Exploring Data Mining in Used Car Data: Drivetrain

University of Nevada, Reno
Department of Computer Science and Engineering
CS 425/625: Project Assignment 4

Team #8:
Patrick Austin, Mile Cilic, Eric Stutzman, Guang Xu

Instructors: Sergiu Dascalu, Devrin Lee

Advisors: Stephen Williams, Lei Yang

11 December, 2017

Table of Contents

Abstract **2**

Introduction **3**

Prototype Objectives **4**

Prototype Functionality **6**

Develop Prototype **7**

Prototype Evaluation **16**

Demo Prototype **17**

Changes Needed to Software Design **17**

Contributions of Team Members **18**

Abstract

Techniques for efficient analysis and machine learning with large datasets have become a major research area in computer science in the last twenty years. Such advances in data mining technology may enable businesses to target advertising to specific groups and maximize the effectiveness of a marketing budget by speaking directly to interested customers. However, data mining algorithms often perform poorly on very large, multidimensional datasets, and it can be difficult to extract results into useful, human-readable information.

In this project we present an approach to enable targeted marketing solutions directed at used car sales. Our solution is based on training a predictive model with a 50GB dataset of used car transactions made available to us by the analytics company Marketing Evolution. In this document we detail prototypes for the client-side app and predictive algorithm for our used car datamining system, which we call Drivetrain.

Introduction

In the "Exploring Data Mining in Used Car Data" project, Team 8 intends to use state-of-the-art data mining techniques together with an exclusive dataset to discover novel and valuable information about used car sales. By building a training model on a large used car sales dataset provided by Marketing Evolution, and using the model to make predictions about customer habits and sales patterns, Team 8 aims to enable predictive, targeted advertising approaches to make efficient use of an available marketing budget.

Working in collaboration with the firm Marketing Evolution and their advisor at Marketing Evolution, Stephen Williams, the team has been granted access to a 50GB dataset documenting used car transactions in the United States over the last several years. Using data mining algorithms and strategies learned from our faculty advisor Dr. Lei Yang, we aim to use a decision tree induction approach to train highly accurate predictive models. These models allow users to enter available demographic or financial details about a customer as input, and receive a recommendation of cars that customer is likely to be interested in as output.

Team 8 has created a functional prototype called Drivetrain in the interim between our previous design-oriented deliverable and this one. This document details the process of that prototype's design, explains its functionality, and provides an account of stakeholder feedback on our results.

The Drivetrain prototype focuses on four common anticipated use cases for our used car datamining system. Drivetrain details our vision for a client-side GUI where users can login to the system and navigate via logically structured and convenient menus, enter available information about a client or customer to be delivered to the server to query a prediction, and can receive and visualize the prediction results received from the server. The Drivetrain prototype also includes a working proof of concept for the generation of a predictive model from the used car dataset, which can be used to make predictions with a high degree of accuracy.

Project requirements have not changed only slightly in the course of prototype development, in a manner largely concerned with UI tweaks and types of prediction we expect to make available to users. These changes are documented in detail later in the document. Mostly, however, this deliverable is concerned with delivering and demonstrating a working, realized prototype of the requirements and design conducted in the previous project documents.

Prototype Objectives

When choosing what objectives to focus on with the prototype development, we first consulted the context diagram created for the project in a prior deliverable, shown in Figure 1, in order to choose systems that we felt were most appropriate for prototyping. Due to stakeholder expectations, consultation with the CS 425 teaching staff, and prior work that had been completed in the previous documents, we focused on four main use cases. These were (i) login and menu navigation, (ii) the user query input experience, (iii) the user interface for displaying and visualizing prediction results, and (iv) the server-side system used to generate a predictive model and give predictions . Overall, we encapsulated these use cases by prototyping (i) a mobile application for clients, and (ii) a proof-of-concept model-building and prediction-generating system to be run server-side.

