Montclair State University
College of Science and Mathematics
Department of Computer Science
Masters Project

Visualizing Weather Financial Impact on Industries and Weather Derivatives

By:
Inga Bondarenko
Priyanka Phapale

Montclair State University
Advisor: Dr. Stefan Robila
Department of Computer Science
Montclair State University
Montclair, NJ

Table of Contents

Acknowledgments

First of all, we would like to express gratitude to Professor Stefan Robila for giving us the opportunity to work on this project. His guidelines for the strategy, support and suggestions helped us to work on this project with inspiration and complete it on time despite all of the challenges we faced and complexity of the implementation because of volume for the joint project.

We would also like to thank the graduate committee members, Professor John Jenq and Professor Mark Hubey, who have managed time to attend our project presentation, demo and being part of our graduate project committee members. We are also thankful for their guidelines and advice during project presentation.

Finally, we would like to thank all our professors from Computer Science Department, classmates and colleagues for the completion of our project.

Thank you,

Priyanka Phapale
Inga Bondarenko

Abstract

With the increasing availability of enormous datasets and processing power data analysis and visualization can be done efficiently. Along with this viewing it from different angles, and extracting the maximum available useful information out of it is a hot research topic. This is especially true for financial systems. Analyzing all available information and predicting short and long-term market trends with their fluctuations has been the cornerstone of trading strategies and giving investment advice for years.

One of the interesting financial instruments, which can potentially utilize big data analysis and algorithms in several different ways are weather derivatives. Their research combines both market and weather analysis. This is quite complex and fits perfectly as an area we can explore big data research capabilities.

The main idea of our project is to learn how to work with big data technologies and tools and familiarize ourselves with financial markets, futures trading, and weather derivatives specifically. In this project, we will show how weather derivatives relate to weather and how predicting the weather leads to predicting index values and therefore prices for weather derivatives. The second step for our work is to implement one of the existing algorithm for weather prediction and based on that, to implement calculations for the weather future prices for the chosen month.

All of these technologies and tools were new for us, and we started to learn them as beginners. It has also allowed us to work as a team and to learn about financial market tools and operations, which helps to increase our marketable skills in one of the most popular industries for software engineering.

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Our concept is to create an interface in the form of web application that allows easy visualization of historical weather data, hdd/cdd data and the relation between weather and weather derivatives using a graph reflecting the predicted weather and prices through the database. The interface is to represent the relation and correspondence between weather and weather derivatives(prices). This area is not widely accessible for educational purposes but very important in the financial market. So, we decided to implement it in a simpler and effortless way how it works for the public which is not familiar with this topic.

1.2. Project goal

The project is intended to complete the following tasks:

- To visualize the main idea of the implementation (will be discussed below in this section) using a graph based on the processed and analyzed Big Data.
- To implement an algorithm for weather forecasting using historical datasets for 40 years.
- To implement an algorithm for the weather derivatives (prices) calculations based on the forecasted weather in the previous step.
- To visualize one of the real correlation between weather and prices in the form of a graph to show the effectiveness of the implemented algorithms for forecasting.

2. Financial instruments and futures

Financial instruments are commercial contracts between parties. They can be created, traded, modified and settled. They can be cash (currency), evidence of an ownership interest in an entity (share), or a contractual right to receive or deliver cash (bond) [1].

Financial instruments can be either cash instruments or derivative instruments:

- Cash instruments – instruments whose value is determined directly by the markets. They can be securities, which are readily transferable, and instruments such as loans and deposits, where both borrower and lender have to agree on a transfer.

- Derivative instruments – instruments which derive their value from the value and characteristics of one or more underlying entities such as an asset, index, or interest rate. They can be exchange-traded derivatives and over-the-counter (OTC) derivatives [2].

A future is a type of the financial instruments. A futures contract (more colloquially, futures) is a standardized forward contract, a legal agreement to buy or sell something at a predetermined price at a specified time in the future [3].

What is a weather future?

A weather future is a type of weather derivative that obligates the buyer to purchase the value of the underlying weather index - measured in heating degree days (HDD) or cooling degree days (CDD) - at a future date. The settlement price for the underlying weather index is equal to the value of the relevant month's HDD/CDD multiplied by $20. Weather futures enable businesses to protect themselves against losses caused by unexpected shifts in weather conditions [4]. In this project, we work with weather futures, but the principles are very similar to any index futures, like S&P, NASDAQ, etc. Below, in Figure 1, is an example, how weather futures are related to weather (temperature). All the calculations for the graph are in Table 1.

Date	NY HDD F18	hdd	Sum to date	Normal hdd	Normal sum	Naïve visualizati	Final price
1/1/2018	1095	52	**52**	31.5	31.5	1023	1033
1/2/2018	1113	45.5	**97.5**	31.5	63	1037	1033
1/3/2018	1103	42	**139.5**	31.5	94.5	1047.5	1033
1/4/2018	1106	41	**180.5**	32.5	127	1056	1033
1/5/2018	1070	51	**231.5**	32.5	159.5	1074.5	1033
1/6/2018	1070	55.5	**287**	32.5	192	1097.5	1033
1/7/2018	1070	53.5	**340.5**	32.5	224.5	1118.5	1033
1/8/2018	1067	41	**381.5**	32.5	257	1127	1033
1/9/2018	1065	28	**409.5**	32.5	289.5	1122.5	1033
1/10/2018	1073	28.5	**438**	32.5	322	1118.5	1033
1/11/2018	1077	18	**456**	32.5	354.5	1104	1033
1/12/2018	1076	12.5	**468.5**	32.5	387	1084	1033
1/13/2018	1076	26.5	**495**	32.5	419.5	1078	1033
1/14/2018	1076	45	**540**	32.5	452	1090.5	1033
1/15/2018	1076	42	**582**	32.5	484.5	1100	1033
1/16/2018	1056	34.5	**616.5**	32.5	517	1102	1033
1/17/2018	1065	36	**652.5**	32.5	549.5	1105.5	1033
1/18/2018	1059	40	**692.5**	33	582.5	1112.5	1033
1/19/2018	1042	32.5	**725**	32.5	615	1112.5	1033
1/20/2018	1042	20	**745**	32.5	647.5	1100	1033
1/21/2018	1042	19	**764**	32.5	680	1086.5	1033
1/22/2018	1035	21.5	**785.5**	32.5	712.5	1075.5	1033
1/23/2018	1038	16	**801.5**	32.5	745	1059	1033
1/24/2018	1038	29.5	**831**	32.5	777.5	1056	1033
1/25/2018	1054	36.5	**867.5**	32.5	810	1060	1033
1/26/2018	1054	33	**900.5**	32.5	842.5	1060.5	1033
1/27/2018	1054	19.5	**920**	32	874.5	1048	1033
1/28/2018	1054	14.5	**934.5**	32	906.5	1030.5	1033
1/29/2018	1054	24	**958.5**	32	938.5	1022.5	1033
1/30/2018	1054	35	**993.5**	32	970.5	1025.5	1033
1/31/2018	1054	39.5	**1033**	32	1002.5	1033	1033
2/1/2018	1054						

Table 1. Correlation between weather (temperature) and weather derivatives shown in Figure 1.

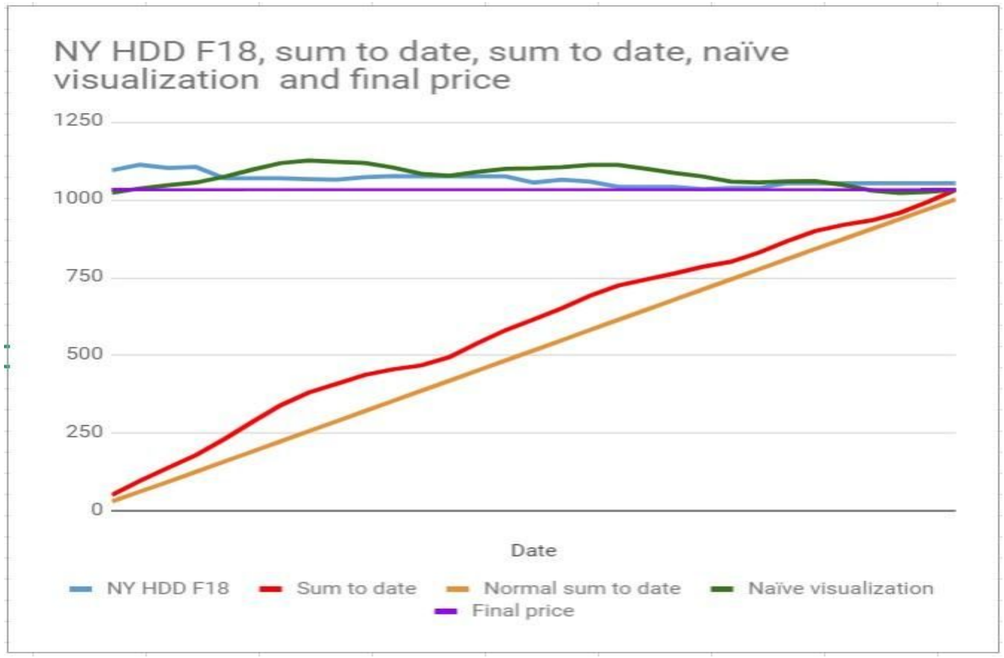


Figure1. The Correlation between weather (temperature) and weather derivatives indexes.

Actual sum to date (red line):

Calculates all HDD (cdd) values by adding a next day value to the previous sum (during one month). Month to date HDD/CDD sum. It is essential to value because the sum to date at the end of the month defines the final index value (as reported by MDA Information Systems [5]) and on the graph, it converges with the final predicted value (the last point on the green line). It may not completely unite with the real price because the bidding may stop a few days before based on the sum to date and a short-term weather forecast.

Standard sum to date (yellow line):

Historical month to day sum. Calculates from all-weather datasets available. It is not very important for us, but it was used for our original estimate and also the first reference point for early trades (several months before the index period starts).

Final price (purple line):

Month-end index value. Equals the last number of our red line. Any source will confirm it. It is also a perfect estimate which we should reach.

NY HDD F18 (blue line):

Actual index prices on CME exchange. In our case, this is copied data from Bloomberg. It was not traded the last few days of the month (Jan particularly). So the light blue line doesn't converge with the final HDD number. The final settlement price for NY Jan for the all futures contracts remaining open at the termination of trading is settled using the CME Degree Days Index reported by MDA Information Systems, Inc. [5][6].

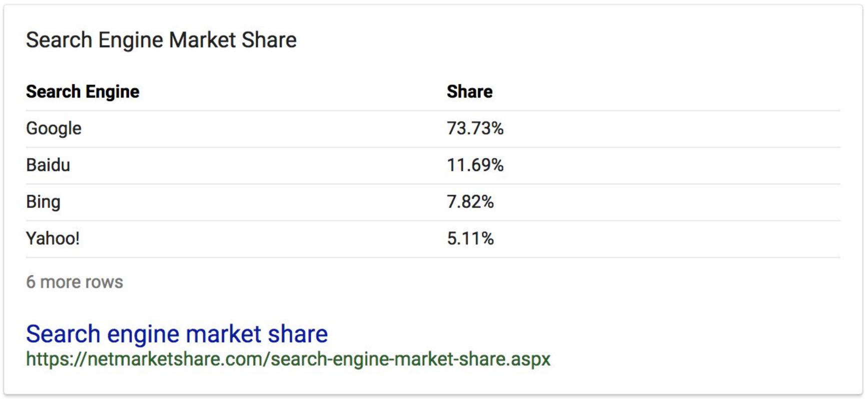
Naive visualization (green line):

The first naive estimate (prediction) calculates from the replacement for any missing date with average historical value. The closer the distance between the green line and purple the more accurate algorithm is. Our task is to make the green line closer to the purple than the blue line. Taking into account weather forecasts will help to make this estimate closer.

3. Literature review

3.1. Current market demand review

To be able to evaluate market demand we used Google Adwords tool to create a hypothetical pay-per-click campaign to analyze how many visitors we will have, which could be converted into clients in case of the project promotion. It shows us if the project has any sense in the commercial world. Google Adwords is an online advertising service developed by Google, where advertisers pay to display brief advertising copy, product listings, and video content within the Google ad network to web users [7]. Google is a most popular Search Engine machine on the market. It shares more than 73% of the market (Figure 2) and, in this case, we think that it contains a sufficient amount of the data, numbers, prices, etc. to analyze.



