**RightPrice - AI Based Price Estimation System**

|  |
| --- |
| A Project Report Presented to  The Faculty of the Computer Engineering Department |
| San Jose State University In Partial Fulfillment Of the Requirements for the Degree Bachelor of Science in Software Engineering |

|  |
| --- |
| By |
| Manmeet Gill  Sidarth Shahri  Manpreet Singh  Karthik Tella |
| 10/2019 |

|  |
| --- |
| Copyright © 2019 |
| Manmeet Gill  Sidarth Shahri  Manpreet Singh  Karthik Tella |
| ALL RIGHTS RESERVED |

|  |
| --- |
| **APPROVED FOR THE COLLEGE OF ENGINEERING** |
|  |
| Dr. Vishnu Pendyala, Project Advisor |
|  |
| Professor Rod Fatoohi, Instructor |
|  |
| Dr. Xiao Su, Computer Engineering Department Chair |

# ABSTRACT

## RightPrice - AI Based Price Estimation System

By Manmeet Gill, Sidarth Shahri, Manpreet Singh, Karthik Tella

Online shopping has grown massively over the past couple of years. For example, Amazon is one of the biggest websites for online shopping. Another very popular online market is buying/selling old and used items. Instead of having an old device lying around, it makes sense to want to get some money for it instead. However, there is no real way of knowing the prices of those items. It would not make sense to overpay for an older device. In the same sense, it would also not make sense to sell a valuable device for a very cheap price.

Manually visiting several seller websites and looking at various listings is not feasible if one wished to unload many products in a short time period. Looking for price quotes on the market for a variety of products is very time-consuming. Therefore, there is pretty much no effective way to find the average price of an item. Additionally, there are issues with sellers misquoting their prices, both accidentally and intentionally. Additionally, the price of used products on the market should degrade with time as the product deteriorates, occupies valuable virtual “shelf space”, and continues to be a burden on the seller. This follows that the seller would want to lower the price to get rid of the item faster or to make the sale.

Our product would essentially do most of the tedious work for the user, and return a value that is the average price of the given product based on the data gathered by our product. This would be possible through web scraping data from various used marketplaces such as Amazon, Craigslist, Ebay, etc. Then, using and filtering the dataset obtained from web scraping, AI models can be trained to predict price of certain item given some parameters based on average price of similar items in the dataset.

**Table of Contents**

**Chapter 1 Introduction**

* 1. Project Goals and Objectives
  2. Problem and Motivation
  3. Project Application and Impact
  4. Project Results and Deliverables
  5. Project Report Structure

**Chapter 2 Background and Related Work**

* 1. Background and Used Technologies
  2. Literature Survey
  3. State-of-the-art Summary

**Chapter 3 Project Requirements**

3.1 Domain and Business Requirements

3.2 System (or Component) Functional Requirements

3.3 Non-functional Requirements

3.4 Context and Interface Requirements

3.5 Technology and Resource Requirements

**Chapter 4 System Design**

4.1 Architecture Design

* 1. Interface and Component Design
  2. Structure and Logic Design
  3. Design Constraints, Problems, Trade-offs, and Solutions

**Chapter 5 System Implementation**

5.1 Implementation Overview

5.2 Implementation of Developed Solutions

5.3. Implementation Problems, Challenges, and Lessons Learned

**References**

Asim, Muhammad & Khan, Zafar. (2018). Mobile Price Class prediction using Machine Learning Techniques. *International Journal of Computer Applications*. 179. 6-11. 10.5120/ijca2018916555.

Deloitte. (2017). Used Smartphones: $17 Billion market you may have never heard of. *Deloitte Network Global,*(4), 3. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Technology-Media-Telecommunications/gx-tmt-prediction-used-smartphones.pdf.

Harrison, Onel. “Machine Learning Basics with the K-Nearest Neighbors Algorithm.” Medium, Towards Data Science, 14 July 2019, www.towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

**List of Figures**

Figure 1. Process Summary Diagram Page No. 15

Figure 2. Class Diagram Page No. 16

Figure 3. Interface and Component Design Page No. 21

Figure 4. Structure and Logic Design Page No. 21

**List of Tables**

Table 1. Functional Requirements Page No.17

Table 2. Non-Functional Requirements Page No.18

Table 3. Technology and Resources Page No.19

# Chapter 1. Introduction

**1.1** **Project Goals and Objectives**

The project’s main goal was to empower customers who are buying and selling items on online marketplaces for used goods such as Craigslist, Facebook Marketplace and eBay. Our product aimed to utilize advertisements from online marketplace as precepts to train a learning agent to utilize the knowledge gained from these precepts to provide estimates and analytics about the availability and price of desired products.

