Oregon State University

CS 461

Fall 2017

Problem Statement

Author:
Shengpei Yuan

Instructor:
D. Kevin McGrath
Kirsten Winters

Abstract

The seasonal climate change is one of the essential factors for the harvest and returns of crops and farming investment programs of enterprises and organizations. As a sub-project of AgBizLogic, AgBizClimate is dedicated to deliver essential information about climate change to farmers, and help professionals to develop management pathways that best fit their operations under a changing climate. This project aims to link the crucial seasonal climate data from a professional climate toolbox database to AgBizLogic so that it could make specific analysis and demonstrating through powerful graphics. And this tool is planed to enable farmers and agriculture enterprises to select appropriate farming investment projects for their crops and products. It helps them to optimize budget and clearly understand the costs and returns of long term farming programs.

Currently there are long-term climate tools for crops but no convenient tool to extract short-term climate data from the Northwest climate toolbox database to AgBizClimate so that farmers could select appropriate seasonal plans for their crops. Moreover, a landing tool will be developed to allow users to switch between short term seasonal tool and long-term climate data tool.

The project generally finishes three tasks. Firstly, it automatically retrieves climate data of certain formats periodically by programs with specific configurations. Secondly, the collected climate data are efficiently managed by database tools and could be clearly displayed on a map. Thirdly, the desired seasonal climate data could be dynamically viewed for user-specified crops and locations by easily GUI interfaces. Powerful program language and frameworks like Python, Django and AngularJS are used to fulfill these tasks. And also we have specific metrics for measurement of completion of the project like outcome documents and standardization of operations.

Definition and description of the problem

The seasonal weather data are important for farmers and land managers as they may have great impacts on harvest crops. For instance, the precipitation on different days across the life cycles of crops may have different impacts on the harvest. Land managers and farmers used to rely on experiences on climate data from the past to make decisions for specific farming operations in order to reduce the negative impacts or make use of favorable climate conditions. However, these individual experiences of weather data are often limited and inaccurate. Professional tools like software systems could be adopted to build models for auto-decision making or based on available seasonal climate data.

AgBizClimate is a sub-project of AgBiz Logic (https://www.agbizlogic.com) that works on helping farmers and ranchs improve profits of by providing constructive advices for decisions on investment and programs. AgBizClimate is designed to deliver essential information about climate change to farmers and land managers for specific farming operations. However, we have to collect sufficient climate data before making any analysis. Therefore, the aim of this project is to link seasonal weather data from somewhere to AgBizClimate. Specifically, we plan to transfer and integrate weather data from Northwest Climate Toolbox database (https://climatetoolbox.org) into AgBizClimate (https://www.agbizlogic.com) for analysis and presentations of influences of climate changing on costs and returns of farming programs.

Generally, the tool is designed to work as follows. Firstly, there are various crops that are supported to be examined about relevant seasonal climate data and users could choose the one they would like to. Secondly, users could pick a location on the map by a pin for the objective climate data and crops. Thirdly, the climate data of the specified location are clearly displayed. Fourthly, users would be able to adjust the crops yields, inputs of plans and prices by easily GUI operations and see the expected returns immediately.

Although it sounds simple, there are several key issues one needs to overcome before successfully demonstrating data by figures on AgBizClimate web pages. Firstly, the weather data could vary a lot in terms of both formats and dimensions on the original Northwest Climate Toolbox database, and one has to choose appropriate formats and necessary dimensions according to the application contexts. Secondly, the dynamically collected data should be appropriately stored and managed so that they could be flexibly employed in AgBizClimate. Thirdly, one need to choose suitable styles and relevant map tools capable of extracting and demonstrating seasonal climate data in order to help users understand the underlying impacts of them on harvest and returns of farming programs according to climate change.

Proposed solution

As we have decided the source of the seasonal climate data is Northwest Climate Toolbox database (https://climatetoolbox.org) that provides professional climate data of Northwest area with various formats, we need a powerful tool to retrieve them from remote database server and save them in the local AgBizClimate server. And then we could parse the local climate data, and extract interested parts or make transformations according to the practical application scenarios. Finally, the desired climate data are provided to the AgBizClimate applications for further analysis or demonstrating purposes.

The main tool we choose is Python programming language, which is lightweight and efficient. It is good at network programming, parsing and transforming various formats of data like XML and

JSON.

For the three main issues mentioned in the previous part, we plan to solve them with following solutions. First, the XML or JSON formats of raw data from the climate toolbox are favored as we could easily handle them by Python, and specific fields names are predefined in configuration files for reducing dimensions of climate data. Secondly, database tools like sqlite and MongoDB would be adapted to efficiently store and mange the retrieved climate data. And it is easy for Python to operate on these databases. Thirdly, for the various tools capable of extracting and demonstrating climate data by colorful graphic depictions with map on web pages, we are in favor of open source frameworks like python Django and AngularJS.

Performance metrics

It is helpful to give some general metrics on the completeness of the project so that the clients could have a basic conception of the products of the project. Overall, we will get actionable climate data that can be integrated into the AgBizClimate for analysis and presentations. More specifically, we have following basic evaluation criteria:

First, interested climate data for crops could be retrieved through network (Internet) by programs (scripts) automatically without manual intervention under normal conditions, and then it could be displayed on a map and easily navigated. That is, once configured, the programs (scripts) should be able to run and download desired climate data from remote climatetoolbox server automatically.

Second, it is known that the climate data is dynamically increased as time goes on. Therefore, climate data should be able to be updated periodically or triggered manually by clicking a button or typing a command on the local programs. And there should be configuration files that define the parameters like time interval, etc.

Thirdly, it is possible that users want to examine the original climate data downloaded from remote climatedtoolbox server. Thus the raw climate data should be able to be exported easily in flat file formats like csv or txt. Moreover, when exporting data, it is required that multiple filtering or query conditions like dates and specific fields (dimensions) could be designated for the climate data as the entire data could be too large.

Last but not least, these operations should be friendly to both professional and non-professional users and always give users warnings or tips if they are making unsafe operations like deleting data files.