Search Engine Market Share

Search Engine	Share
Google	73.73%
Baidu	11.69%
Bing	7.82%
Yahoo!	5.11%

6 more rows

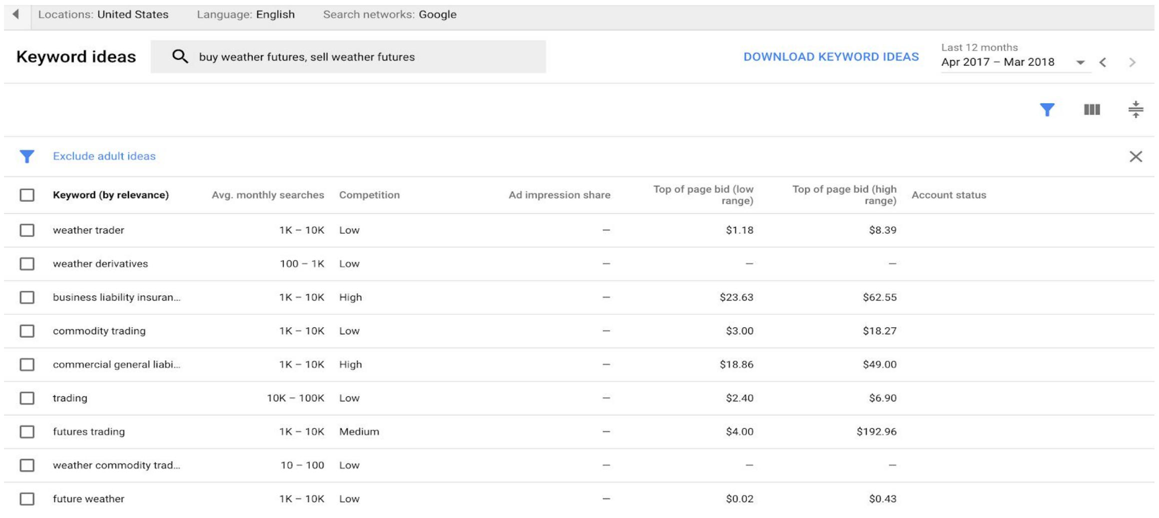
Search engine market share
https://netmarketshare.com/search-engine-market-share.aspx

Figure 2. Search Engine Market Share.

The next step is to choose commercial keywords to start collecting all possible keywords and combinations related to this topic, which Google collected over the years and stores for any analytical purposes. We choose keywords "buy weather futures," "sell weather futures" ("buy" and "sell" are explicit commercial additions to any word, if you are doing industrial research). Below is a screenshot of the first relevant results we have collected (Figure 3) with average monthly searches, competition, average prices, etc.

As we can see from keywords and numbers in below Figure 3, we can make an initial conclusion that users are looking for particular information related not just about what weather derivatives are, but also trading related information, futures trading, commodity trading (prices of the commodities also have a relation for a weather forecasting). Competition rate and prices above zero show that there are at least several competitive campaigns exist in Google AdWords system. It brings us to the next step when we can finalize our campaign and evaluate our possible expenses and conversion if we needed to decide to launch this project as a business idea.

Below is a chart (Figure 4) which show that we can get more than 250K impressions and 10K clicks, which means we will have 10K visitors to our project with $14K investment per month. It also gives us additional information to consider. For example, we should take in the account the devices listed on the figure, and it means that we should adjust our website to all of them to make it more usable and not to distract visitors.



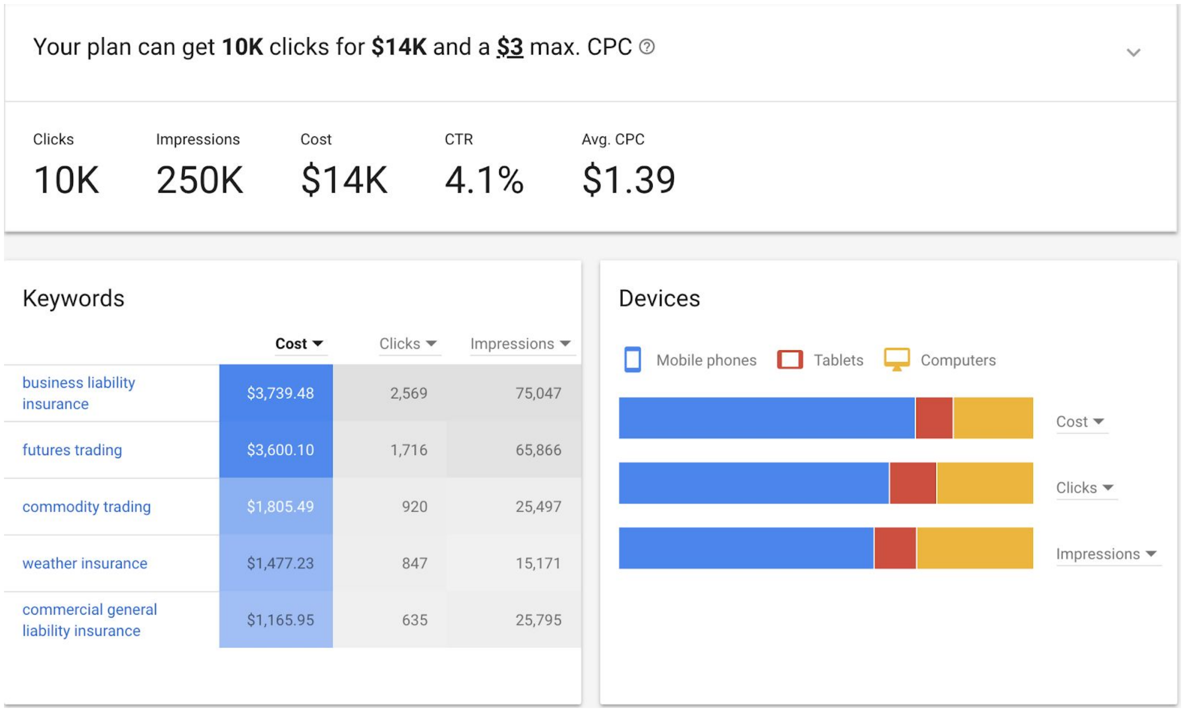
Locations: United States Language: English Search networks: Google

Keyword ideas buy weather futures, sell weather futures DOWNLOAD KEYWORD IDEAS Last 12 months Apr 2017 – Mar 2018

Exclude adult ideas

Keyword (by relevance)	Avg. monthly searches	Competition	Ad impression share	Top of page bid (low range)	Top of page bid (high range)	Account status
weather trader	1K – 10K	Low	–	$1.18	$8.39	
weather derivatives	100 – 1K	Low	–	–	–	
business liability insuran...	1K – 10K	High	–	$23.63	$62.55	
commodity trading	1K – 10K	Low	–	$3.00	$18.27	
commercial general liabi...	1K – 10K	High	–	$18.86	$49.00	
trading	10K – 100K	Low	–	$2.40	$6.90	
futures trading	1K – 10K	Medium	–	$4.00	$192.96	
weather commodity trad...	10 – 100	Low	–	–	–	
future weather	1K – 10K	Low	–	$0.02	$0.43	

Figure 3. Screenshot of the first relevant results for the keywords "buy weather futures," "sell weather futures."



Your plan can get **10K** clicks for **$14K** and a **$3** max. CPC

Clicks	Impressions	Cost	CTR	Avg. CPC
10K	250K	$14K	4.1%	$1.39

Keywords

	Cost	Clicks	Impressions
business liability insurance	$3,739.48	2,569	75,047
futures trading	$3,600.10	1,716	65,866
commodity trading	$1,805.49	920	25,497
weather insurance	$1,477.23	847	15,171
commercial general liability insurance	$1,165.95	635	25,795

Devices

Figure 4. Approximate calculations for the AdWords campaign.

If we take into the account the statistical conversion of visitors to possible customers, up to 2-3%, we get after computing about 200-300 (optimistically) clients per site per month for all chosen combinations of the keywords. In our case, we decided just general words excluding city related keywords, which can add more visitors and investments, of course. In the end, we can use the final number of possible clients to calculate the price of our online service. Therefore, we see that all the numbers and calculations give us positive figures which prove that we have demand.

3.2. Weather prediction algorithms review

Weather forecasting is the application of current technology and science to predict the state of the atmosphere for a future time and a given location. Weather forecasts are made by collecting as much data as possible about the current state of the atmosphere (particularly temperature, humidity, and wind) and using an understanding of atmospheric processes (through meteorology) to determine how the atmosphere evolves in the future. During the data assimilation process, information gained from the observations is used in conjunction with a numerical model's most recent forecast for the time that comments were made to produce the meteorological analysis. Mathematical weather prediction models are computer simulations of the atmosphere. In our case, we were looking for the algorithms which could be implemented by using the technologies we are learning as beginners at the allotted time. After some research, we decided to take a close look at three algorithms to pick one of them for the implementation: Linear Regression, a variation of Functional Regression and an algorithm which is based on Artificial Neural Networks Theory. All of them can be implemented by using Python which we chose as the primary programming language for our server-side scripts.

After reviewing few research paper, we decided to move forward with Linear Regression algorithm. The most important document which we used as a base for our decision [8], is examining machine learning algorithms, linear regression and a variance of the functional regression by implementing them and making similar calculations of effectiveness, as well as giving a brief description of the Neural and Bayesian networks. As stated in conclusion: "both linear regression and functional regression were outperformed by professional weather forecasting services, although the discrepancy in their performance decreased significantly for later days, indicating that over longer periods of time, our models may outperform professional ones. Linear regression proved to be a low bias, high variance model whereas functional regression proved to be a high bias, low variance model. Linear regression is inherently a high variance model as it is unstable to outliers, so one way to improve the linear regression model is by a collection of more data. Functional regression, however, was high bias, indicating that the choice of model was poor and that further collection of data cannot improve its predictions. This bias could be due to the design choice to forecast weather based upon the weather of the past two days, which may be too short to capture trends in weather that functional regression requires. If the forecast depends on the weather of the past four or five days, the bias of the functional regression model could likely be reduced"[8]. Therefore, the Linear Regression became our choice.

The professional weather forecasting service consistently outperforms all of the models we are exploring. But the main idea for the project is, first of all, to learn new technologies in the areas where we are just beginners, the second is to implement at least one of the existing algorithms and complete the application in the allotted time.

3.3. Prices calculation algorithm description

For HDD/CDD weather derivatives the price (index value) is directly related to the temperature. The accumulation period (discussed above in section 1.2) of each contract begins on the first calendar day of the contract month and ends on the last calendar day of the contract month. The heating degrees for a given day are the total number of degrees that the average outside air temperature drops below the base temperature of 65°F (or 18°C for non-U.S. cities) [9].

4. Datasets

4.1. Weather Dataset

Weather dataset includes many facts and numbers about the state of the atmosphere, including temperature, wind speed, rain or snow, humidity, and pressure. These days, we have some fantastic ways to collect this kind of data. We have high-tech equipment that can measure everything with reasonable accuracy. And, we can estimate it from all sorts of places: the ground, the air, and even from space. To predict the weather, we need to measure the weather. If you want to know what is weather like tomorrow, it's pretty significant to know what the weather was like today and yesterday. Knowing the average weather on a particular day of the year is also useful. Collecting data every day can show you patterns and trends, and help you figure out how our atmosphere works.

The primary dataset which we have used for our project is the weather data for last 40 years. We have gathered this data from "Weather Underground [10]". This website provides historical weather data, local and long-range weather forecasts, weather reports, maps and tropical weather conditions for locations worldwide. We have used python script (will be described in section 4.2, Data Extraction) to read each day data and collected {'Date','City Code','Temp mean','Temp min','Temp max','HDD','Dew point','Humidity Avg','Humidity max','Humidity min','Precipitation','Sea level pressure','Wind speed avg','Wind speed max','Visibility','Events'} the following parameters. The data is collected for the following 15 cities Atlanta, Chicago, Cincinnati, Dallas, Des Moines, Houston, Kansas City, Las Vegas, Minneapolis, New York, Philadelphia, Portland, Sacramento, and Tucson. After visiting and copying datasets from Bloomberg terminal (described below in section 1.5), we found out that the price datasets are available only for eight cities. So, we have decided to work only with weather datasets for these particular cities, which are listed further in Table 2. The description which is given in this table relates to the files after data cleaning, which will be described further in section 4.3.

Using our current script it took us a while to download the historical data, but in future, we are planning to improve our script for data download or for bigger purposes we can automate the script for better use.

Below is a table with a list of all collected datasets for weather information.

