Café Data Product

Chris Arnold, Cory Gray, and Ryan Zembrodt

University of Kentucky
CS 499 – Senior Design Project
May 2016

Abstract

The steady growth of digital storage capacities and the connection of an increasing variety of devices to the internet allows for the collection of data sets so large and complex that traditional methods of data processing are rendered obsolete. While such data sets generally prove challenging to analyze, their sheer scope and comprehensiveness provide many opportunities to identify subtle trends in business, crime, and information. For this reason, many modern scientists and analysts employ machine learning and data mining techniques, allowing for the automation of big data analysis and rendering the process of identifying trends and relationships in large data sets much more efficient.

We present a web application which makes predictions about customer activity at a local cafe, based on factors such as time of day, external weather conditions, and advertising decisions. Predictions are made using models generated through machine learning algorithms, including regression and decision trees, applied to multiple large data sets provided by Dr. Julie Whitney of Lexmark International, Inc. and collected from a Lexmark campus café. The predictions made by the application achieve an average of at least 80% accuracy (using the Mean Absolute Scaled Error).

Introduction

Objective

The purpose of this project is to provide a web-based application that allows the user, an owner or manager of a café or restaurant, to estimate staffing and supply needs based upon predicted customer activity, based upon a user-modifiable time scale and input parameters. This allows for more efficient utilization of physical resources and personnel, potentially reducing overhead costs and increasing net profits.

Background

In February 2016, Dr. Julie Whitney, senior technical staff member at Lexmark International, Inc., presented our team with a large amount of data (described below) collected from a café serving Lexmark employees and visitors to the Lexmark campus. We were then tasked with using machine learning techniques to analyze the data sets and obtain models for predicting future customer activity. After initial exploration, we divided the data into two sets, one for training our models, and one for testing them. We then applied regression and decision tree algorithms to the training set to obtain predictive models, which we then refined until the models' predictions achieved 80% accuracy when compared to actual results from the testing set of the collected data. At this point, the models were implemented into a web application which allows the user to choose between models and adjust input parameters based upon their needs.

The Data Set

Included in the data set provided by Dr. Whitney are the following essential data points:

- Date and time of purchase (a string in format MM/DD/YYYY hh:mm)
- Item(s) purchased (given by integer item IDs)
- Perceived customer age group: unknown, child, young adult, adult, or senior (represented by integers from 0 to 4, respectively)
- Perceived customer sex: unknown, male, or female (represented by integers from 0 to 2, respectively)
- Time spent in the vicinity of an advertising screen (an integer in milliseconds)
- Time spent looking at the advertising screen (an integer in milliseconds)
- Item being advertised at time of purchase (an integer item ID)
- External temperature (an integer value representing degrees Fahrenheit)
- External humidity (an integer between 0 and 100 representing relative humidity percentage)
- External precipitation state (one of the following strings: "Clear", "Clouds", "Mist", "Rain", "Snow")

(Note: Some of the supplied data points have been fabricated to protect customer identities.) Our team will examine the relationships between these data point in order to find useful trends.

Specifications

The client, Dr. Whitney, outlined the following specifications for the product:

- The product will offer an interactive dashboard which depicts and predicts customer activity at a given café over a user-modifiable time scale and customer demographic range.
- The dashboard will feature a sign-in page for security purposes. Users will be able to create an account with a minimum character requirement for both usernames and passwords.
- The display of the dashboard must be non-technical and user-friendly, able to be understood by a layperson. The product will therefore be tested by non-technical users, who may then report on the understandability and ease-of-use of the product.
- During development, each team member will generate a prediction model using machine learning algorithms of his/her choice, focusing on a unique aspect of customer activity. Only the most accurate model will need to be included in the final product.
- Each model must achieve at least 80% accuracy in its predictions.
- The software must be designed with modularity in mind, and will allow for new or improved prediction models to be included in the future.