**1.2** **Problem and Motivation**

The main problems with our project were mostly technical and legal. Technical problems included making sense of online advertisements, which are mostly unstructured, web scraping from dynamic websites, and hosting our learning agent. Legal problems mainly came down to the accessing rights on websites. Some websites are against third parties accessing their source code and benefiting from it, thus claiming them to be property. Such websites can be scraped but usually should not be scraped without the consent of the domain owner. Therefore, gaining rights or consent to access some marketplaces was a problem. The main motivation of the project was to empower online customers and sellers given the vast and growing marketplaces online.

The needs for our project mainly included ad data, hosting space and source code/data access privileges to preexisting marketplaces. With online marketplaces for used products gaining popularity, the sheer number of advertisements could overwhelm buyers and sellers in making a decision. Our project used web scraping and machine learning to analyze a big portion of these ad space to analyze common trends for a given product simplifying the process of researching and decision making for users.

**1.3 Project Application and Impact**

The goal of our project was to be able to return to the user an accurate price of an item on the used marketplace. Our completed project has the potential to directly affect the used marketplace. Our project also has the potential to impact academics, society, and industry. Firstly, in school students tend to buy used textbooks because they tend to be a lot cheaper than buying brand new. Our project helps with this process because students are now able to get a general price for the textbook, and they have the power to find the best deal. Students are now able to find expensive materials such as laptops, textbooks, and software for a good price with our project. This also applies to society. Technology is becoming more and more advanced, and with it, prices have also been gradually going up. For example, the iPhone 11 Pro costs around $1000 brand new. This is where the used marketplace comes in handy. Consumers can buy slightly used technologies for a fraction of the price, making some of these expensive technologies obtainable for the lower/middle class. This will also obviously have an impact on industry as used marketplaces take customers away from the new marketplaces. Industry will not take a direct hit in terms of revenue because in order for an item to be on the used marketplace, someone has to have bought it first. In the future, if the used marketplace becomes very popular, it could take a large amount of customers away from buying new items.

**1.4 Project Results and Deliverables**

We are now able to present a clean front-end website/application, where the user is able to search for an item and receive an accurate price for that item. We have also designed a back-end. There are two parts to the back-end: data collection and data parsing. We have also obtained data from various used marketplaces, and parsed through that data to be able to calculate an accurate average price of a given item. **Chapter 2. Background and Related Work**

**2.1** **Background and Technologies**

The project was divided into 3 main segments: data procurement, data analysis (learning), and interface. Data procurement mainly dealt with accessing advertisements from online marketplaces. To make the process more seamless our project implemented web scraping using Python 3 with the beautifulsoup and requests libraries. As a part of procurement, we structured the data and saved it as (.csv) comma separated values format. We heavily used Python 3 and Microsoft Excel with the Pandas library.

Data analysis mainly encompassed our intelligent agent which took the data from the previous step and trained a model written using Python 3 and the Tensorflow API. It was built using the K-Nearest Neighbors algorithm.

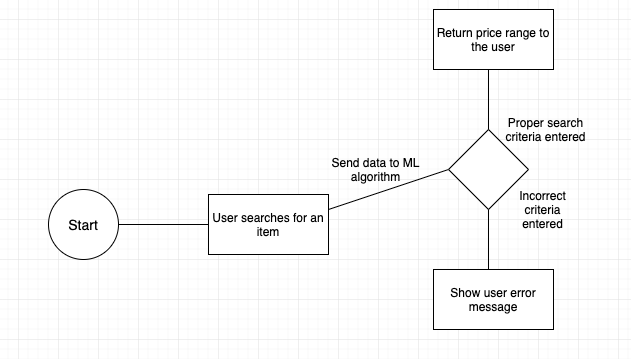
The interface of the project mainly dealt with our GUI and hosting. The GUI was developed using JavaScript with ReactJs as our choice of library. Some of the other dependencies of GUI included npm, webpack and other third-party styling libraries. Hosting and backend was mainly done on AWS with technologies such as dynamoDb and Node Js being used.