File Name	Size	Description
Atlanta.csv	1.088 MB	Data are available starting from 08/05/1981. For the earliest periods of time only rare portions are available.
WeatherData_Chicago.csv	894 KB	An error occurred during parsing and cleaning. Not resolved.
Cincinnati.csv	1.248 MB	Data are available starting from 01/01/1977 until 05/04/2018
Dallas.csv	1.290 MB	Data are available starting from 01/01/1977 until 05/04/2018
LasVegas.csv	610 KB	Data are available starting 07/01/1981. For the earliest periods of time only rare portions are available.
Minneapolis.csv	808 KB	Data are available starting from 01/01/1977. It has a missing period from 11/02/1983 - 09/10/1997.s
NewYork.csv	1.101 MB	Data are available starting from 01/01/1977 until 05/04/2018
Sacramento.csv	1.270 MB	Data are available starting from 01/01/1977 until 05/04/2018

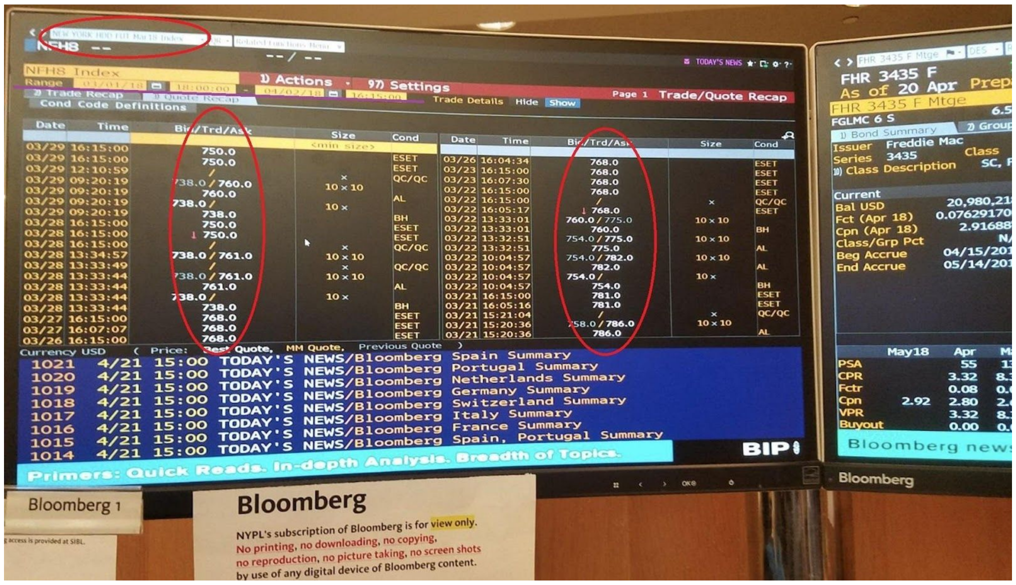
Table 2. The Weather datasets.

4.2. Weather derivatives dataset

Weather derivatives dataset is needed for the final comparison for our prices calculations which will make based on the forecasted weather. This data is not available to the wide public because of the specific of the information. The only one source is Bloomberg terminal [11] which is available to the licensed companies or institutions, such as banks, for example. Another option to use the Terminal is a donated station to the New York Public Library located in Manhattan, 188 Madison Ave, New York, NY 10016 (Figure 5). No download was allowed, so we copied everything manually (Table 3). The data was available only for HDD futures starting December until the last day of our visit, which is April, 20. Manual for the instructions is available on the Yale University website [12].

Steps to obtains weather derivatives datasets from the Bloomberg Terminal include the following:

1. Go to the Terminal and use WEAT command [13] to get to the Main Menu of Bloomberg Functions > Commodity Market > Weather
2. In the search bar type City and HDD (CDD) fut, month and year command and hit the search button
3. We are getting to the Main Menu > Indices > Index Derivatives > Analyze New York HDD Fut March 18 Index
4. Choose Trade/Quote Recap
5. In the filter, pick the date range and hit the button "Go" and we are getting a table with all the prices



```
NEW YORK HDD FUT Mar18 Index
NFH8 --
                         --/ --
NFH8 Index                  1) Actions  97) Settings
Range 03/01/18  18:00:00 - 04/02/18  16:15:00       Page 1  Trade/Quote Recap
2) Trade Recap  3) Quote Recap                      Trade Details  Hide  Show
Cond Code Definitions

Date  Time      Bid/Trd/Ask     Size      Cond   Date  Time      Bid/Trd/Ask    Size     Cond
                                <min size>
03/29 16:15:00        750.0               ESET   03/26 16:04:34        768.0             ESET
03/29 16:15:00        750.0               ESET   03/23 16:15:00        768.0             ESET
03/29 12:10:59         /          ×       QC/QC  03/23 16:07:30        768.0             ESET
03/29 09:20:19  738.0 / 760.0     10 × 10        03/22 16:15:00        768.0             ESET
03/29 09:20:19        760.0               AL     03/22 16:15:00         /         ×      QC/QC
03/29 09:20:19  738.0 /           10 ×           03/22 16:05:17        768.0             ESET
03/29 09:20:19        738.0               BH     03/22 13:33:01  760.0 / 775.0    10 × 10
03/28 16:15:00        750.0               ESET   03/22 13:33:01        760.0             BH
03/28 16:15:00        750.0               ESET   03/22 13:32:51  754.0 / 775.0    10 × 10
03/28 16:15:00         /          ×       QC/QC  03/22 13:32:51        775.0             AL
03/28 13:34:57  738.0 / 761.0     10 × 10        03/22 10:04:57  754.0 / 782.0    10 × 10
03/28 13:33:49         /          ×       QC/QC  03/22 10:04:57        782.0             AL
03/28 13:33:44  738.0 / 761.0     10 × 10        03/22 10:04:57  754.0 /          10 ×
03/28 13:33:44        761.0               AL     03/22 10:04:57        754.0             BH
03/28 13:33:44  738.0 /           10 ×           03/21 16:15:00        781.0             ESET
03/28 13:33:44        738.0               BH     03/21 16:05:16        781.0             ESET
03/27 16:15:00        768.0               ESET   03/21 15:21:04         /         ×      QC/QC
03/27 16:07:07        768.0               ESET   03/21 15:20:36  758.0 / 786.0    10 × 10
03/26 16:15:00        768.0               ESET   03/21 15:20:36        786.0             AL
Currency USD   ( Price: Best Quote,  MM Quote,  Previous Quote )
1021  4/21 15:00 TODAY'S NEWS/Bloomberg Spain Summary
1020  4/21 15:00 TODAY'S NEWS/Bloomberg Portugal Summary
1019  4/21 15:00 TODAY'S NEWS/Bloomberg Netherlands Summary
1018  4/21 15:00 TODAY'S NEWS/Bloomberg Germany Summary
1017  4/21 15:00 TODAY'S NEWS/Bloomberg Switzerland Summary
1016  4/21 15:00 TODAY'S NEWS/Bloomberg Italy Summary
1015  4/21 15:00 TODAY'S NEWS/Bloomberg France Summary
1014  4/21 15:00 TODAY'S NEWS/Bloomberg Spain, Portugal Summary
Primers: Quick Reads. In-depth Analysis. Breadth of Topics.
```

```
FHR 3435 F Mtge
FHR 3435 F
As of 20 Apr
FHR 3435 F Mtge
FGLMC 6 S
1) Bond Summary
Issuer  Freddie Mac
Series  3435        Class
10) Class Description
Current
Bal USD
Fct (Apr 18)
Cpn (Apr 18)
Class/Grp Pct
Beg Accrue
End Accrue
         May18   Apr
PSA               55
CPR             3.32
Fctr            0.08
Cpn      2.92   2.80
VPR             3.32
Buyout          0.00
```

Bloomberg

NYPL's subscription of Bloomberg is for view only.
No printing, no downloading, no copying,
no reproduction, no picture taking, no screen shots
by use of any digital device of Bloomberg content.

Figure 5. Bloomberg terminal in the New York Public Library. Screenshot of the real prices on the screen.

File Name	Size	Description
HDDAllCleaned.xlsx	408 KB	File contains all the prices for HDD available from December until April 20 for eight cities.

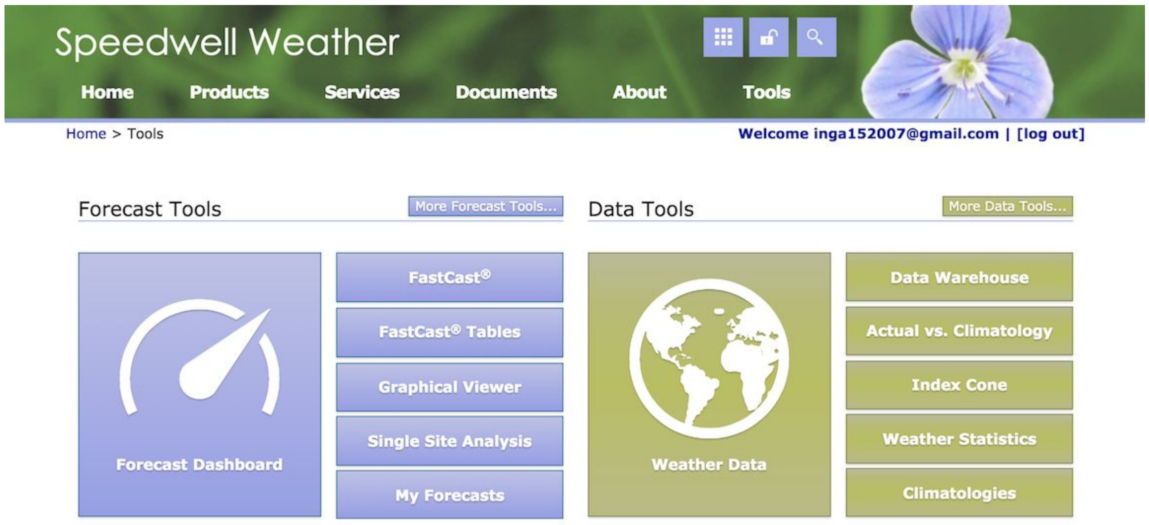
Table 3. Weather derivatives dataset.

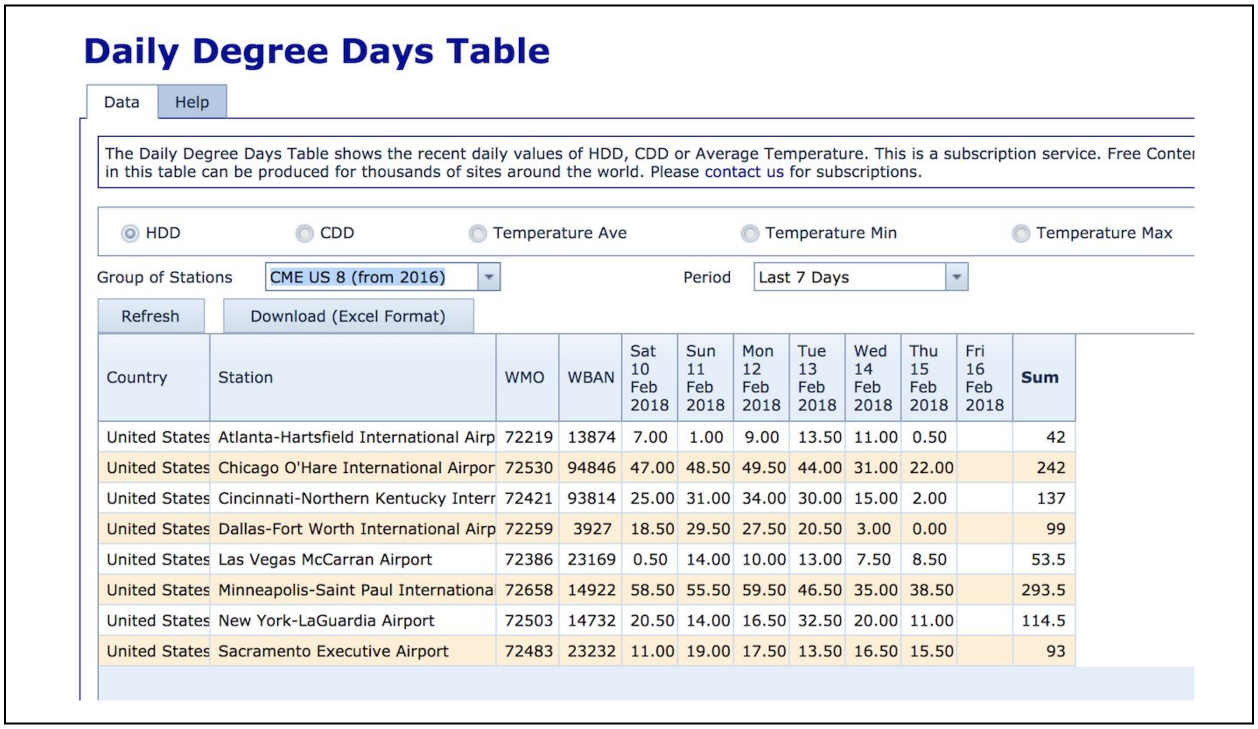
5. Similar solutions in the market

As we mentioned above, in section Motivation, we are working with a specific area, which is not widely popular but very important in the financial industry. The more accuracy you get at forecasting weather and prices means fewer risks in economic losses. We found several articles related to weather forecasting models and discussions about prices prediction, but we noticed that not so many websites or tools are publicly available for indexes prediction for free. The reason for this could be that this is internal commercial information for all the companies which are working in this industry.

Despite all the difficulties in this research, we were able to find at least one commercial website which provides similar services and products - Speedwell Weather [22]. Founded in 1999, Speedwell Weather provides quality weather data, weather forecasts, software, and consultancy. From offices in the UK and the USA, they serve clients worldwide in sectors including weather risk, energy, and agriculture. Our data products include SuperPack® which provides unlimited access to our thousands of high-quality worldwide weather data sets. Speedwell Weather is the dominant settlement agent for parametric weather risk contracts. Speedwell supplies quality historical and real-time weather data for tens of thousands of sites across the globe. They have direct data supply agreements with a wide range of national meteorological services.