Product Planning

Effort and Size Estimations

The project is split into three major sections, each size estimated individually below:

- *Data Parser*: Used to parse the provided data set, preprocess the data, and implement machine learning algorithms to analyze the data - (2000 lines)
- *3x Prediction Models*: Some prediction models can be quite simplistic, such as a single formula taking a small number of variables to obtain a certain output. Others are much more complex, requiring a medium to large algorithm to generate predictions from input data - (500 lines each)
- *In-Browser Dashboard*: This constitutes the user interface of our product, and determines how the user navigates different screens in the application. Here we detail how interfaces behave, and how the display should look - (750 lines)

In actuality, the sizes of the three major sections varied from initial estimates. The actual size of each section is listed individually below:

- Data Parser: Used to parse and preprocess the data sets into usable subsets for the machine learning algorithms. – (2 versions: Java parser 2000 lines, C++ parser : 750 lines)
- 3x Individual Prediction Models: Apply machine learning algorithms from libraries to generate models of data for predicting. – (Average of 40 lines each) Create scripts to implement model without requiring the datasets or libraries, some simpler, others highly complex. – (Wide variety, simple: 40 lines, complex: 500 lines)
- In-Browser Dashboard: The user interface allowing practical usage of the prediction models. – (400 lines)

The final sizes of each section vary from the estimates, sometimes by very high percentages. The major source of this overestimating comes from misconceptions about the major sections. Initially, we believed the prediction models would be the most difficult and data parsing would be simple. However, we discovered that the data parsing is the most important and time consuming of the sections. Meanwhile the creating of models was extremely simply by using pre-existing libraries. Creating scripts to apply these models independently of their creation varied widely depending on the factors used in the model.

Schedule and Milestones

February 12:	Project Webpage Created
March 2:	Midterm Meeting with Dr. Piwowarski
March 7:	Project Design Completed
March 7:	Data Parsers Completed
March 7:	Final Trends Selected
March 9:	Midterm Presentation
March 21:	First Guess Models Completed
April 6:	Dashboard Interface Completed
April 4:	Testing Schedule Review with Dr. Piwowarski
April 11:	Code Review with Dr. Piwowarski
April 24:	Final Models Added to Dashboard
April 27:	Final Presentation
April 27:	Code Delievery

Platforms, Languages and Tools

We decided that it would be best to implement out final dashboard as an in-browser application. This simplified design and implementation by allowing us to separate our application from individual machines. This did require us to attempt to create our dashboard in a way that would operate on multiple browsers. We chose to implement out dashboard using RShiny, a web application creator designed for R with simple and widely applicable uses. By keeping the dashboard simple, we allowed the dashboard to be very flexible to work in many browsers. Although we each chose a different language for parsing our data (C++, Java and R), this has little impact on the final dashboard as it is internal to creating the models. We all chose to use the language R for creation of our machine learning models. This was suggested by our customer and she provided us with materials about machine learning techniques using R and its libraries. The libraries allowed us to generate the machine learned models with extreme ease.
We chose to use the machine learning algorithms provided by the R libraries as tools for creating our models. Again, this was suggested by our customer as it is unnecessary to “reinvent the wheel.”

Design

The client indicated that the project was largely designed to serve as a course in machine learning for the development team, as opposed to a request for a professionally developed product. For this reason, we as a development team were given a large amount of flexibility in the way we approached the client's specifications. As a result, we chose to focus largely on establishing accurate prediction models for the given data, providing us with a greater understanding of the underlying techniques, while assigning secondary importance to the production of a high-quality application interface.

During the machine learning stage of development, we used different techniques such as naïve Bayes or regression trees depending on what type of output our models were trying to predict. An algorithm like naïve Bayes is used for predicting factors such as true or false, while regression trees such as M5Prime is used for predicting continuous numerical data. Our use of R allowed this to easily and quickly be done due to its library support for these types of machine learning algorithms. Although there are different machine learning algorithms each of us used, the basic concept of training a model is the same. We took a large amount of data in a data table that contains both the dependent variable we want to predict and the independent variables that will be used for the prediction. In order to find the optimal independent variables needed for the prediction, we randomized the data and split it in half. One half of the data is the training set while the other half is split again, one set being validation data and the other being testing data. The purpose of this was to train several different models on the same training set with different combinations of independent variables and calculating their accuracy on the validation data set to choose the best model. The winning model would then be tested again on the test set of data to confirm its accuracy. Doing this ensures that any accuracy numbers calculated in the validation phase can be confirmed or denied in the testing phase. We did this hundreds of times with the dataset randomized each time for the most accurate data sets in order to confirm the best combination of independent variables. Using the optimal combination we could then create our predicting models and create linear functions in R to then represent those models. Figure 1 below shows an example of one of these models. It is the customer predictor created using the M5Prime algorithm, which creates a decision tree where each node is a linear regression model.