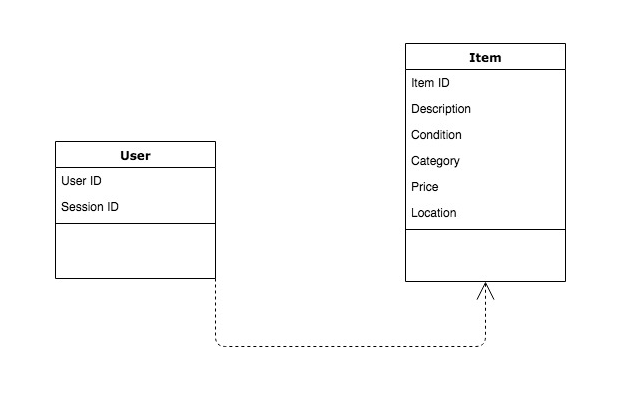
**2.2** **State-of-the-art**

According to a Deloitte Global research paper, the used cell phone market is one of the fastest growing markets which has an economic value of about 17 billion with an annual increase of about 80 million every year (Deloitte, 2017, p. 1). Our application provides an estimate on prices of the cell phone given the information about physical and internal attributes of the cell phone using machine learning algorithms on data from other advertisements on public marketplaces. Any product providing similar service for same domain i.e. cell phones is not yet available in the market. Dr. Zafar Khan and Dr. Asim, from University of Engineering and Technology, Lahore, has written a research paper which revolves around using classification to predict prices of new cell phones using dataset available on GSMarena.com (Asim, 2018, p. 2). Our solution solved the problem of underpaying or overpaying for a used product.

# Chapter 3 Project Requirements

**3.1 Domain and Business Requirements**







**3.2 System (or Component) Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | User story | Functionality | Degree of Necessity |
| 1 | As a user, I shall be able to browse through items so I get to see what items are available | Functional | Essential |
| 2 | As a user, I should be able to search for items so I can find things easier and faster. | Functional | Desired |
| 3 | As a user, I should be able to submit request for price estimate for items that are not available on the application | Functional | Desired |
| 4 | As a user, I shall be able to select items to find a price estimate. | Functional | Essential |
| 5 | As a user, I shall be able to provide parameters for the selected item. | Functional | Essential |
| 6 | As a user, I shall be able to get the price estimate as a price range for the item selected. | Functional | Essential |
| 7 | As a user, I should be able to upload pictures from the camera or gallery to provide condition of the item. | Functional | Desired |
| 8 | As a user, I should be able to use the picture recognition feature so I can easily provide condition of an item. | Functional | Desired |
| 9 | As a user, I shall be able to change the parameters provided by me at any time. | Functional | Essential |
| 10 | As a user, I should be able to create an account so I save my searches for the items. | Functional | Optional |
| 11 | As a user, I should be able to log into my account so I can access my information. | Functional | Optional |



**3.3 Non-functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | User story | Functionality | Degree of Necessity |
| 1 | As a user, I should have an overview of how to use the application on the first landing page. | Non- functional | Essential |
| 2 | As a user, I should get a new estimate every time some parameter is changed(auto- refresh). | Non-functional | Optional |
| 3 | As a user, I expect a simple user-friendly interface so I can navigate easily through the app. | Non-Functional | Desired |
| 4 | As a user, I need an application that takes under 15 seconds to give me results. | Non-Functional | Desired |
| 5 | As a user, I should have a secure authentication protocol so no one hacks my account. | Non-Functional | Desired |



* 1. **Context and Interface Requirements**

The project is designed and implemented by using the ReactJS framework for the front-end. ReactJS is dependent on Javascript and the MVC model. The back-end of the project is developed in python. The script for data collection as well as the machine learning algorithms are developed in python. In order for the user to be able to run the application, a Javascript enabled browser is required. The website will be responsive, meaning it will ensure the formatting is compatible with almost all screen sizes. Thus, allowing our application to run on smartphone browsers as well. In order for the client and server to properly we will be using NodeJS as our back-end service. This will allow the client to send information to the server so that the user can get the appropriate information.