After exploring the website, we have decided that it could be taken as a prototype for the initial UI design (Table 5).

Main Page example
 Speedwell Weather Home Products Services Documents About Tools Home > Tools Welcome inga152007@gmail.com | [log out] Forecast Tools More Forecast Tools... Data Tools More Data Tools... Forecast Dashboard FastCast® FastCast® Tables Graphical Viewer Single Site Analysis My Forecasts Weather Data Data Warehouse Actual vs. Climatology Index Cone Weather Statistics Climatologies
Daily Degree Days Table example



Daily Degree Days Table

The Daily Degree Days Table shows the recent daily values of HDD, CDD or Average Temperature. This is a subscription service. Free Conter
in this table can be produced for thousands of sites around the world. Please contact us for subscriptions.

[✓ HDD] CDD Temperature Ave Temperature Min Temperature Max

Group of Stations: CME US 8 (from 2016)

Period: Last 7 Days

Country	Station	WMO	WBAN	Sat 10 Feb 2018	Sun 11 Feb 2018	Mon 12 Feb 2018	Tue 13 Feb 2018	Wed 14 Feb 2018	Thu 15 Feb 2018	Fri 16 Feb 2018	**Sum**
United States	Atlanta-Hartsfield International Airp	72219	13874	7.00	1.00	9.00	13.50	11.00	0.50		42
United States	Chicago O'Hare International Airpor	72530	94846	47.00	48.50	49.50	44.00	31.00	22.00		242
United States	Cincinnati-Northern Kentucky Interr	72421	93814	25.00	31.00	34.00	30.00	15.00	2.00		137
United States	Dallas-Fort Worth International Airp	72259	3927	18.50	29.50	27.50	20.50	3.00	0.00		99
United States	Las Vegas McCarran Airport	72386	23169	0.50	14.00	10.00	13.00	7.50	8.50		53.5
United States	Minneapolis-Saint Paul Internationa	72658	14922	58.50	55.50	59.50	46.50	35.00	38.50		293.5
United States	New York-LaGuardia Airport	72503	14732	20.50	14.00	16.50	32.50	20.00	11.00		114.5
United States	Sacramento Executive Airport	72483	23232	11.00	19.00	17.50	13.50	16.50	15.50		93

Table 4. Screenshots of the pages for the UI prototype of our application.

The access was limited to ten days, and we were not able to get full access to all the indexes related data they forecast because of a very high price for the subscription. For example, the cost of historical forecasted weather is $175 per location, so for 15, the total would be $2,625.

6. Project Requirements

The project has some requirements to be addressed. These include system requirements, as well as specifications regarding functionality, usage, database design (discussed in section 4.4) and interface design (discussed in section 4.5).

6.1. System Requirements

For the implementation this project uses the following technologies and tools:

- AMPPS is a solution stack of Apache, MySQL, MongoDB, PHP, Perl and Python for Windows NT, Linux and macOS [23].
- IntelliJ Idea is a Java integrated development environment for developing computer software. IntelliJ Idea is also useful for editing Angular project files [24].
- Anaconda is a freemium open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment [25].

6.2. Specifications

The functionality of the project is listed below in Table 7.

Function	Description
Main Page	Main page is a landing page for the users. The main page welcomes users and has a general description of the project. It contains a menu for the next pages: weather stations and related cities, a graph page which shows a relation between weather and weather futures, page with the weather forecast and a correlation graph with a historical data, pages for indexes calculation and a correlation graph with a historical data.
Weather Station	The Weather Station page displays chosen city graph with temperatures, allows user to apply the filter, which includes a date picker for a month and a city, and to have a description/help for a visitor below the chart.
Weather vs. Weather Derivative	The Weather vs. Weather Derivative section displays the graph of relation between weather and weather futures with applied filters,

	which is discussed above in section 1.2.
Weather Forecast	The Weather Forecast page has a filter for a month for which we want to apply our forecasting algorithm. It implements Linear Regression algorithm which is discussed below in below section 7.7.
Forecasted Weather vs. Historical Weather	All of the calculations are displayed in the form of a graph and some description to understand the graph.
Prices calculation	The price calculation section has a description of the method in use and contains a chart for the result representation.
Predicted prices vs. Historical prices vs. Current prices vs. Predicted temperatures (sum to date)	It is displayed in the form of a graph with the description of each component.
HDD/CDD page	This page contains a filter, where a user can pick a month, and a type for the derivative (HDD or CDD), a city from the list and then a user is able to get related information in the form of a table.

Table 5. The initial objectives.

6.3. Users

The project is intended to be a simple web application for the wide public. The basic idea is to understand how a financial tool, in our case weather derivatives, works, then to describe in a simple way the basic principles, implement existing algorithms and visualize them in the form of graph or charts. So, any user of this website is able to get access to any information published without any registration. All information changes, as well as code implementation, will be under control of the developers.

6.4. Technology review

Python programming language for Big data processing

Python is an interpreted high-level programming language for general-purpose programming. Python has a design philosophy that emphasizes code readability and a syntax that allows programmers to express concepts in fewer lines of code [14]. Python is relatively easy to learn and also has the distinct advantage of being easy to read by humans. Python is also good for small- or medium-scale projects to build models and analyze data, especially for fast startups or small teams [15]. There are so many discussions that exists as to which language is better for big data processing. Python is one of the most popular among them (R, Scala, Java, etc.). Another reason to choose Python is that it is one of the languages in demand in the market of software developers.

For our needs, we chose Pandas software library written for the Python programming language for data manipulation and analysis. Library features include the following functionalities and some of we were using in our scripts [16]:

- The data frame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.

MEAN Stack for web development

MEAN is a collection of JavaScript-based technologies: MongoDB, Express.js, Angular, and Node.js (first capital letters of each technology combine in the word “MEAN”), are used to develop web applications from the client and server sides to databases. MEAN is a full-stack development toolkit (Table 4).

MEAN Component	Description
MongoDB	a NoSQL Database [17]
Express.js	a web application framework for Node.js [18]
Angular	is a TypeScript-based open-source front-end web application platform [19]
Node.js	an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side [20]

Table 6. MEAN stack components.

One of the most important functions which is implemented when working with Big Data is visualization. Data visualization is the presentation of data in a pictorial or graphical format. It enables decision-makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns. With interactive visualization, we can take the concept a step further by using technology to drill down into charts and graphs for more detail, interactively changing what data you see and how it’s processed. For that reason, we choose Chart JS, a powerful data visualization library [21], which can be used with a combination of MEAN Stack.

7. Application Development

7.1. Source Control

As a source control for our project, we choose GitHub [26], which is a web-based hosting service for version control which uses Git [27]. We need to store two separate folders for the project like backend (python code) and frontend (MEAN stack) which are not connected directly to each other. So, for the initial commit we used the command line with the next steps from the original GitHub website[28]:

- Change the current working directory to your local project (parent directory for the backend and frontend directories).
- Initialize the local directory as a Git repository (git init).
- Add the files to your new local repository. This stages them for the first commit (git add .).
- Commit the files that you've staged in your local repository (git commit -m "First commit").
- Copy the remote repository URL.
- In Terminal, add the URL for the remote repository where your local repository will be pushed (git remote add origin remote repository [29]; git remote -v).
- Push the changes in your local repository to GitHub (git push -u origin master).

Project name:	Visualizing Weather Financial Impact on Industries and Weather Derivatives
GitHub link:	https://github.com/ingamontclair/Volunteer
Contributors:	Inga Bondarenko, Priyanka Phapale

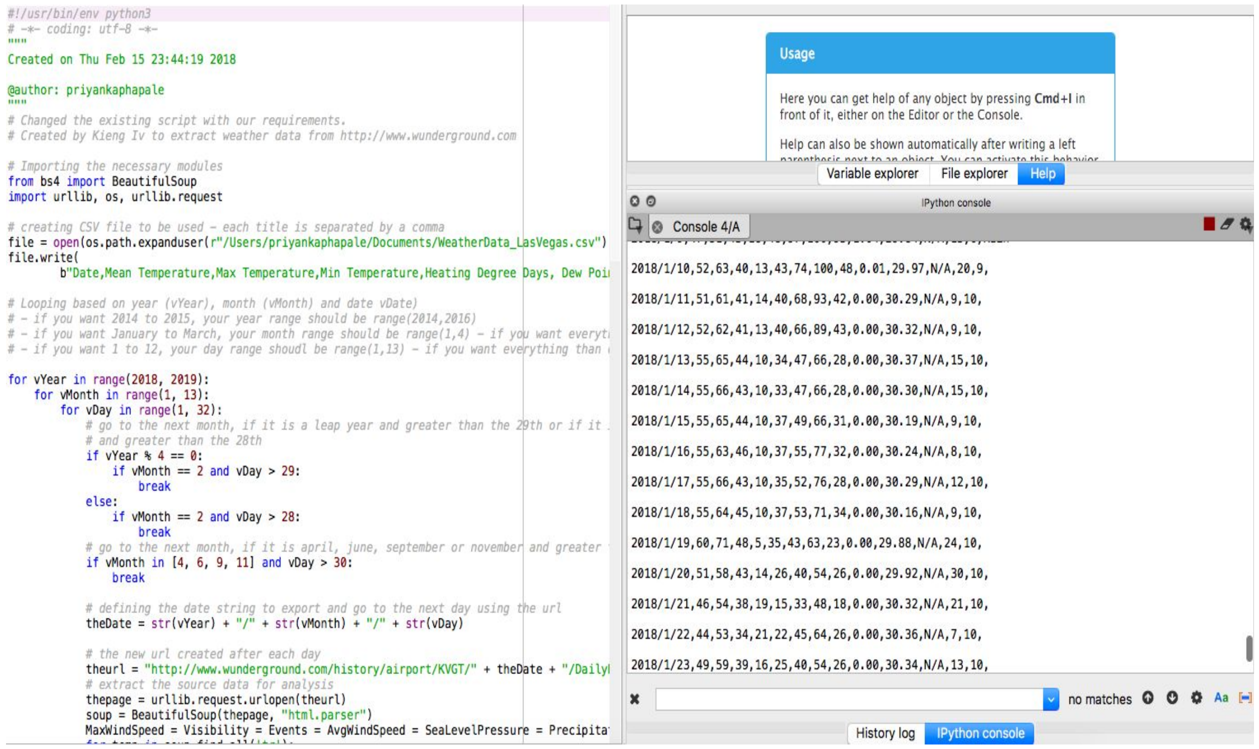
Table 7. The GitHub web link for the project.

7.2. Data Extraction

Data extraction is the output of the data extraction process, a very important aspect of data warehouse implementation. A data warehouse gathers data from several sources and utilizes these data to serve as vital information for the company. These data is used to spot patterns and trends both in the business operations as well as in industry standards. Since the data coming to the data warehouse may come from a different source which commonly is of disparate systems resulting in different data formats, a data warehouse uses three processes to make use of the data. These processes are extraction, transformation, and loading (ETL).

The script for Weather Dataset downloading

The historical data download for 40 years was the initial big step which was required to start our project. We decided to download it using the script because when we tried to download it online, we realized that the size of the data is massive, and the number of stations present in the city created more complexity. The python script is using the Weather Underground API [10]. The website gives you a unique key when you create an account, then using this key in your script you can download any data from their site and format it as you want.