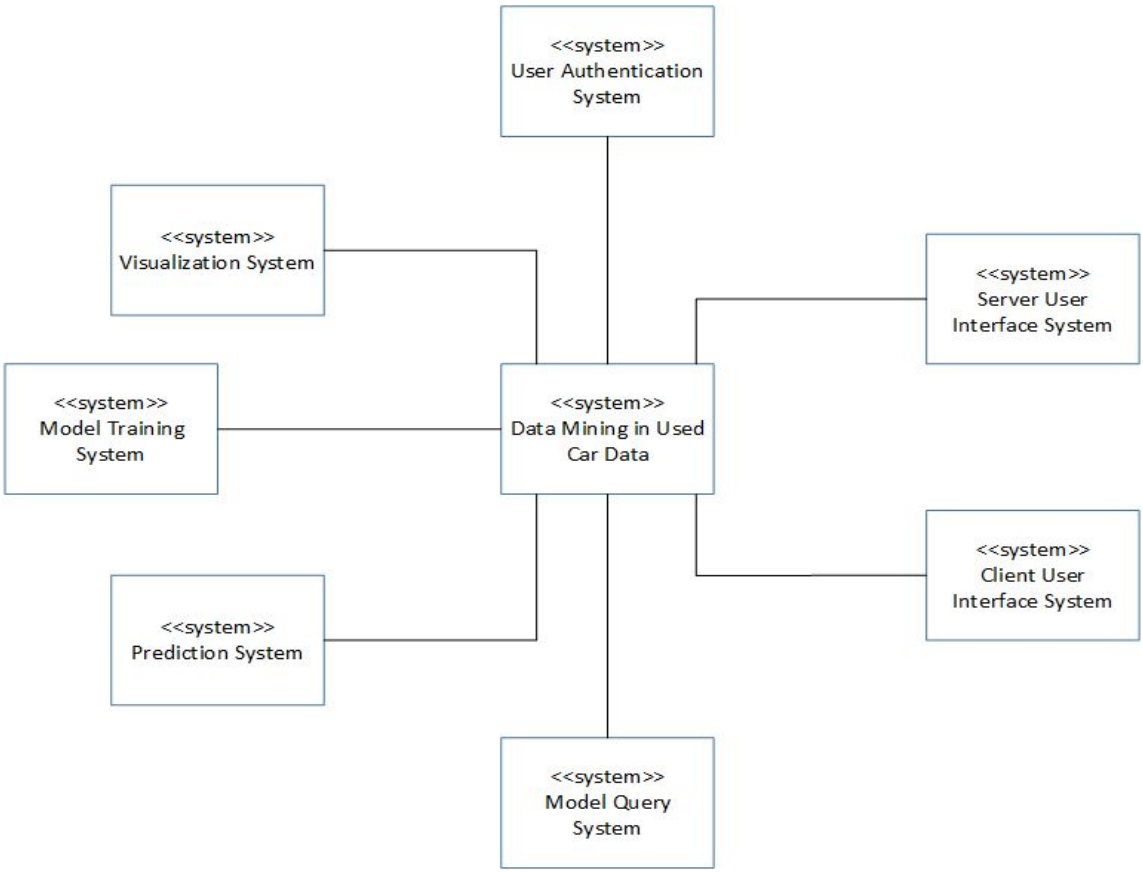


Figure 1: The context diagram for the used car datamining project. This diagram was closely studied to select prototype objectives and use cases. This diagram has been reproduced from the third project deliverable.

The mobile application encapsulates many common use cases for the overall datamining system, including login and convenient menu navigation, model query, and model prediction. Therefore creating a prototype UI of sufficient quality, convenience, clarity, and usefulness became one of the main objectives of our prototyping process.

The main objective of the prototype prediction module was to create a proof of concept for the overall process of model generation. With a model successfully generated, queries can be predicted as needed with low computational requirements. Goals for this model generation process largely involved achieving the highest possible rate of accuracy for the model generated, as well as generating the model in a way that would be scalable to the full 50GB used car dataset.