* 1. **Technology and Resource Requirements**

# 

|  |  |
| --- | --- |
| **Requirements** | **Description** |
| Github | A GitHub repository will be used for code management |
| ReactJS/React-Native | Front-end framework for both web and mobile |
| MockFlow | Application used to create the wireframe for the prototype |
| MongoDB | Database will be used to store the data obtained from web scraping |
| AWS | Amazon AWS will be used to deploy the application |
| Python | Python libraries such as scikit, scrapy, and beautifulsoup will be used for data collection. We will also use testing libraries like unittest or pytest to test our software. |
| NodeJS | Back-end development language |
| Documentation | Documentation should be kept up to date with instructions on using our code. Libraries like pydoc or readthedocs will be used. |



# Chapter 4 System Design

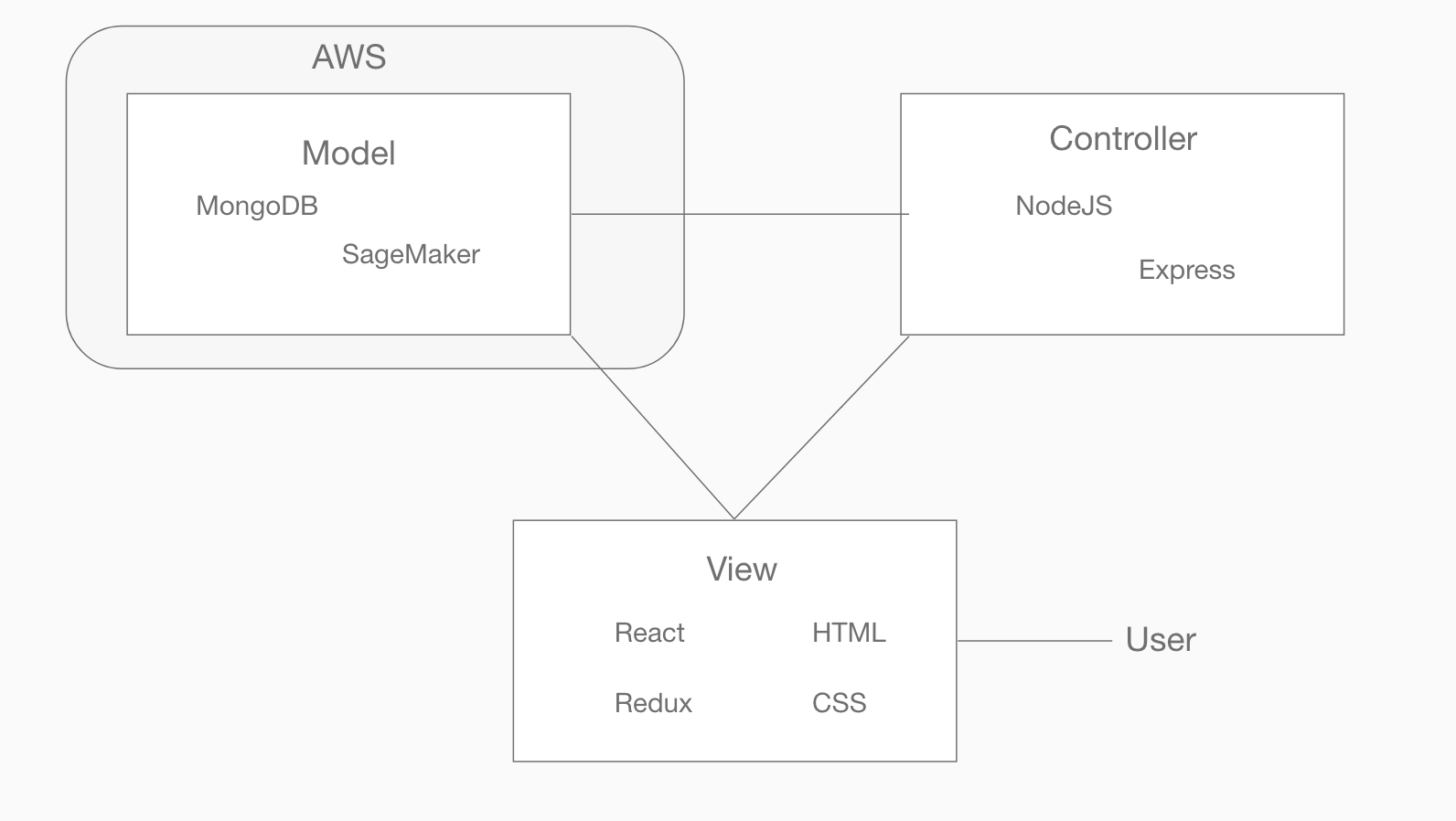
**4.1 Architecture Design**

We decided to use the MVC architecture better known as Model View Controller to implement our project. We are using React with Redux and Webpack for our View part of the implementation. Our Controllers are implemented using NodeJS and Express.JS. MongoDB and SageMaker encompases the Model part of the implementation. Since we are using React to implement our View section we will be further implementing a MVVM (Model View ViewModel) architecture to create high quality front end components that react to user response dynamically.