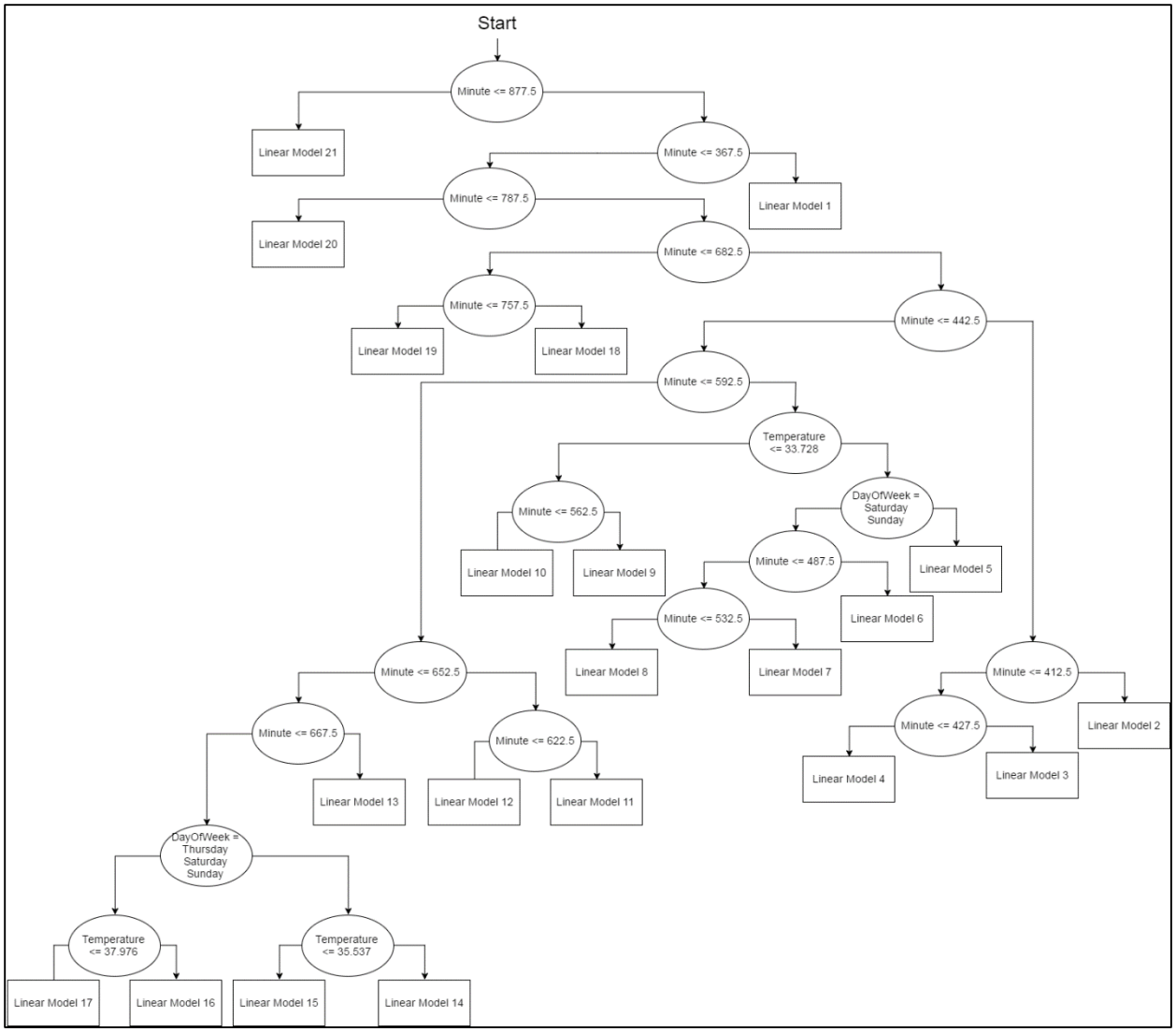

Figure 1: *Customer predictor decision tree*

Because the final product is to be utilized by business owners/operators, the design must be readily usable on almost any system that the business may use, eliminating the need for a business to utilize a specific system in order to make use of the product. In order to address this, the application will be run via a web browser, making it available to nearly every modern system in the world. For this we decided to use Shiny, a library in R that allows for the creation of dashboards the user can interact with in a web browser. Shiny allows us to feed the user's input into our previously created predictive algorithms and display the result to the user. The data is never uploaded to the server and the algorithms we created from our models is the only thing stored on the server that our dashboard accesses for its predictions. For additional security, the application requires the user to sign in, preventing other users from accessing his/her data.

