ML course on Coursera. Upon completion of the above sections, other ML algorithms to be added.

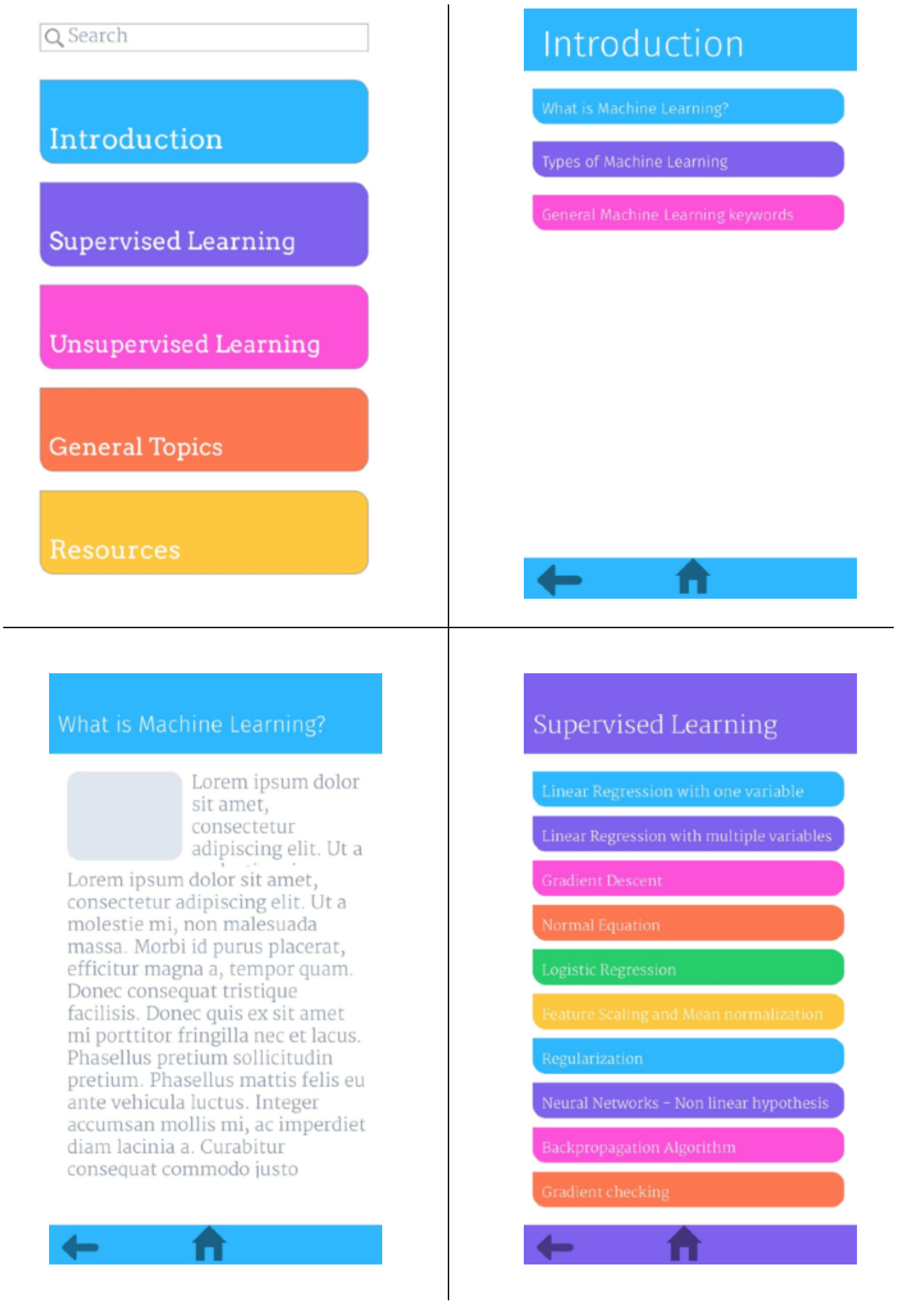
<u>RESOURCES</u>

- Blogs / Websites / ML news sites / Journals
- Online courses
- Online competitions
- Notable scientists to follow
- Twitter accounts to follow
- Facebook and google groups
- Softwares / Packages / Frameworks / Programming languages
- Where to get help?
- Videos to watch
- Notable conferences
- ML companies to follow
- Notable apps which use machine learning
- Books to read

System design

Rough GUI Mockups

The UI would use google material design elements and cards based lists.