Our project aims to implement MVC in a client-server setting. where our web app is communicating with different server entities such as the database, the machine learning model or the web scraper. Our project is going to be hosted using AWS, which allows us to API’s such as SageMaker, MongoDB Atlas and DocumentDB.

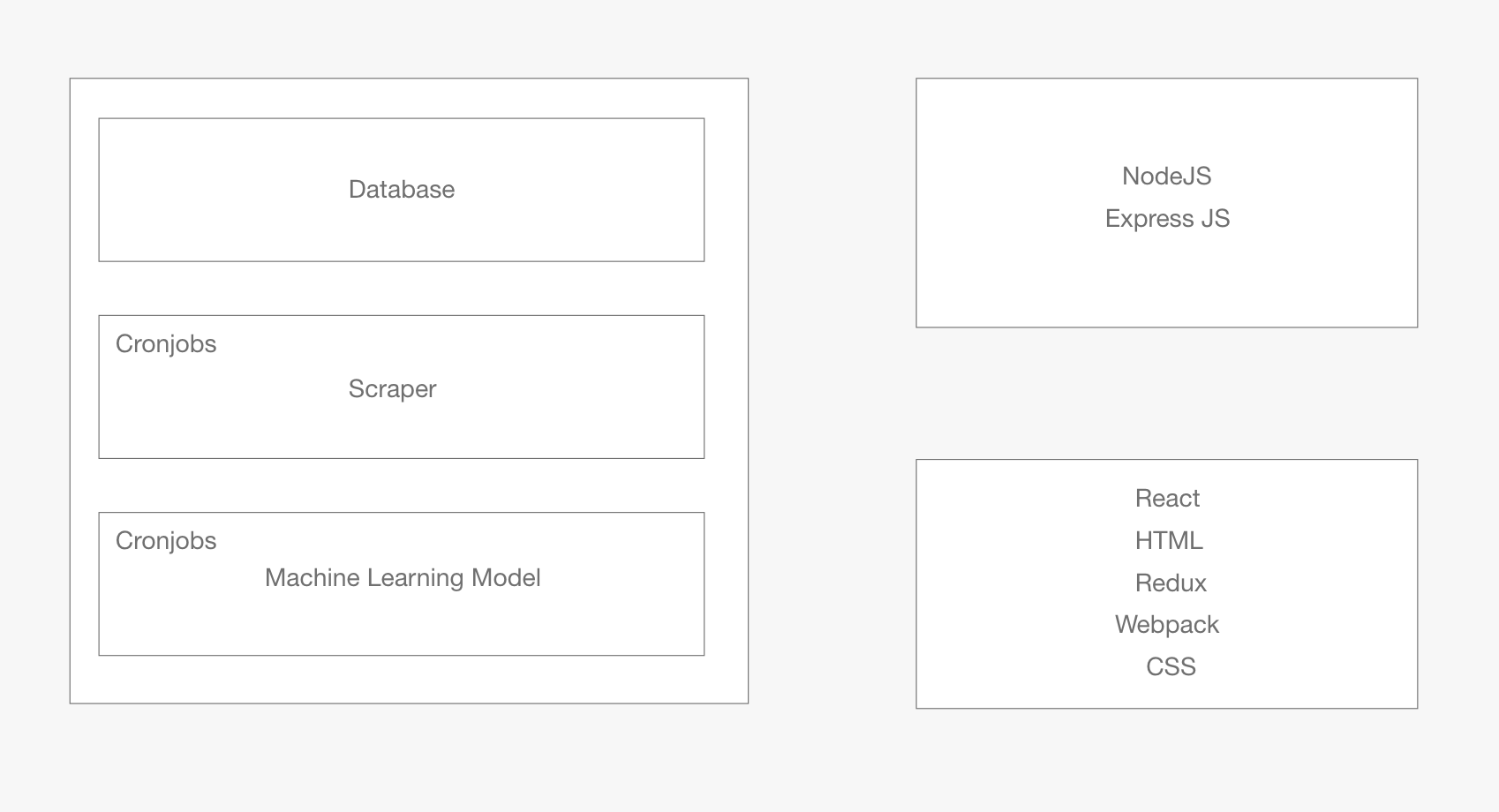
* 1. **Interface and Component Design**

The view is presented to users of the system, event handler within the view track user behavior and sends request to the controller. Upon receiving a request the controllers access or modify the model and depending on the users action the controller renders a suitable view update that displays a given version of the model. In the current implementation React with track user events and will pass on the actions to the NodeJS depending on what kind of an action NodeJS will either direct Express to make an API call or contact a backend entity by itself. Once this call has been processed, NodeJS either ends up manipulating the backend or calls for a render on Front end. If a render call is made React updates the ReactDOM which hot reloads components in the browser DOM without refreshing the page all while the state is being maintained by Redux.