The client also emphasized that the web application should be easy to understand, eliminating the need for the end-user to have any technical expertise, and easy to expand, allowing new prediction models to be implemented into the application dashboard. To this end, we focused on using large, easy-to-read print, simple language, and minimalistic design in order to provide an

easily understood experience for the user. Additionally, we implemented the prediction models as individual modules in the dashboard, allowing models to be added, removed, or edited without affecting the rest of the application. Figure 1, Figure 2, and Figure 3, show different user screens within the web application.

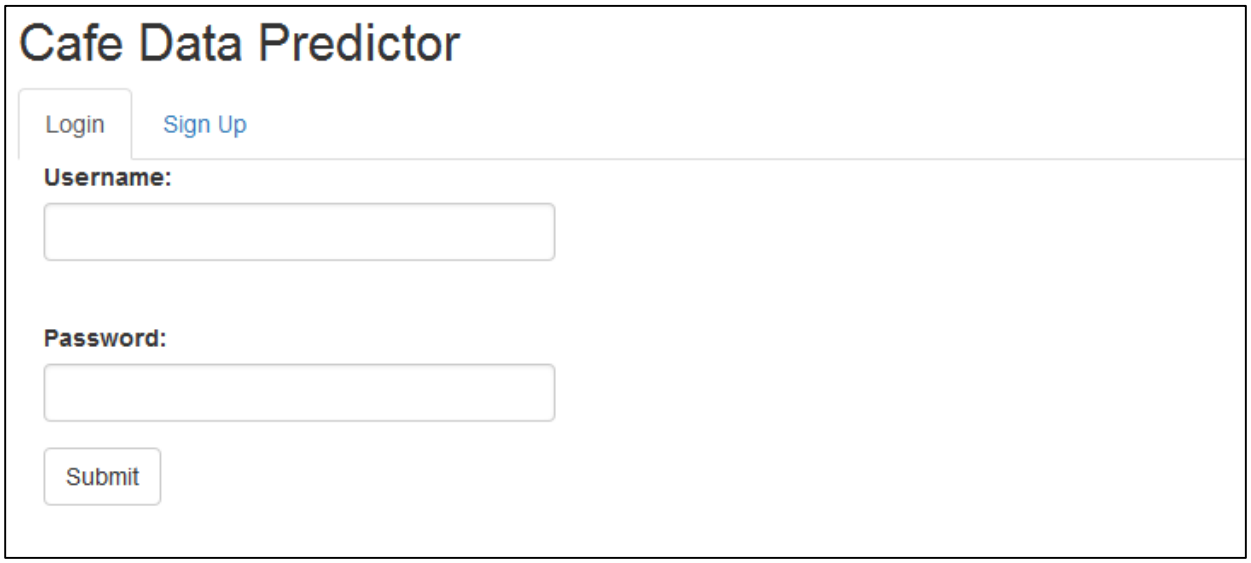

Figure 2: *Login screen*

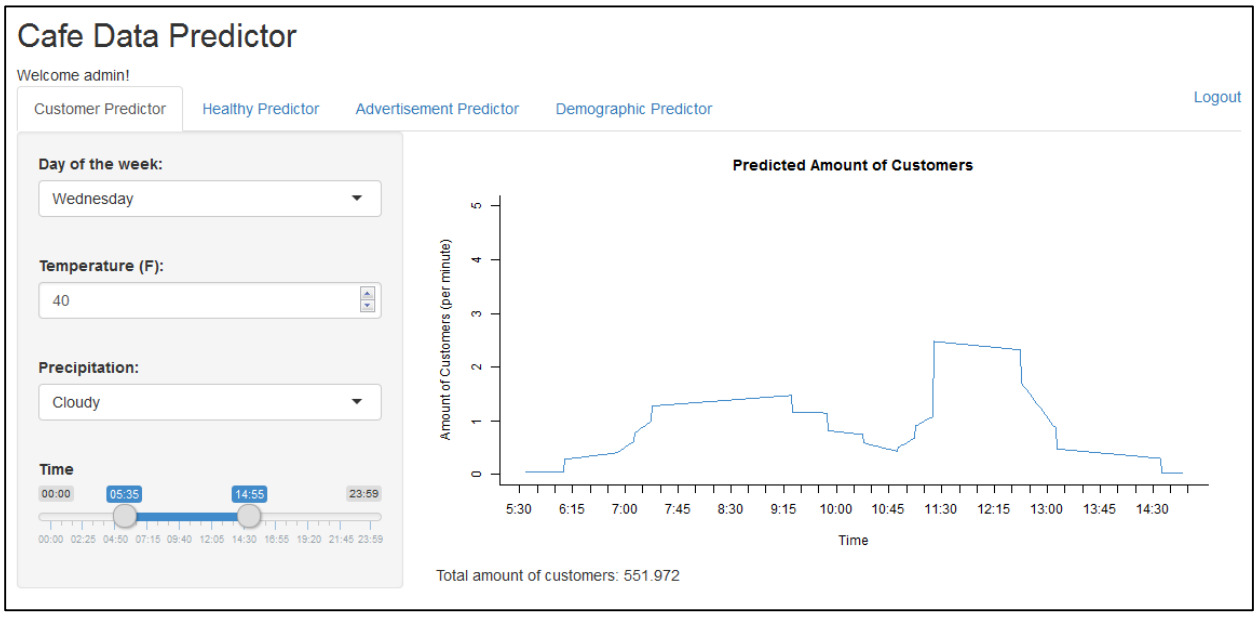

Figure 3: *Customer predictor*

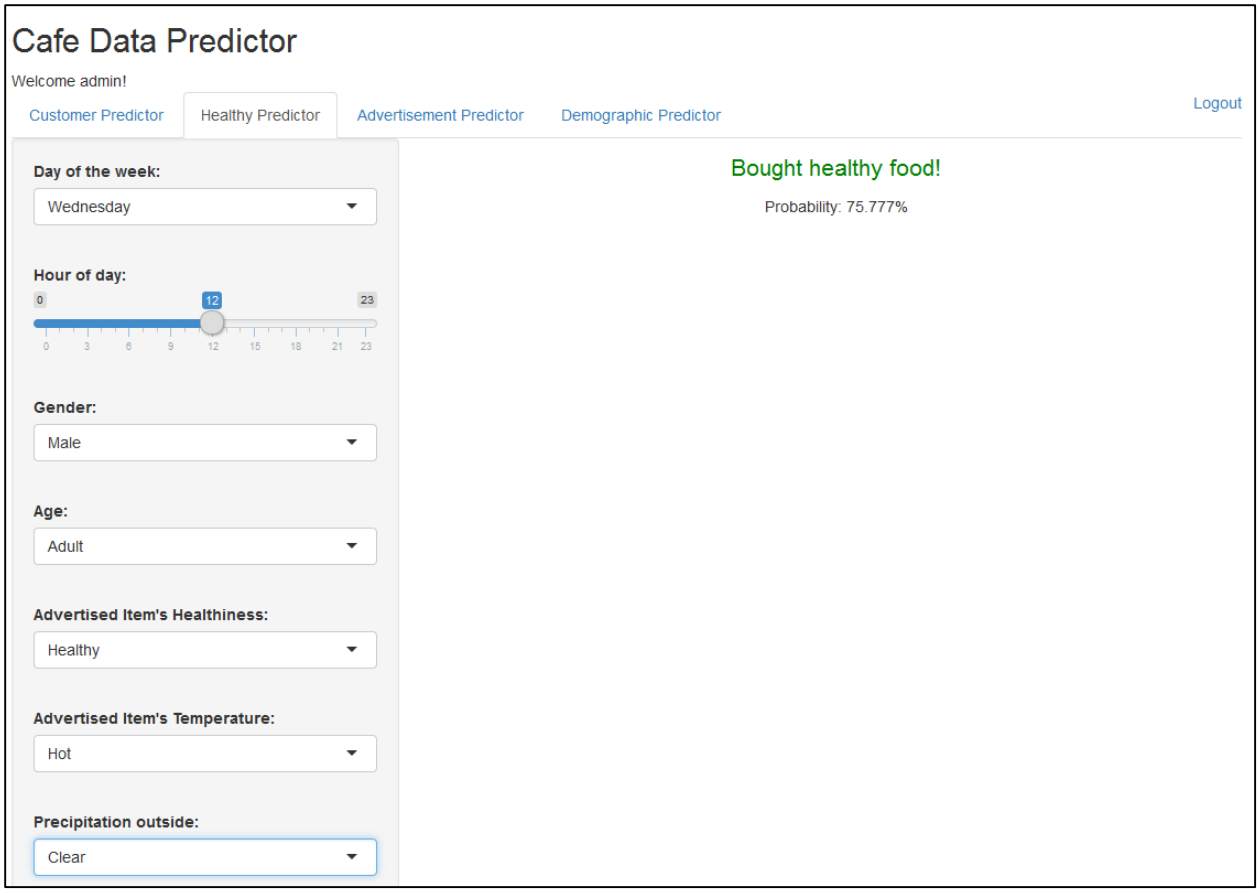

Figure 4: *Probability of customer purchasing healthy food screen*

Implementation

The initial issues with implementation of the project was finding trends within the data that could be predicted with high accuracy. The team spent a large part of the time allotted for the project to data exploration and not as much on creating the prediction models and dashboard as originally anticipated.

While Shiny was useful for quickly creating a dashboard in R out of the predictive algorithms we created, its limited on its customization. Inserting things such as JavaScript or CSS style is clunky or requires extra files and libraries. Further time for this project would allow us to embed the Shiny dashboard's graphs and other output into an existing web application that would allow for more customization. Since we used the free, open-source version of Shiny, some of its more advanced features were not available. Things such as sending specific data from the server to the client and back, such as login status, was tricky and required some unsecure methods to do this, while the paid version of Shiny has these features.

The project's predictive models are currently developed from a small set of data, all of which occurred in December 2015 and January 2016, therefore some of the predictions may be