Prototype Functionality

Due to stakeholder expectations, consultation with the CS 425 teaching staff, and prior work that had been completed in the previous documents, we focused on the four main use cases mentioned in the previous section.

Encapsulating these use cases, we planned to demonstrate (i) basic user interface functionalities and (ii) model generation and query processes in this prototype.

We started with the basic mobile client application design. We chose to design this part first because most of our previous deliverables were focused heavily on user interface design and client experience, because this would give stakeholders a clear view of our goals for the project, and due to feedback from the CS 425 teaching staff. We wanted to make our product simple and convenient for everyday use. Furthermore, visualizing and implementing the user-interface would also enable us to more conveniently access our datamining results. This section implemented three of our four prototyped items: login and main menu navigation, query input, and query result display.

Having completed the user interface prototype, we also implemented a fourth prototype feature: a proof-of-concept model generation and query process. We chose to implement this next because it was integral to the core functionality of our application. Highly accurate prediction is notoriously difficult, and we deemed it best to get started and gain experience as soon as possible. How our predictions work, their degree of accuracy, and our methodology are all likely to be of vital interest to our stakeholders as well.

During the implementation process for these prototypes we have discovered several different tools and methods, faced many unexpected challenges, and realized several productive areas for changing the project design for the better.

We decided not to prototype the server user interface aspect of the original project context diagram for this prototype due to a lack of perceived stakeholder interest in a back-end convenience feature. In terms of server performance, model generation and prediction were deemed much more pressing tasks, from which higher quality feedback could be generated. Also, we did not prototype the user authentication system, but instead assumed that we already have a created account from which we can access app's features. While this system will be of some interest and difficulty in final design and implementation, we again focused on areas where we anticipated higher quality and more urgent feedback from our stakeholders.

Develop Prototype

We will briefly detail the client-side Drivetrain app prototype with “close to final” UI screenshots and descriptions. We also provide a few representative images of prototype operation from the module generation prototype point of view; as this is not a client-facing functionality no UI was implemented. Figures 2-8 document the prototype Drivetrain app. Figures 9-10 document the model generation prototype.