****



* 1. **Structure and Logic Design**





Based on functionality the project is divided into three major parts on the backend: Database, Data Scraper and Machine Learning Lodel. Data flow happens between the frontend and the backend using the NodeJS connector which uses Express.JS. The Data Scraper and Machine Learning Model run on unix cron jobs to stay updated with the most recent data. Cron jobs is schedules once every 7 days to get a full data reset from most ad posting marketplaces.

* 1. **Design Constraints, Problems, Trade-offs, and Solutions**

In this section, we examine the design constraints considered when designing our software, and the trade-offs and solutions made to solve or work around those constraints.

**4.4.1 Design Constraints and Challenges**

Some of the design constraints involved in the design of our software included dealing with how advertisements were posted on online marketplaces, the vernacular used in different areas in the United States, the current culture of shopping online for used goods, etc.

For example, advertisements were posted in different ways depending on the online marketplace. In the case of CraigsList, users can post their device for sale and tag various metadata about the device, such as condition, color, carrier, operating system, or manufacturer, or the user may leave these tags out. This is different than on Ebay where the specification of the device are required for the posting. This led to our design focusing on pulling data from one source, cleaning the data as much as possible, and feeding the cleaned data into our machine learning model.

Another example, as we are pulling data from different areas in the United States, we have to deal with differences in language. As an example, several posts in the Bay Area tend to be less wordy and have far less detail with respect to the specification of the device. In contrast, posts in New York used more sophisticated language with more detail. Our design should be able to pull data from both areas of the United States accurately.

Another main design constraint on the technical side of things deals with the machine learning models. We tried to design a model that was efficient and produced accurate and precise results as much as possible. This was incredibly tough as used device prices varied heavily throughout different regions and with dirty data.

Finally, consideration was given to how the user interface of our application will present data to the user. A good user interface should present the result of the data computation succinctly and neatly. Additionally, a good user interface should be easy to use and intuitive.

**4.4.2 Design Solutions and Trade-offs**

To solve or work around some of these design constraints, our team designed changes to our software or developed new methods to clean data. For example, when facing the challenge of how advertisements were posted on CraigsList, we designed a way to pull as much data as possible from every post under the smartphone postings section. Then, we filled in data manually as much as possible before feeding the data into our machine learning model. This helped immensely in generated a model that could predict prices accurately and with great precision. This conveniently also helped with our main design goal of designing a machine learning model with accurate and precise results. The process of “cleaning data” was also necessary to decode how posts were created in various regions of the United States.

The user interface design constraint of our software was solved by working closely with all team members to design wireframes before implementing the final design. This allowed the team to provide continuous feedback as an ideal user interface was designed. Finally, when a user interface design was approved by all members of the team, it was implemented with Javascript tools as described later in this report.

# Chapter 5 System Implementation

**5.1 Implementation Overview**

Using Python and libraries like Scrapy, we designed an algorithm that scraps data from public marketplace Craigslist and puts that data into a CSV file. Running the algorithm on multiple locations we are able to gather about 10000 data points. Then, the K Nearest Neighbor(kNN) regressor algorithm, we are able to predict the price range of the cell phone which is being queried from the front end.

* 1. **Implementation of Developed Solutions**

We are using the K nearest neighbor as our choice of machine learning algorithm to process the data gained by scraping online marketplaces. The Kth nearest neighbor or kNN algorithm that can be used for classification on regression.

kNN identifies k nearest neighbors of an estimate point in a pool of pre existing training vectors. For example, after training the kNN model with k as 3, when we test it by predicting the price of a cell phone, it returns 3 nearest neighbors for that cell phone measuring the Euclidean distance between those points in a multidimensional space. Using the neighbor's prices, we can predict the price and the price range for the cell phone.

* 1. **Implementation Problems, Challenges, and Lesson Learned**

Implementation of our Machine Learning Model was hindered by the lack of a robust feature list that makes the process much easier. Our dataset consisted of non metric features such as color, brand and model name. Such features made it harder for us to write an algorithm that would either classify or perform regression. For our current iteration we are scraping values for first party description pages and utilizing some of the numeric data for instance the screen size of a phone, pixel density, battery capacity, etc.