incorrect at different times of the year. The overall accuracy of our models can likewise only be tested against different subsets of the data so currently it is impossible to tell how accurate the models will be at a different point in time.

The dashboard for our project was tested in most modern browsers, but has not been tested in all browsers, such as older versions of Internet Explorer. Test cases for the dashboard include:

Test Case	**Result**
Valid login credentials	User logged in
Incorrect login credentials	User notified username/password incorrect
Valid username/password in registration	User registered
Existing username in registration	User notified username already exists
Passwords do not match in registration	User notified passwords do not match
Username less than 4 characters	User notified username must be at least 4 characters
Password less than 8 characters	User notified password must be at least 8 characters
Valid input for predictors	Predictive algorithm successfully executed and displayed
Invalid input for predictors (such as characters in a numeric input field)	Predictor reverts to a default value to insert into predictive algorithm and then displays the result to the user

For most invalid input cases in our predictive algorithms, Shiny is able to validate the input for us so our dashboard code only has to check that the input provided is valid or not, and if not revert to a default value for that specific input value. Further test cases were created for our prediction accuracy functions such as R^2 or MASE to validate that they are giving correct accuracy estimations.

Future Enhancements/Maintenance

Future enhancements to this project would allow for the refinement of current models and the addition of new models using a larger data set. With the addition of larger amounts of data over time the prediction models would become more and more accurate. If a continuous stream of data could be gathered overtime, the project's predicting algorithms created from the machine learning process could be automated with the new data so that it automatically becomes more accurate overtime with minimal human interaction. Likewise, future enhancements would be needed for the login and security system of the dashboard. Using only the Shiny library in R there was little way to check if a user is