Search
Introduction
Supervised Learning
Unsupervised Learning
General Topics
Resources
Introduction
What is Machine Learning?
Types of Machine Learning
General Machine Learning keywords
What is Machine Learning?
Supervised Learning
Linear Regression with one variable
Linear Regression with multiple variables
Gradient Descent
Normal Equation
Logistic Regression
Feature Scaling and Mean normalization
Regularization
Neural Networks - Non linear hypothesis
Backpropagation Algorithm
Gradient checking

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut a

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut a molestie mi, non malesuada massa. Morbi id purus placerat, efficitur magna a, tempor quam. Donec consequat tristique facilisis. Donec quis ex sit amet mi porttitor fringilla nec et lacus. Phasellus pretium sollicitudin pretium. Phasellus mattis felis eu ante vehicula luctus. Integer accumsan mollis mi, ac imperdiet diam lacinia a. Curabitur consequat commodo justo

Use cases -

1. Accessing an algorithm implementations

Goal - to access an algorithm's implementation

Scope - user interacting with the main application

Level - user level

Preconditions - Network connection active, permission to read from the file system

Success condition - all the implementation details including the tweaks, graphs load on screen

Failure condition - the algorithm implementation details do not load

Trigger - the user clicks on the algorithm's name in the list

Notes - The algorithm will be slow to load if the network connection is slow

Steps -
- The user opens the app
- He is shown the home screen with a search bar at top
- The user enters the search term in the search bar and presses enter
- The application searches the term internally and lists the most relevant results
- The user clicks on the desired result
- The clicked result is presented to the user

Exception -
- There is no network connection
- The user will be displayed a message saying 'please check your network connection'
- Also, in the center of the screen, there will be a refresh button which the user can press to reload the implementation details after the network connection is activated

Extension -
- If the network connection is not present and the user is displayed the 'Exception', when the network connection is active again, the system listens for it and automatically refreshes the algorithm

2. Playing with the algorithm

Goal - to tweak the algorithm parameters and play with the graph

Scope - user interacting with the main application

Level - user level

Preconditions - network connection is active, permission to read from the system's file system

Success condition - the user is able to tweak all the parameters of the algorithm and the graphs updates in real time.

Failure condition - the user is unable to tweak the parameters or the graph does not update

Triggers - the user chooses to change the parameters of the algo, the user clicks on alternative dataset

Steps -
- The user opens the app and accesses a particular algorithm
- With the algorithm details on screen, the user scrolls down and is shown a graph with a particular dataset out of the 3 datasets
- The user is also shown options on the top of the screen to choose a dataset out of the 3 datasets.
- A particular set of parameters in the algorithm are also presented with an option to tweak those parameters.
- The user can change the dataset and the graph and other details change upon doing so.
- The user also tweaks the parameters of the algorithm which are then reflected on the graph and other details such as cost function, thetas, etc.

Exceptions -
- the user changes the parameters and the graphs do not update - display a message that "we are unable to update the graph right now, please try in sometime"

3. Bookmarking a section or an algorithm and accessing it

Goal - to bookmark a section of the app and accessing the bookmark section

Scope - System and user

Level - user level

Preconditions - network connection is active

Success condition - bookmark created for the section and added to the bookmark section

Failure condition - bookmark could not be created

Trigger - the user clicks the bookmark button inside the section at the top corner

Steps -

- The user opens the section or the algorithm and presses a button on the top right of the screen to bookmark the algorithm
- The particular algorithm or section is added to the bookmarks tab in the application
- The user clicks on the bookmarks section in the navigation bar below and accesses the section
- The user is presented with a list of bookmarked sections and algorithms

Exception -
if the bookmark display a message “the bookmark could not be created please try again”

4. Starting the learning mode [for ML beginners]

Goal - start the learning mode

Scope - System and user

Level - user level

Preconditions - network condition is active

Success conditions - the learning mode starts

Failure conditon - the learning mode does not start

Trigger - the user clicks the play button in the app

Steps -

- The user opens the app and is presented with the home screen
- The user presses a play button(icon) on the top right of the screen
- After pressing the play button, the first section with the first subsection of the app is shown to the user
- The user reads the content on the first section and reaches the end of the text. Below the text he is shown a next button which the user presses
- After pressing the next button, the user is navigated to the next relevant section of the app.

Exception - if the learning mode does not start, display a message “the learning mode could not start”

5. Displaying the implementations in a particular algorithm with a chosen dataset

Goal - change the implementations by choosing an alternate dataset

Scope - System

Level - system level

Preconditions - the network connection is active, the database is connected

Success condition - the implementation details change when alternate dataset is chosen

Failure condition - the dataset doesn't get loaded or the network connection is not active and the updation doesn't happen

Steps -

- The user opens a particular algorithm and is presented with the implementation details of the app.
- The graph section of the algorithm is presented with the default dataset
- The user is shown a tabbed option of 3 datasets in the database
- the user chooses the dataset, the dataset is queried and loaded from the database
- automatically all the details of the algorithm is updated with the choosen dataset
- the graphs also update automatically with the chosen datasets

Exceptions - if the updation doesn't happen then display a message "please try again in sometime"