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Feb 15 23:44:19 2018

@author: priyankaphapale
"""
# Changed the existing script with our requirements.
# Created by Kieng Iv to extract weather data from http://www.wunderground.com

# Importing the necessary modules
from bs4 import BeautifulSoup
import urllib, os, urllib.request

# creating CSV file to be used - each title is separated by a comma
file = open(os.path.expanduser(r"/Users/priyankaphapale/Documents/WeatherData_LasVegas.csv")
file.write(
        b"Date,Mean Temperature,Max Temperature,Min Temperature,Heating Degree Days, Dew Poi

# Looping based on year (vYear), month (vMonth) and date vDate)
# - if you want 2014 to 2015, your year range should be range(2014,2016)
# - if you want January to March, your month range should be range(1,4) - if you want everyt
# - if you want 1 to 12, your day range shoudl be range(1,13) - if you want everything than

for vYear in range(2018, 2019):
    for vMonth in range(1, 13):
        for vDay in range(1, 32):
            # go to the next month, if it is a leap year and greater than the 29th or if it
            # and greater than the 28th
            if vYear % 4 == 0:
                if vMonth == 2 and vDay > 29:
                    break
            else:
                if vMonth == 2 and vDay > 28:
                    break
            # go to the next month, if it is april, june, september or november and greater
            if vMonth in [4, 6, 9, 11] and vDay > 30:
                break

            # defining the date string to export and go to the next day using the url
            theDate = str(vYear) + "/" + str(vMonth) + "/" + str(vDay)

            # the new url created after each day
            theurl = "http://www.wunderground.com/history/airport/KVGT/" + theDate + "/Daily
            # extract the source data for analysis
            thepage = urllib.request.urlopen(theurl)
            soup = BeautifulSoup(thepage, "html.parser")
            MaxWindSpeed = Visibility = Events = AvgWindSpeed = SeaLevelPressure = Precipita
```

Usage

Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left

```
2018/1/10,52,63,40,13,43,74,100,48,0.01,29.97,N/A,20,9,

2018/1/11,51,61,41,14,40,68,93,42,0.00,30.29,N/A,9,10,

2018/1/12,52,62,41,13,40,66,89,43,0.00,30.32,N/A,9,10,

2018/1/13,55,65,44,10,34,47,66,28,0.00,30.37,N/A,15,10,

2018/1/14,55,66,43,10,33,47,66,28,0.00,30.30,N/A,15,10,

2018/1/15,55,65,44,10,37,49,66,31,0.00,30.19,N/A,9,10,

2018/1/16,55,63,46,10,37,55,77,32,0.00,30.24,N/A,8,10,

2018/1/17,55,66,43,10,35,52,76,28,0.00,30.29,N/A,12,10,

2018/1/18,55,64,45,10,37,53,71,34,0.00,30.16,N/A,9,10,

2018/1/19,60,71,48,5,35,43,63,23,0.00,29.88,N/A,24,10,

2018/1/20,51,58,43,14,26,40,54,26,0.00,29.92,N/A,30,10,

2018/1/21,46,54,38,19,15,33,48,18,0.00,30.32,N/A,21,10,

2018/1/22,44,53,34,21,22,45,64,26,0.00,30.36,N/A,7,10,

2018/1/23,49,59,39,16,25,40,54,26,0.00,30.34,N/A,13,10,
```

Figure 6. Screenshot of the data extraction script.

The only thing which needs to be changed is the year range, city code and filename you want your data to be saved in the script. Using the year range and city code the URL is generated to access the respective from the website, and it is saved in a CSV format with proper column names. The data entry is saved in your CSV file for each data. This data can be further used to do required analysis. The script which is shown above in Figure 6, is attached in the appendix for your reference.

7.3. Data Conversion

One of the essential steps for working with Big Data is a cleaning process. As we have noticed after we got all the datasets for the weather, a lot of data is missing, like several years with the occasional appearance of some numbers. Some files had some minor losses. In both cases we used different approaches to be sure that the data we will use to process in our algorithms is clean, meaning that the temperatures are present (min, max, mean).

- For the minor losses, we used a python script which is replacing all missing temperatures with the average value for each day which is calculated for all of the years available in the processed file. During the research process, we found out that it is reasonable to use Excel application if your data is less than the 1M record, but we used Python for two reasons: 1) we needed to document this process and 2) for the educational purposes.
- For the missing data for years, we have decided just to delete all of the rows, because during simple replacing it produces even more errors and the script doesn't work correctly and leaves missing cells.

7.4. Database Schema

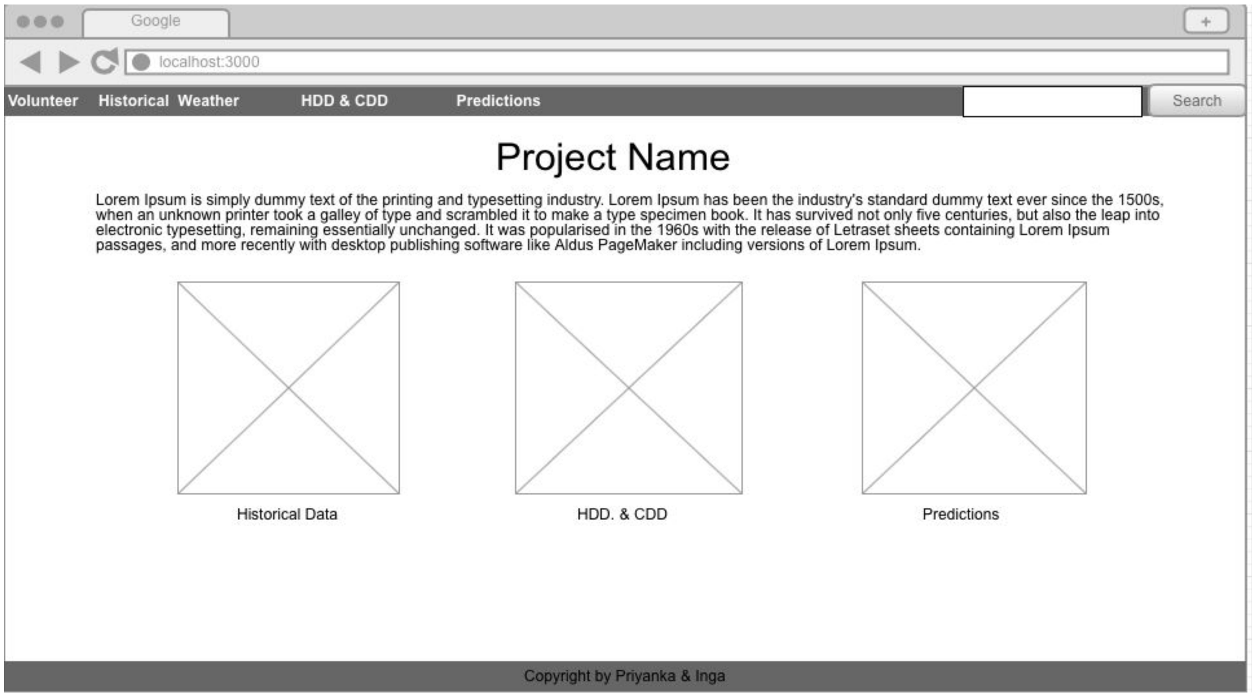
In general, MongoDB is a schema-less database, but below we are listed the collections (tables) we are using:

Collection Name	Description
HDD	Contains all of the information about HDD/CDD (business date, price, city code, description of the security).
HistoricalWeatherData	Contains all of the weather datasets for all eight cities (business date, city code and temperatures are most important attributes in it).
HistoricalWeatherData_avg	Contains all of the averages temperatures for each day of the year from all available years from a particular city.

Table 8. Database schema.

7.5. User Interface

Below, in Figure 7, is the initial design of the first page where we will have our project name and welcome description, and navigation to view historical data, HDD/CDD data, and price predictions.



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Figure 7. The prototype of main page.

Below, in Figure 8, is the initial design of the Historical weather display page where the user can select the date range and city and view the historical data of required year or month or week or days.

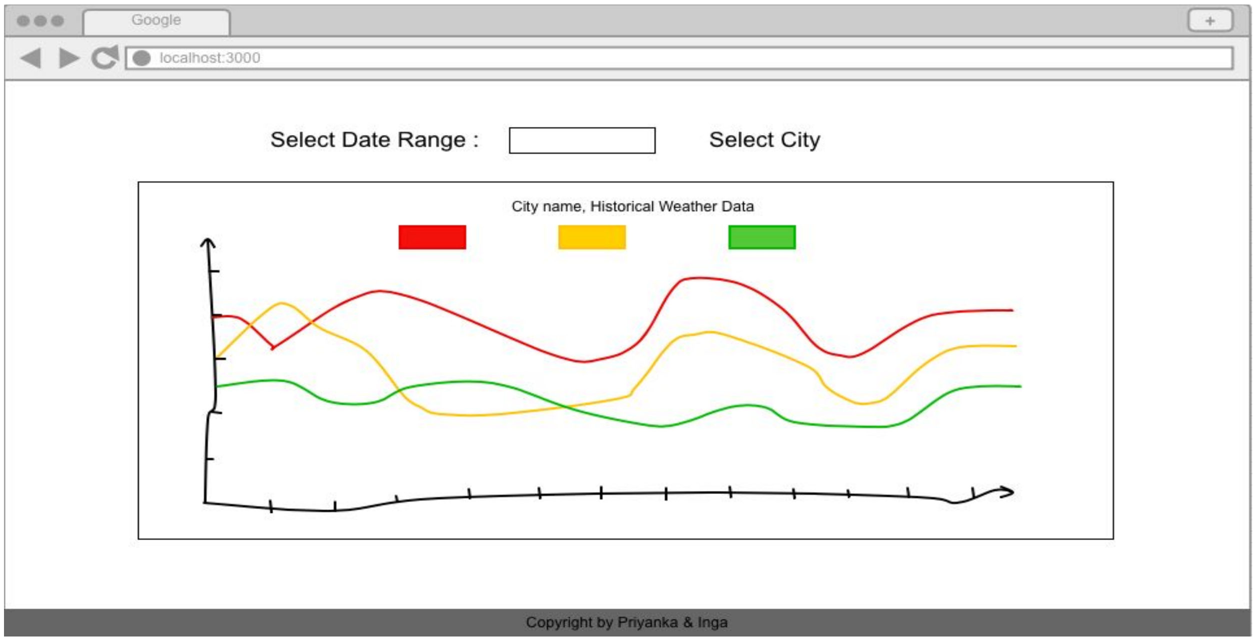
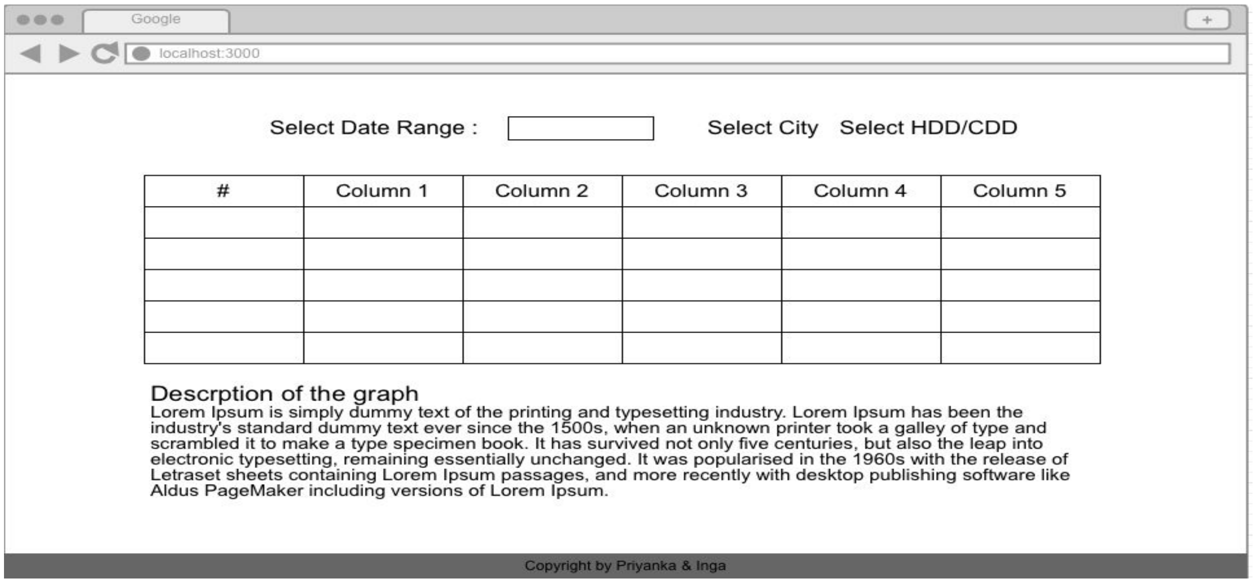

Google
localhost:3000
Select Date Range :
Select City
City name, Historical Weather Data
Copyright by Priyanka & Inga

Figure 8. The prototype of Historical Weather page.

Below, in Figure 9, is the initial design of the HDD/CDD page where the user can select the date range and city and view the details of HDD data. This data we will only be used to compare it with our predicted data to show the accuracy.



#	Column 1	Column 2	Column 3	Column 4	Column 5

Descrption of the graph
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Figure 9. The prototype of the page for displaying HDD/CDD table.

Below, in the Figure ... is the initial design of the prediction page where the user can select the month and city and view the predicted prices of the selected month. This data we will be useful for the user to analyze predicted prices.

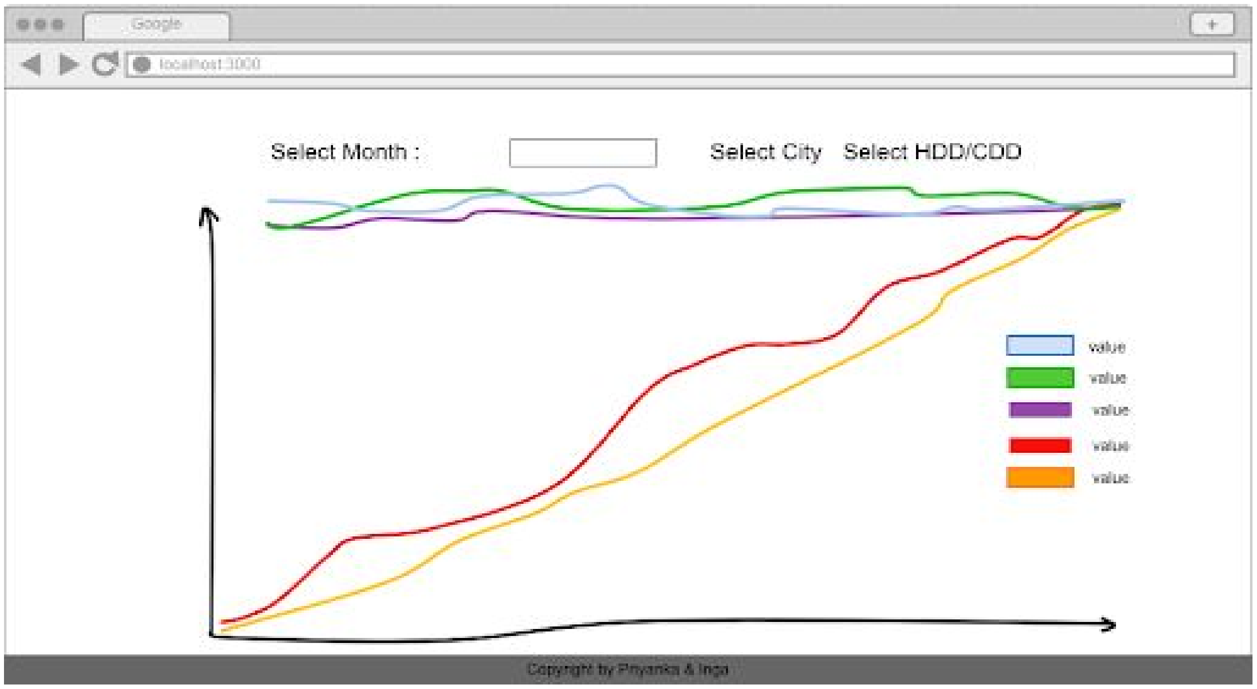


Figure 10. The prototype for Magic wand prediction page

7.6. Web Development

Set up the environment for the development

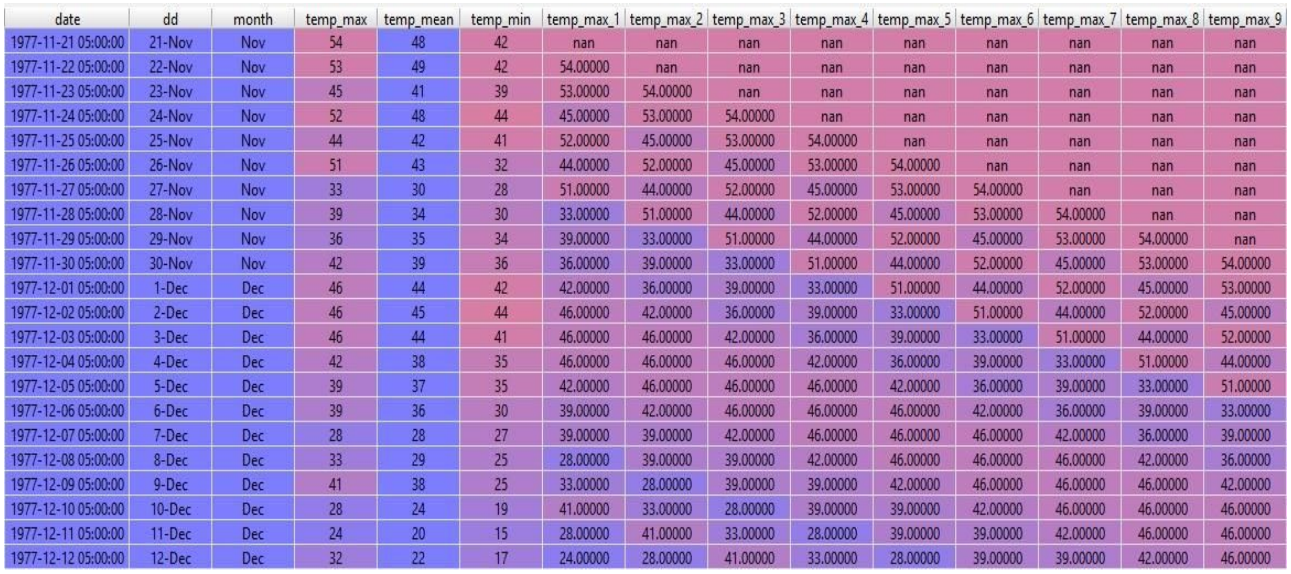
- For the Angular installation, we've used the instructions from official website [19]
- Install NodeJS and npm [20].
- Install the Angular CLI globally by using command npm install -g @angular/cli.
- We are using AMPPS Stack for MongoDB usage. Download it from here [http://www.ampps.com/downloads] PHP Developer Package. Install as mentioned in the instructions. Grant the permissions when you are launching AMPPS. To start MongoDB go to the /Applications/ampps/mongodb/bin directory and in Terminal use ./mongo command.
- Anaconda environment was installed for using Python.

7.7. Linear Regression Algorithm implementation and Price calculation

Listed below are the major steps which we were using for the weather and prices prediction script:

- Passing arguments to the script (startDate, endDate and city Code for prediction). The way how we are getting and passing these arguments described in section 4.9.

- The results of Linear Regression projections depend on predictors we have chosen. The better predictors we have, the more accuracy we will get. The predictors we have selected from available data are the previous day's temperature (min, max, mean) and the historical average for these days of the year.
- We need to fetch all of the temperatures data starting from the predicted day and ten days before that. So, the next query to the database is to chosen days of the year we are going to predict the weather. For example, if we are forecasting for Dec 1-31, then the query should return Nov 21-Dec 31 and collect the corresponding data (Figure 11).
- These data are divided into two sets: everything, before the current year is used as a train set for the algorithm and the data for the current year, is used as a test set.
- Once we populated our previous day's weather data, we can remove now unneeded the last month days from our data frame.
- The next step is to call precomputed early average temperatures from the collection HistoricalWeatherData_avg (described in section 4.4 above) and use this data as a one more predictors set. Then we need to merge these two sets of predictors: previous days weather this year and historical averages.
- Using SciKit-Learn LinearRegression Module [http://scikit-learn.org/stable/] analyze our train set and make a prediction on our test set. Call the function fit in LinearRegression library to train the algorithm and call predict method to get the result (below is a code sample:



date	dd	month	temp_max	temp_mean	temp_min	temp_max_1	temp_max_2	temp_max_3	temp_max_4	temp_max_5	temp_max_6	temp_max_7	temp_max_8	temp_max_9
1977-11-21 05:00:00	21-Nov	Nov	54	48	42	nan	nan	nan	nan	nan	nan	nan	nan	nan
1977-11-22 05:00:00	22-Nov	Nov	53	49	42	54.00000	nan	nan	nan	nan	nan	nan	nan	nan
1977-11-23 05:00:00	23-Nov	Nov	45	41	39	53.00000	54.00000	nan	nan	nan	nan	nan	nan	nan
1977-11-24 05:00:00	24-Nov	Nov	52	48	44	45.00000	53.00000	54.00000	nan	nan	nan	nan	nan	nan
1977-11-25 05:00:00	25-Nov	Nov	44	42	41	52.00000	45.00000	53.00000	54.00000	nan	nan	nan	nan	nan
1977-11-26 05:00:00	26-Nov	Nov	51	43	32	44.00000	52.00000	45.00000	53.00000	54.00000	nan	nan	nan	nan
1977-11-27 05:00:00	27-Nov	Nov	33	30	28	51.00000	44.00000	52.00000	45.00000	53.00000	54.00000	nan	nan	nan
1977-11-28 05:00:00	28-Nov	Nov	39	34	30	33.00000	51.00000	44.00000	52.00000	45.00000	53.00000	54.00000	nan	nan
1977-11-29 05:00:00	29-Nov	Nov	36	35	34	39.00000	33.00000	51.00000	44.00000	52.00000	45.00000	53.00000	54.00000	nan
1977-11-30 05:00:00	30-Nov	Nov	42	39	36	36.00000	39.00000	33.00000	51.00000	44.00000	52.00000	45.00000	53.00000	54.00000
1977-12-01 05:00:00	1-Dec	Dec	46	44	42	42.00000	36.00000	39.00000	33.00000	51.00000	44.00000	52.00000	45.00000	53.00000
1977-12-02 05:00:00	2-Dec	Dec	46	45	44	46.00000	42.00000	36.00000	39.00000	33.00000	51.00000	44.00000	52.00000	45.00000
1977-12-03 05:00:00	3-Dec	Dec	46	44	41	46.00000	46.00000	42.00000	36.00000	39.00000	33.00000	51.00000	44.00000	52.00000
1977-12-04 05:00:00	4-Dec	Dec	42	38	35	46.00000	46.00000	46.00000	42.00000	36.00000	39.00000	33.00000	51.00000	44.00000
1977-12-05 05:00:00	5-Dec	Dec	39	37	35	42.00000	46.00000	46.00000	46.00000	42.00000	36.00000	39.00000	33.00000	51.00000
1977-12-06 05:00:00	6-Dec	Dec	39	36	30	39.00000	42.00000	46.00000	46.00000	46.00000	42.00000	36.00000	39.00000	33.00000
1977-12-07 05:00:00	7-Dec	Dec	28	28	27	39.00000	39.00000	42.00000	46.00000	46.00000	46.00000	42.00000	36.00000	39.00000
1977-12-08 05:00:00	8-Dec	Dec	33	29	25	28.00000	39.00000	39.00000	42.00000	46.00000	46.00000	46.00000	42.00000	36.00000
1977-12-09 05:00:00	9-Dec	Dec	41	38	25	33.00000	28.00000	39.00000	39.00000	42.00000	46.00000	46.00000	46.00000	42.00000
1977-12-10 05:00:00	10-Dec	Dec	28	24	19	41.00000	33.00000	28.00000	39.00000	39.00000	42.00000	46.00000	46.00000	46.00000
1977-12-11 05:00:00	11-Dec	Dec	24	20	15	28.00000	41.00000	33.00000	28.00000	39.00000	39.00000	42.00000	46.00000	46.00000
1977-12-12 05:00:00	12-Dec	Dec	32	22	17	24.00000	28.00000	41.00000	33.00000	28.00000	39.00000	39.00000	42.00000	46.00000

Figure 11. Screenshot of the populating previous temperature for ten days