logged in beyond sending the client a variable that their credentials are correct/incorrect as we used the free version of Shiny server. Future enhancements to this would include using the Shiny Server pro, which includes user permissions, or using another, more secure, method of checking a user has logged in and have the dashboard embedded in a web-page they can access. Embedding a Shiny dashboard into an existing web application could be done by adding R and required libraries to the web server in order to execute the dashboard's code.

Conclusions

The project was ultimately successful in producing a functional product which allows users to forecast customer activity with sufficient accuracy. The client was satisfied with the quality of Unfortunately, the effectiveness of the predictive models within the product is limited by various factors. For instance, the accuracy of predictions in one model is drastically reduced as the temperature parameter is increased past a certain point. This is largely due to the scope of the provided data set, which only covers the months of December and January. This means that our models were unable to train on data points with temperate to high external temperatures, and their accuracy suffers with higher temperature as a result. As discussed above, our product could be greatly improved with an extended data set, which would cover a wider variety of inputs and establish greater consistency in customer activity.

Near the conclusion of the project, we decided to remove two models from the dashboard because they were unable to achieve sufficient accuracy. Of the models that were removed, one attempted to predict customer populations based on demographic throughout the day, and one which predicted whether or not a customer would purchase an advertised item. This leaves us with two remaining models, one which predicts customer populations throughout the day based on weather, and one which predicts whether or not a customer will purchase healthy food items. While the inclusion of the other two models would clearly have improved the scope of our product, the remaining models still provide significant utility and information for the user, and allow our project to satisfy the client's specifications.