6. Tweaking the algorithm parameters

Goals - tweaking the algorithm parameters

Scope - system

Level - system

Preconditions - network connection is active

Success conditions - the graph updates and other parameters update

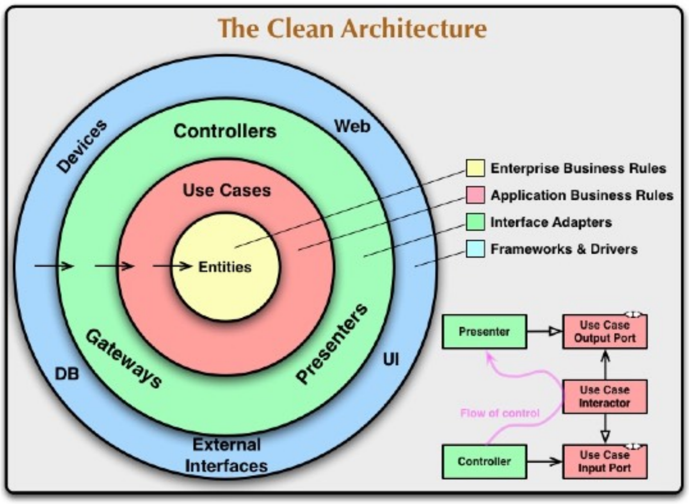
Failure conditions - the graph and other parameters do not update

Steps -

- The algorithm parameters such as thetas, cost function, etc. have optimal values set in them.
- The user now chooses to change the value of theta, (the given range of thetas is displayed below)
- once the theta is changed, the graph changes according to the theta, the cost function updates according to thetas.

Exceptions - if the updation does not happen then display a message "network connection is not active, please try again"

App components and architecture -



Architecture -

The architecture will be a three layer architecure based on 'Clean Architecture' by Robert C. Martin as shown above minus the entities core layer as that is the enterprise logic layer. Also, Clean architecture is in line with "object oriented software engineering : use case driven approach" by Ivar Jacobson.

Advantage of Clean - Though it takes a lot longer to code, we can use the same architecture in future when we plan to release an iOS app or a web app as this architecture is highly modular and very use case oriented.

1. Outer layer(implementation layer) - UI, Storage, Network
2. Middle layer(interface adapter layer) - Presenters, Converters
3. Inner layer (core application logic layer) - Interactors, Models, Repositories, Executor

Outer layer -

1. UI - this is where we put all the activities, fragments, adapters and other android code related to the UI
2. Storage - Storage specific code that implements the interface our interactors use for accessing data and storing data. This includes for example content providers in android.

Middle layer -
Glue code layer which connects the implementation details with the application logic

1. Presenters - handles events from the UI (eg. User click) and usually serve as callbacks from inner layers (interactors)
2. Converters - converter objects are responsible for converting inner models to outer models and vice versa.

Inner layer -

- the core layer contains the most high level code. All classes here are POJO (plain old java objects).
- Classes and objects here have no idea that they are run in an android app and can easily be ported to any machine running jvm

1. Interactors - these are the classes which actually contain the app logic code. These are in the background and communicate events to the upper layer using callbacks. They are also called use cases in some projects (probably a better name). It is normal to have a lot of smaller interactor classes that solve specific problem. This conforms to the single responsibility principle and in my opinion is easier on the brain.
2. Models - these are the app models that you manipulate in your business logic
3. Repositories - this package only contains interfaces that the database or some other outer layer implements these interfaces are used by interactor to access and store data. This is called a repository pattern
4. Executor - This package contains code for making interactors run in the background by using a worker thread executor. This package is generally not something we need to change.

Tools and libraries to be used -

1. App development -	Android SDK, Android studio
2. ML library / framework -	Weka or Tensorflow
3. Version control and branching -	Git and GitFlow
4. Documentation -	Javadocs and Github
5. Continuous integration -	Jenkins
6. Continuous deployment -	fastlane
7. Automated testing -	Robotium
8. Server for Jenkins -	AWS
9. Project management tool -	Asana(free version)
10. Static Analysis -	Android Lint
11. Database and file server -	AWS

Proposed Version releases in Phase 1 -

Version 0.1 - Only the first topic “Introduction” and the “Resources” section
Version 0.2 - First four topics in Supervised Learning i.e. Linear Regression with one/multiple var
Version 0.3 - Topic five to nine of “Supervised Learning” section
Version 0.4 - Topic ten to thirteen of “Supervised Learning” section
Version 0.5 - Topic fourteen and fifteen of “Supervised Learning” section
Version 0.6 - First three topics of “Unsupervised Learning” section
Version 0.7 - Topic four and five of “Unsupervised Learning” section
Version 0.8 - First four topics of “General Topics” section
Version 0.9 - Last four topics of “General Topics” section
Version 1.0 - Include features of bookmarking, keyword hyperlinking, etc.

App Components -

Work in progress