```
# instantiate the regressor class
regressor = LinearRegression()

# fit the build the model by fitting the regressor to the training
data
```

```
regressor.fit(X_train, y_train)

# make a prediction set using the test set
prediction = regressor.predict(X_test)
print(prediction)

# Evaluate the prediction accuracy of the model
from sklearn.metrics import mean_absolute_error,
median_absolute_error
```

- This library also provides the evaluation of the quality of our prediction:

```
print("The Explained Variance: %.2f" % regressor.score(X_test,
y_test))
print("The Mean Absolute Error: %.2f degrees celsius" %
mean_absolute_error(y_test.sort_index(), prediction))
print("The Median Absolute Error: %.2f degrees celsius" %
median_absolute_error(y_test.sort_index(), prediction))
```

- After the weather prediction is made, evaluating our predicted index value is straightforward. HDD is defined as 65F-Tmean, so we just add the calculated HDD numbers for the month, and this is our index price prediction.

7.9. Tunneling between User Interface and Python scripts

NodeJS side code (api.js file) takes all of the parametres, combines arguments with command line instruction and executes script in synchronous mode:

```
let stdout = execSync('python ../backend/prediction.py '+ req.params.startDate + ' '
+ req.params.endDate + ' ' +req.params.cityCode).toString();
```

The result from the Python script returns as a json file in standard output (stdout) which we can process now:

```
stdout = stdout.substring(stdout.indexOf("jsonresult:") + "jsonresult:".length);
res.setHeader('Content-Type', 'application/json');
res.send(stdout);
```

Python script takes all of the arguments and process them:

```
import sys
startDate = sys.argv[1]
endDate = sys.argv[2]
endDate = sys.argv[3]
```

After the script is executed, it generates JSON file and prints it in standard output, which will be read by NodeJS side.