In addition to creating a useful product, our team has gained a wealth of knowledge and experience in machine learning techniques throughout the course of the project, particularly with respect to its utility in analyzing and deriving useful information from large, complex data sets. Specifically, we have learned that regression is a particularly useful tool for identifying linear relationships between continuous independent variables, such as temperature, and a dependent variable, while recognizing that nonlinear relationships require more time and understanding to analyze. Decision trees, on the other hand, prove much more useful for analyzing the relationships between categorical independent variables, such as weather conditions, and a dependent variable. We find that such lessons will be invaluable for future machine learning or big data analysis projects that we may be a part of.

User Manual and Installation Guide

Starting the Application

Because the application is web-based, no installation is required. To begin utilizing the application, open a web browser (we tested heavily on Google Chrome and Mozilla Firefox, though other browsers should be compatible), and load https://cafe-predicting.shinyapps.io/Dashboard/ into the address bar.

Login

When you load the application, you will be presented with a login screen. If this is your first time using the application, you will need to create an account. To do so, use the tab space at the top of the page and click "Sign Up". You will be presented with three text boxes: the first in which to enter a username, the second to enter a password, and the third to confirm your password. Usernames must contain at least four characters, and passwords must contain at least eight characters. When you have finished, click the button that says "Sign Up". If you provided a valid username and password, you will see a message stating that your account was registered successfully. You may now navigate back to the login tab.

At the login screen, you may enter a previously created valid username/password combination, and then press "Submit". If the credentials you entered were not valid, you will receive a message stating "Incorrect username or password", and you will be allowed to reenter your username and password. Upon the submission of valid credentials, you will be taken to the dashboard.

Navigation

The dashboard may be navigated using the tabs at the top of the page. These tabs will allow you to select the various prediction models provided with the product. Currently, only two models, "Customer Predictor" and "Healthy Predictor", are available, though more may be added in the future. At the conclusion of your session, you may select the rightmost tab to log out of the dashboard.

"Customer Predictor" Model

This model allows you to select the day of the week, input the external temperature, and select the level of outside precipitation in order to predict the number of customers that will enter the shop/cafe over a time period which you specify. The day of the week is selected from a drop-down box, temperature (in degrees Fahrenheit) is entered into a text box, and precipitation is selected from a drop-down box which includes the options: clear, cloudy, drizzling, fog, misty, raining, and snowing. The time period is specified using a 24-hour slider bar with two sliders, one for the start time, and one for the end time. You may refer to Figure 3 for reference. As you adjust these parameters, a display window on the right will automatically update to include a graph of the amount of customers entering per minute over the selected time period, and the expected total number of customers to visit during that time period.

"Healthy Predictor" Model

This model allows you to estimate the probability that a particular customer will purchase healthy food items by selecting the day of the week and time of day, the customer's gender and age, whether or not healthy food is currently being advertised, whether the advertised item is a hot or cold food item, and the level of outside precipitation. The day of the week is selected via a drop-down box, and the hour of the day is selected using a 24-hour slider bar. The customer's gender and age are also selected from drop down boxes, and the customer may be classified as either male or female, and as a child, young adult, adult, or senior. The advertised item's traits are also selected from drop down boxes, and it may be classified as healthy or unhealthy, and hot or cold. Finally, the outside precipitation is selected from a drop-down box which includes the options: clear, cloudy, drizzling, fog, misty, raining, and snowing. You may refer to Figure 4 for reference. As you adjust these parameters, the display window on the right will automatically update to indicate whether the customer is more likely to have bought healthy food, or to have purchased only unhealthy food items, with the calculated probability of the selected outcome being displayed directly underneath.

Logging Out

When you are finished using the application, you may select the rightmost tab in the tab space. You will be logged out of the application and returned to the login page, where another user may login, or where the browser can be exited until later.

References

- Lantz, Brett (2013). *Machine Learning with R*. Birmingham, UK: Packt Publishing
- R Development Core Team. *The R Manuals*. Web. Accessed May, 2015. <https://cran.r-project.org/manuals.html>
- Maechler, Martin. *The R Manual*. Web. Accessed May, 2011. <http://stat.ethz.ch/R-manual/>
- RStudio. *Shiny – Function Reference (version 0.13.0)*. Web. Accessed May, 2015. <http://shiny.rstudio.com/reference/shiny/latest/>