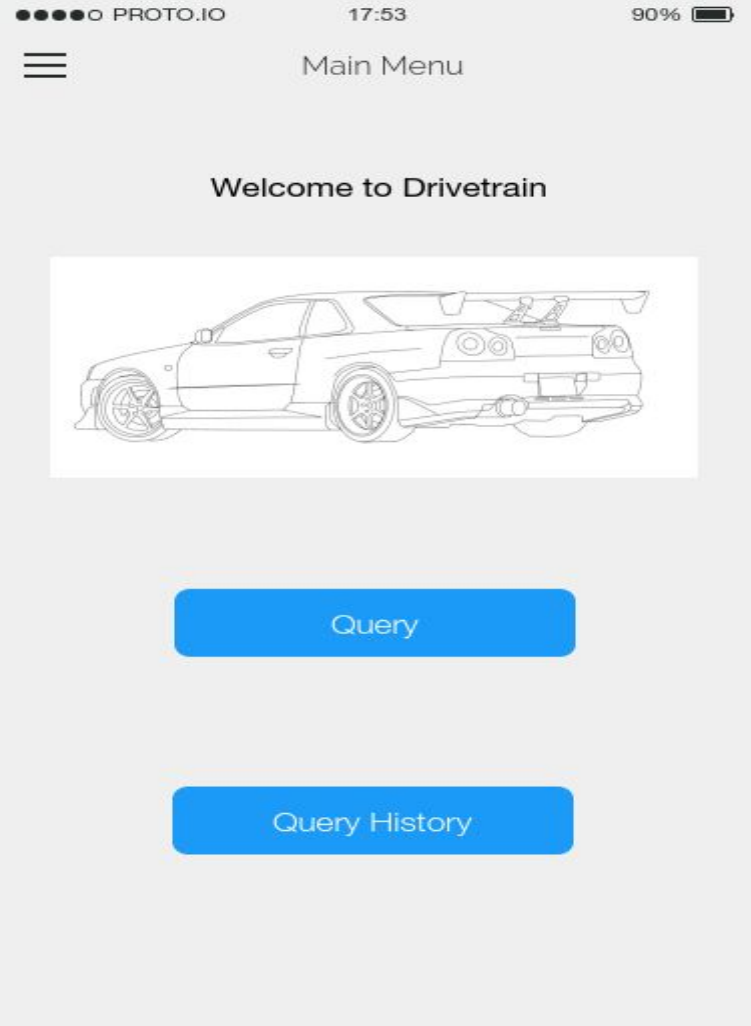


Figure 2: The Main Menu has a simple design reflecting the APP GUI activity diagram from the previous deliverable. It allows users to quickly navigate between Query and Query History, which we anticipate to be the most used parts of the app. The 3 horizontal lines at the top-left indicate the presence of a Slide-In Menu, which gives users access to less frequently used functions.