```
response["data"]=requiredData
json_data = json.dumps(response)
print ("jsonresult:"+json_data)
```

8. Application Testing

8.1. Objective and Success Criteria

The following table summarizes the techniques planned for the testing of the website. The objective and success criteria of each test are summarized. Also, the application module(s) for the testing technique is also identified.

Testing Technique	**Objective**	**Success Criteria**	**Modules**
Unit Testing	To test each independent module. - All sub-components of a module will be tested. - Stub and drivers utilized to do independent testing	The module should work for functionalities like -Insert -Select -Update -Delete with appropriate stubs and drivers (if applicable)	- All data visualization related modules - Data extraction - Data conversion - Algorithms
Integration Testing	To test each module after it is integrated with the main module.	The modules continue to work for all functionalities after integrating with other modules and that it interacts with other modules after replacing stubs	- All data visualization related modules - Data extraction - Data conversion - Algorithms
Functional Testing Round 1	To test by mapping functionalities with user specifications.	Functionalities should match user specifications.	
Data Integrity Testing	Ensure database access methods functions properly. -Invoke each database access method with valid and invalid data. -Inspect the database to verify the inserted data. -Verify the data populated from the	All database related activities should work like: -data insertion -data query -data delete -data update (if applicable)	- All data visualization related modules - Data extraction - Data conversion - Algorithms

	database.		
User Interface (UI) Testing	To test appropriate access and navigation features.	UI features should be easily accessible. Navigation design should be comprehensible. UI should match user needs.	All validation should be performed for all fields
Functional Testing Round 2	To repeat the functional testing after bug fixes and improvements.	High priority bugs, showstoppers should befixed. The system should adhere to user specifications.	All of the modules

Table 9. Application testing. Objective and Success Criteria

8.2. Use cases with test data

Below are the test scripts for each module of the website with evaluation criteria.

Module: Chart drawing for weather displaying

Number	Test Objective	Completion Criteria
1	Select a date range and leave city field empty.	An empty graph should appear
2	Select date range for several days and up to one month, and select city.	The graph on the page displays with min, max and mean temperature for each selected day. X-axis should be labeled for each **day**.
3	Select date range for more than 30 days up to 3 months and select city.	The graph on the page displays with min, max and mean temperature for each selected day. X-axis should be labeled for each **week**.
4	Select date range for more than 90 days up to 6 months and select city.	The graph on the page displays with min, max and mean temperature for each selected day. X-axis should be labeled for each **month**.
5	Select date range for more than six months up to 12 months and select city.	The graph on the page displays with min, max and mean temperature for each selected day. X-axis should be labeled for each

		quarter.
6	Select the date range for more than 12 months up to several years and select city.	The graph on the page displays with min, max and mean temperature for each selected day. X-axis should be labeled for each **year**.

Table 10. Test Cases – Chart Drawing

Module: Weather derivatives displaying

Number	Test Objective	Completion Criteria
1	Select the date range and city	The data for the respective city is displayed in tabular format (i.e., Business Date, Price, Currency, Security Description, City Code, Index Code).

Table 11. Test Cases – Weather Derivatives

Module: Data extraction script

Number	Test Objective	Completion Criteria
1	Change the city_code, date range and the file name to be saved.	The script generates the required URL with the city code, date range and gets the day to day weather data for the given city.

Table 12. Test Cases – Data Extraction Script

Module: Data conversion script

Number	Test Objective	Completion Criteria
1	Change the file name with the data which needed to be cleaned.	The script generates a new file with cleaned data for temperatures, meaning that there are no missing temperatures for the min, max and mean value. The output should have 0 as some rows of the file which have temperature value.

Table 13. Test Cases – Data Conversion Script

Module: Linear Regression Algorithm

Number	Test Objective	Completion Criteria
1	Add 10 days before the first day of the month in Train and Test sets for the algorithm	The output data frame contains predictors based on averages for all of the years for these days plus temperatures from previous 10 days for this month.
2	Run the algorithm for the first day for prediction	The temperatures are predicted as expected. The train and test sets update accordingly by dropping unneeded columns and rows.
3	Run the algorithm for the rest of the days of chosen month	The temperatures are predicted as expected.
4	Run the cycle to calculate HDD and prices based on the predicted temperatures for each day	The json file with predicted temperatures, HDD values and prices is generated.

Table 14. Test Cases – Linear Regression Algorithm

Module: Price calculation algorithm

Number	Test Objective	Completion Criteria
1	Choose the month with 30 days	The script generates exactly 30 prices for chosen month
2	Choose the month with 31 days	The script generates exactly 31 prices for chosen month
3	Choose the month with 28 days	The script generates exactly 28 prices for chosen month

Table 15. Test Cases – Price calculation Algorithm

9. Application Usage

9.1. Running the web interface locally

Below are the steps which should be done if any user wants to run the application on his/her machine:

1. Follow the instructions given in the section 4.6 Web Development to set up the environment which user will use to run the application on the local machine.
2. Pull all of the code from GitHub link [29] to a new project.
3. Use the following instruction to restore database from dump file:

```
./mongorestore -d weatherData /d/dump/weatherData
```

where weatherData is a database name and "/d/dump/weatherData" is a path for data extraction for your convenience. In case, if you need to make a database dump, use the following instruction:

```
./mongodump -d weatherData  -o <directory_backup>.
```

You can choose your path for the directory.

4. Run terminal window on your frontend folder and use the following sequence of the instructions to build and run the project:

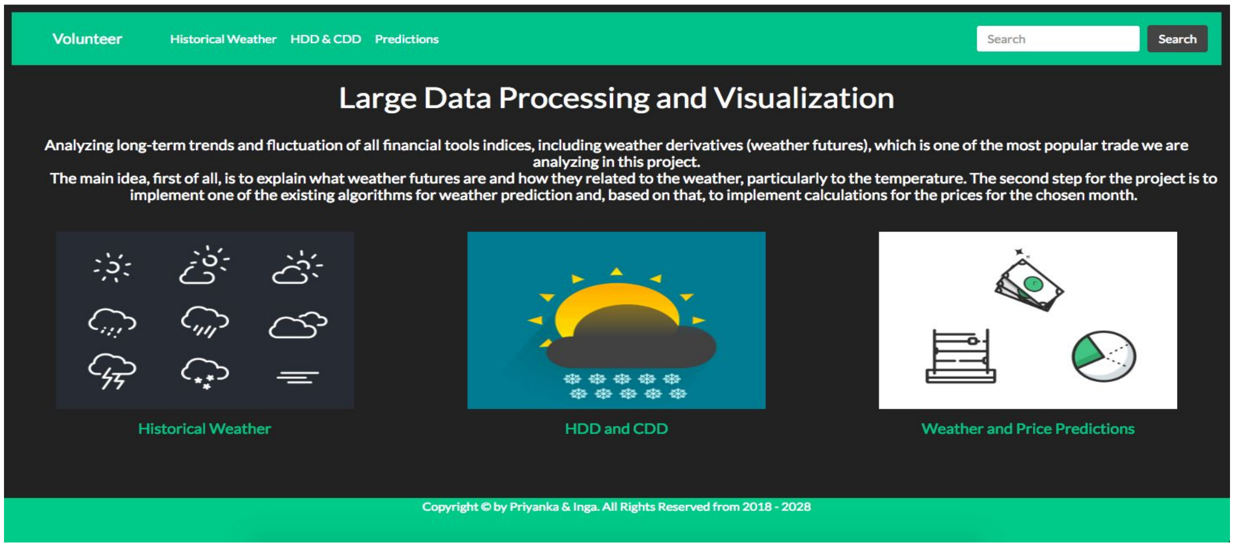
```
ng build
node server
```

5. Go to localhost:3000 in your browser and use the application.

9.2. Screenshots

Below are the screenshots of the description and user manual instructions.

Main Page



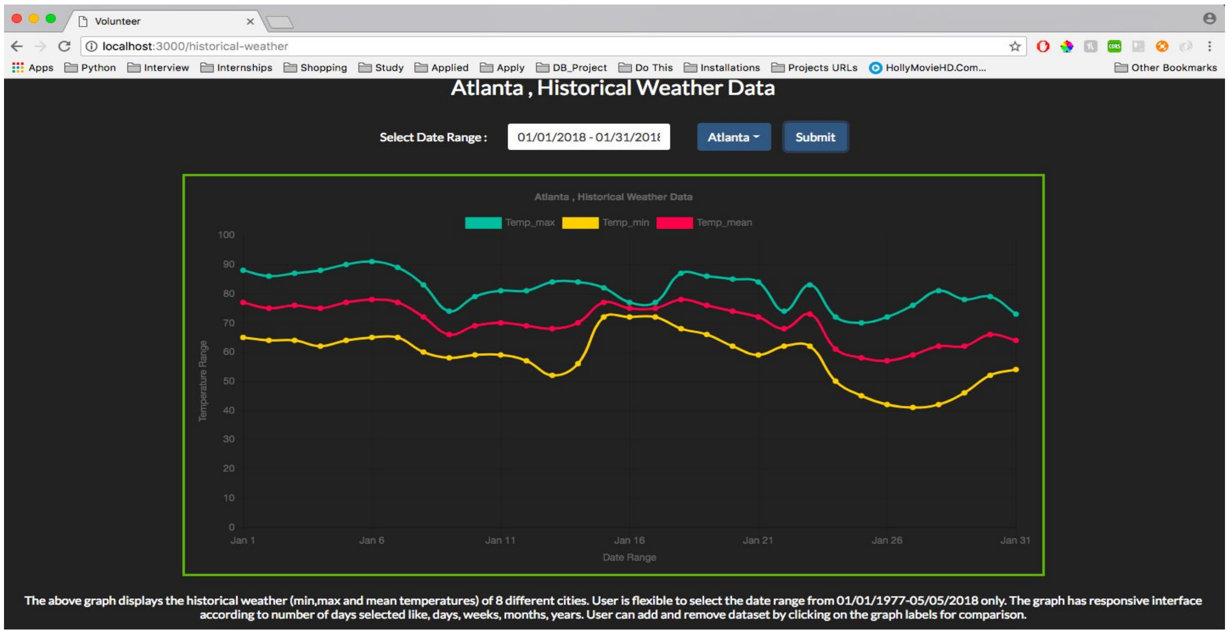
Large Data Processing and Visualization

Analyzing long-term trends and fluctuation of all financial tools indices, including weather derivatives (weather futures), which is one of the most popular trade we are analyzing in this project.
The main idea, first of all, is to explain what weather futures are and how they related to the weather, particularly to the temperature. The second step for the project is to implement one of the existing algorithms for weather prediction and, based on that, to implement calculations for the prices for the chosen month.

Copyright © by Priyanka & Inga. All Rights Reserved from 2018 - 2028

This is the final design of the first welcome page where we will have our project name and project description about what this website is doing, and navigation to view historical data, HDD/CDD data, and price predictions.

Historical Weather Page

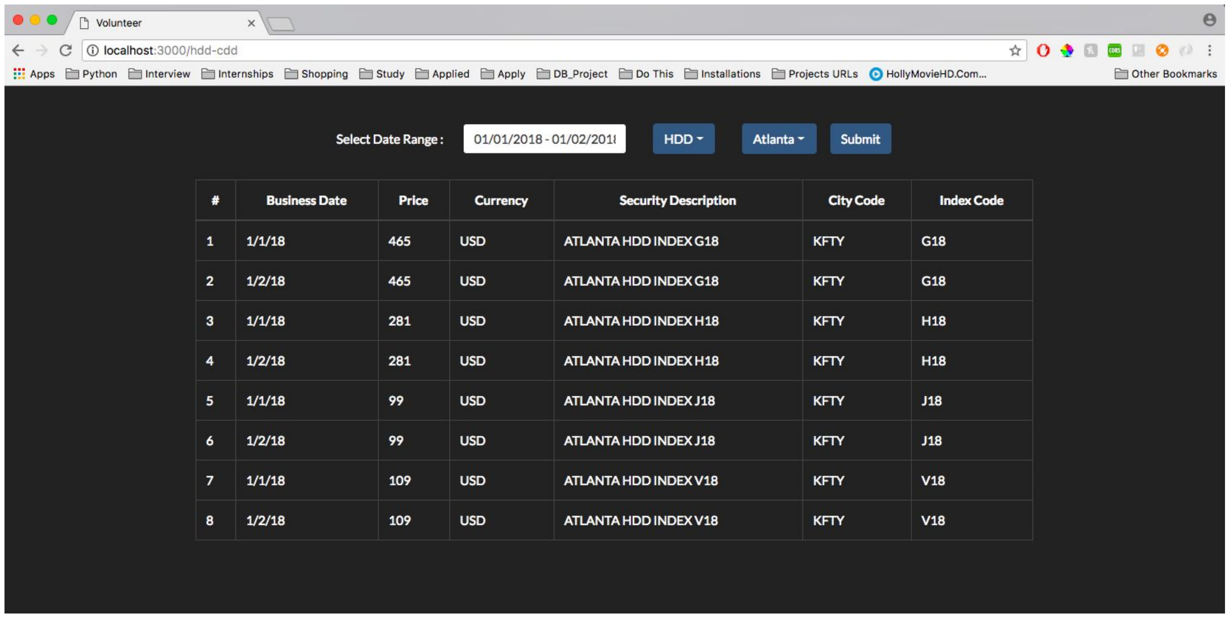


Atlanta , Historical Weather Data

The above graph displays the historical weather (min,max and mean temperatures) of 8 different cities. User is flexible to select the date range from 01/01/1977-05/05/2018 only. The graph has responsive interface according to number of days selected like, days, weeks, months, years. User can add and remove dataset by clicking on the graph labels for comparison.

This is the final design of the Historical weather display page where the user can select the date range and city and view the historical data of required year or month or week or days.

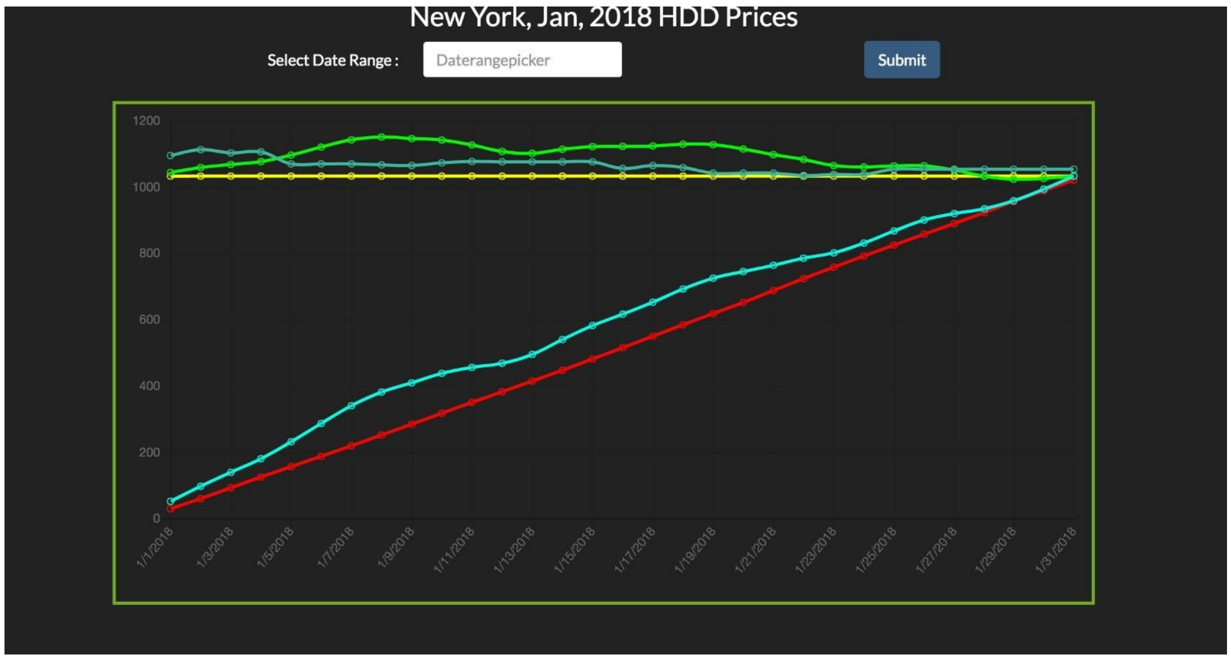
HDD and CDD Page



#	Business Date	Price	Currency	Security Description	City Code	Index Code
1	1/1/18	465	USD	ATLANTA HDD INDEX G18	KFTY	G18
2	1/2/18	465	USD	ATLANTA HDD INDEX G18	KFTY	G18
3	1/1/18	281	USD	ATLANTA HDD INDEX H18	KFTY	H18
4	1/2/18	281	USD	ATLANTA HDD INDEX H18	KFTY	H18
5	1/1/18	99	USD	ATLANTA HDD INDEX J18	KFTY	J18
6	1/2/18	99	USD	ATLANTA HDD INDEX J18	KFTY	J18
7	1/1/18	109	USD	ATLANTA HDD INDEX V18	KFTY	V18
8	1/2/18	109	USD	ATLANTA HDD INDEX V18	KFTY	V18

This is the final design of the HDD/CDD page where the user can select the date range and city and view the details of HDD data. This data we will only be used to compare it with our predicted data to show the accuracy.

Weather and Price Prediction Page



This is the final design of the prediction page where the user can select the month and city and view the predicted prices of the selected month. This data we will be useful for the user to analyze anticipated prices.

Table 16. Application Screenshots

10. Future work

10.1. The current state of the project

Below is a table which is describing what parts of the defined specifications are done and with what functionality and limitations.

Functionality at the final phase

Function	**Implemented**	**Description**
Main page	Done	It has our project name and project description of what this website is doing, and navigation to view historical data, HDD/CDD data, and price predictions.
Displaying weather (temperature) page with filter	Done	This is the Historical weather display page where the user can select the date range and city and view the historical data of required year or month or week or days.
Displaying weather derivatives page with filter	Done	This is the page where the user can select the date range and city and view the details of HDD data. This data we will only be used to compare it with our predicted data to show the accuracy.
Magic wand page (divided into sections)		
Filter to pick a month for prediction	Done	The filter allows to pick a desired month for the prediction temperatures and prices
Predicted prices graph vs. real historical prices vs. final price for the month vs. sum to date HDD values vs. sum to date historical HDD values	Done	The final graph shows all of the predicted numbers in the defined form i.e. (Prices for each day, final price of the month, sum to date HDD values, sum to date historical HDD values)

Table 17. Functionality at the final phase

10.2. Short-term perspectives

Our application can be improved in several directions, separately or all together as listed below:

- The user interface can be improved by adding a tutorial on how to use the application and glossary.
- A mobile version of the website can be created.
- Several algorithms can be implemented and presented as options for a user for a comparison and a decision making.
- More parameters from weather dataset could be included for training existing algorithms

10.3. Long-term perspectives

Based on the market research, competitive solutions and articles review our application can be converted from a free education to a monetized tool which can be customized for clients depended on their needs and expanded for more financial tools based on weather prediction, like options based on weather derivatives, commodities, etc.

Promotion plan

Online promotion

- Distribution of the application in the AppStore for free in basic configuration
- Distribution of the application on Google Play in basic configuration
- Supporting website
- Supporting social media page on Facebook
- PPC AdWords advertisement campaign

Offline promotion

Creating a marketing plan for presenting the application for 12 months to attend all events in the area related to the development of software in the financial industry.

11. Collaboration

History

The history of our collaborations did not appear as a sudden decision. It was based on the previous experience of working together. During last two semesters, we completed eight projects together in many areas for different classes. The master project is our ninth joint work.

Idea

The idea of this project was approved during the discussion about the area to explore for Big Data learning with a combination of personal preferences. Priyanka suggested that there are many interesting topics can be studied by using weather-related datasets, because the weather is an extensive and accessible area, datasets are available from many sources. Inga suggested finding a topic which could be related to the financial industry, because being familiar with this area making you more competitive in the job market in one of the most payable sectors.

Strategy and Technologies

Based on our past working experience and preferences for the future specializations, we divided our responsibilities in the following directions: Inga has created a plan for the projects, managing the whole process, deadlines, and requirements. Priyanka was responsible for choosing the technologies which fulfill all of the needs of the project, as well as learning and deciding time consumption for the implementation. Based on that, we both make changes in our strategy and schedule our next steps accordingly. We both agree that flexibility and trust to each other in decisions are the most essential keys which help us to complete our projects successfully.

Implementation

We divided our responsibilities for the implementation part easily based on the personal preferences. Priyanka prefers to learn more and implement all modules related to the User Interface. Inga is responsible for the backend part.

Documentation

Based on the discussed strategy for the topic and technologies, Inga creates the outline for the final document, which we both are filling out depending on who completed which section, research, module or testing. Priyanka is responsible for proofreading and formatting.

12. Conclusion

At the end of this project, we achieved all required goals which we defined at the beginning. We learned about an entirely new area, such as Big Data, and which techniques can be applied to download and process a significant amount of data, and how to make it usable for our needs. We learned how Python and corresponding libraries can be used for data processing and found out that there are many useful libraries to work with big files and built-in algorithms which simplifies the implementation process significantly. We also started to learn an entirely new stack called MEAN Stack technology for Web Development. As a result, we delivered a fully functional website which visualizes all of our algorithm calculations, filters, graphs and has a user-friendly interface.

As for the topic of this project, we researched two fields: weather prediction area and financial contract, particularly weather futures. We learned that the weather prediction has a significant impact on many industries, not just everyday life. In our case, not only weather derivatives (weather futures) directly depends on the temperature prediction, but other types of financial contracts, such as commodities, for example. Prices forecast for the financial industry has significant value considering that accuracy of the prices prediction corresponds to the gains and losses.

References

[1]. Financial instrument [Online], 2017. Available: Wikipedia, https://en.wikipedia.org/wiki/Financial_instrument [Accessed: May 5, 2018]

[2].Understanding Derivatives: Markets and Infrastructure [Online], 2018. Avalable: Federal Reserve Bank of Chicago https://www.chicagofed.org/publications/understanding-derivatives/index [Accessed: May 5, 2018]

[3]. Futures contract [Online], 2018. Available: Wikipedia, https://en.wikipedia.org/wiki/Futures_contract [Accessed: May 5, 2018]

[4]. Weather Future [Online], 2018. Available: Investopedia, https://www.investopedia.com/terms/w/weatherfuture.asp [Accessed: May 5, 2018]

[5]. MDA Information Systems LLC [Online], 2018. Available: MDA Information Systems http://www.mdaus.com/ [Accessed: May 5, 2018]

[6]. CME Group, “CME Rulebook. Chapter 403
CME Degree Days Index Futures”, *Chicago Mercantile Exchange Inc.* [Online], 2018. Available: https://www.cmegroup.com/rulebook/CME/IV/400/403/403.pdf [Accessed: May 5, 2018]

[7]. Google AdWords, Advertising Tool [Online], 2018. Available: https://adwords.google.com/home/how-it-works/#?modal_active=none [Accessed: May 5, 2018]

[8]. Holmstrom, M., Liu, D. and Vo, C. (2016). Machine Learning Applied to Weather Forecasting. [online] Cs229.stanford.edu. Available at: http://cs229.stanford.edu/proj2016/report/HolmstromLiuVo-MachineLearningAppliedToWeather Forecasting-report.pdf [Accessed 6 May 2018].

[9]. CME Group, “Monthly Heating Degree Days (HDD) Futures Daily
Settlement Procedure”, *Chicago Mercantile Exchange Inc.* [Online], 2018. Available: https://www.cmegroup.com/trading/weather/files/Monthly-HDD-Futures-Daily-Settlement-Procedures.pdf [Accessed: May 6, 2018].

[10]. Weather Underground, [Online], 2018. Available: https://www.wunderground.com [Accessed: May 6, 2018].

[11]. Bloomberg, [Online], 2018. Available: https://www.bloomberg.com [Accessed: May 6, 2018].

[12]. Yale University, “Bloomberg for Education. Getting started guide.”, [Online], 2018. Available: https://guides.library.yale.edu/ld.php?content_id=11942142 [Accessed: May 6, 2018].

[13]. University of Delaware, “Bloomberg Common Functions”, [Online], 2018. Available: https://lerner.udel.edu/sites/default/files/pdfs/Bloomberg_Common_Functions.pdf [Accessed: May 6, 2018].

[14]. Python (programming language) [Online], 2018. Available: https://en.wikipedia.org/wiki/Python_(programming_language) [Accessed: May 6, 2018].
[15]. David, T. (2018). Which Programming Language Is Better: R, Scala, or Python? - DZone Big Data. [online] dzone.com. Available at: https://dzone.com/articles/which-programming-language-is-better-r-scala-or-py [Accessed 6 May 2018].
[16]. pandas (software) [Online], 2018. Available: Wikipedia, https://en.wikipedia.org/wiki/Pandas_(software) [Accessed 6 May 2018].
[17]. MongoDB (software) [Online], 2018. Available: https://www.mongodb.com/ [Accessed 6 May 2018].
[18]. Express. Fast, unopinionated, minimalist web framework for Node.js (software) [Online], 2018. Available: https://expressjs.com/ [Accessed 6 May 2018].
[19]. Angular 5 (software) [Online], 2018. Available: https://angular.io/ [Accessed 6 May 2018].
[20]. NodeJS (software) [Online], 2018. Available: https://nodejs.org/en/ [Accessed 6 May 2018].
[21]. ChartJS (software) [Online], 2018. Available: http://www.chartjs.org/ [Accessed 6 May 2018].
[22]. Speedwell Weather (software) [Online], 2018. Available: http://www.speedwellweather.com/ [Accessed 6 May 2018].
[23]. AMPPS Stack (software) [Online], 2018. Available: https://www.ampps.com [Accessed 6 May 2018].
[24]. IntelliJ IDEA (software) [Online], 2018. Available: https://www.jetbrains.com/idea/ [Accessed 6 May 2018].
[25]. Anaconda The Most Popular Python Data Science Platform (software) [Online], 2018. Available: https://www.anaconda.com/ [Accessed 6 May 2018].
[26]. GitHub (software) [Online], 2018. Available: https://github.com [Accessed 6 May 2018].
[27]. Git [Online], 2018. Available: Wikipedia, https://en.wikipedia.org/wiki/Git [Accessed 6 May 2018].
[28].Adding an existing project to GitHub using the command line [Online], 2018. Available: https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/ [Accessed 6 May 2018].
[29]. Visualizing weather financial impact on industries and weather derivatives. Master Project. Technologies in use: MEAN, Python, Big Data [Online], 2018. Available: https://github.com/ingamontclair/Volunteer [Accessed 6 May 2018].