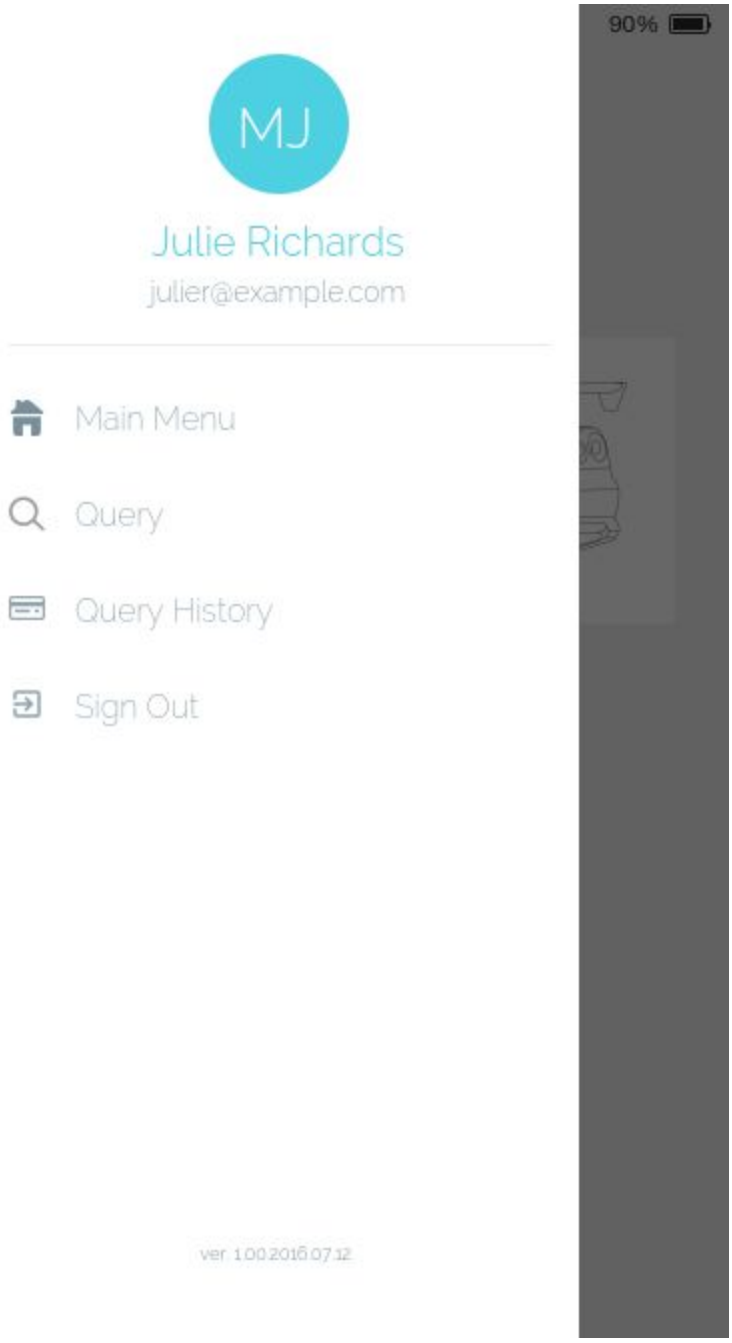


Figure 3: The Slide-In Menu is a part of every screen in the app except for Sign-In and Sign-Out. This allows the user to quickly determine if they are logged into the server. It also provides a menu navigation redundancy for most of the common functions. Furthermore, it allows a user to sign in, sign out, and sign up quickly and conveniently.

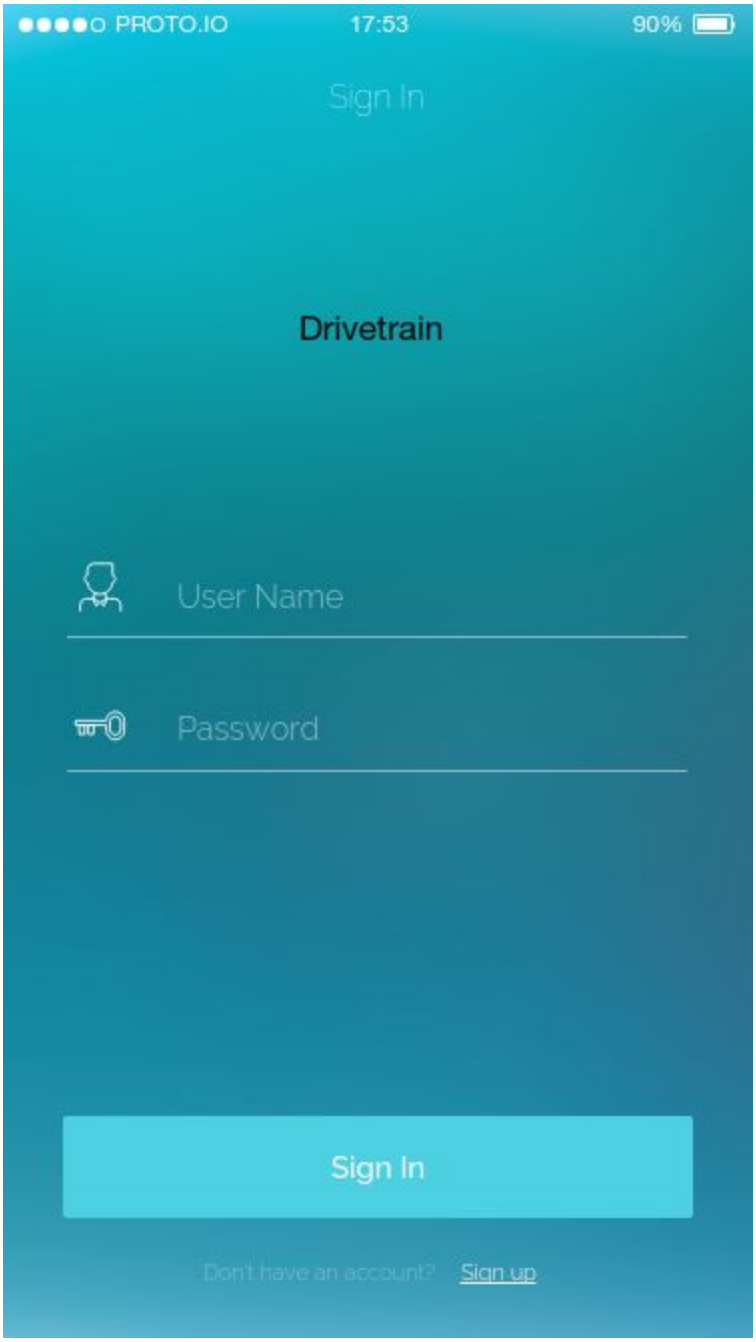


Figure 4: The Sign In screen allows a user to sign in. At the bottom, it also provides access to the Sign Up screen if the user does not already have an account.

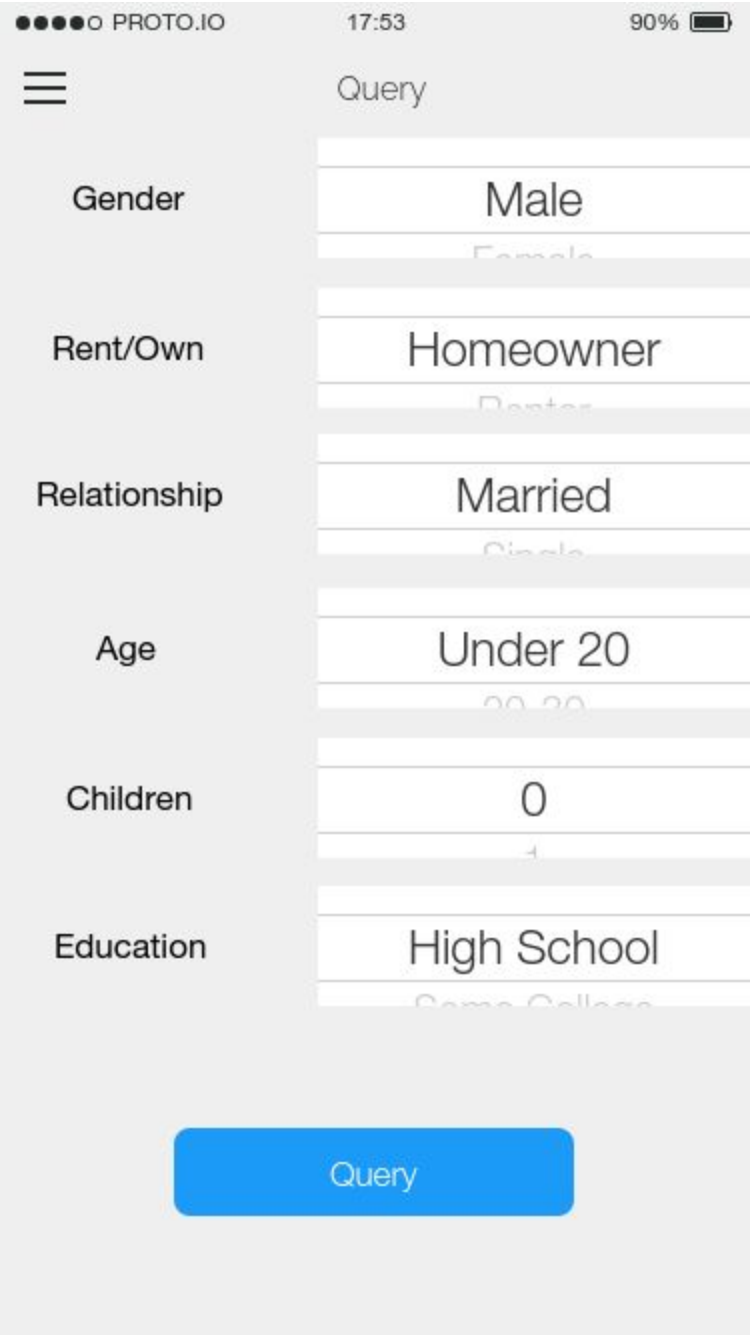


Figure 5: On the Query screen, the user is given multiple attributes to select to make a query. Each attribute has multiple values which can be easily selected to influence the final prediction. These attributes can be left blank as necessary if information is not obtainable.

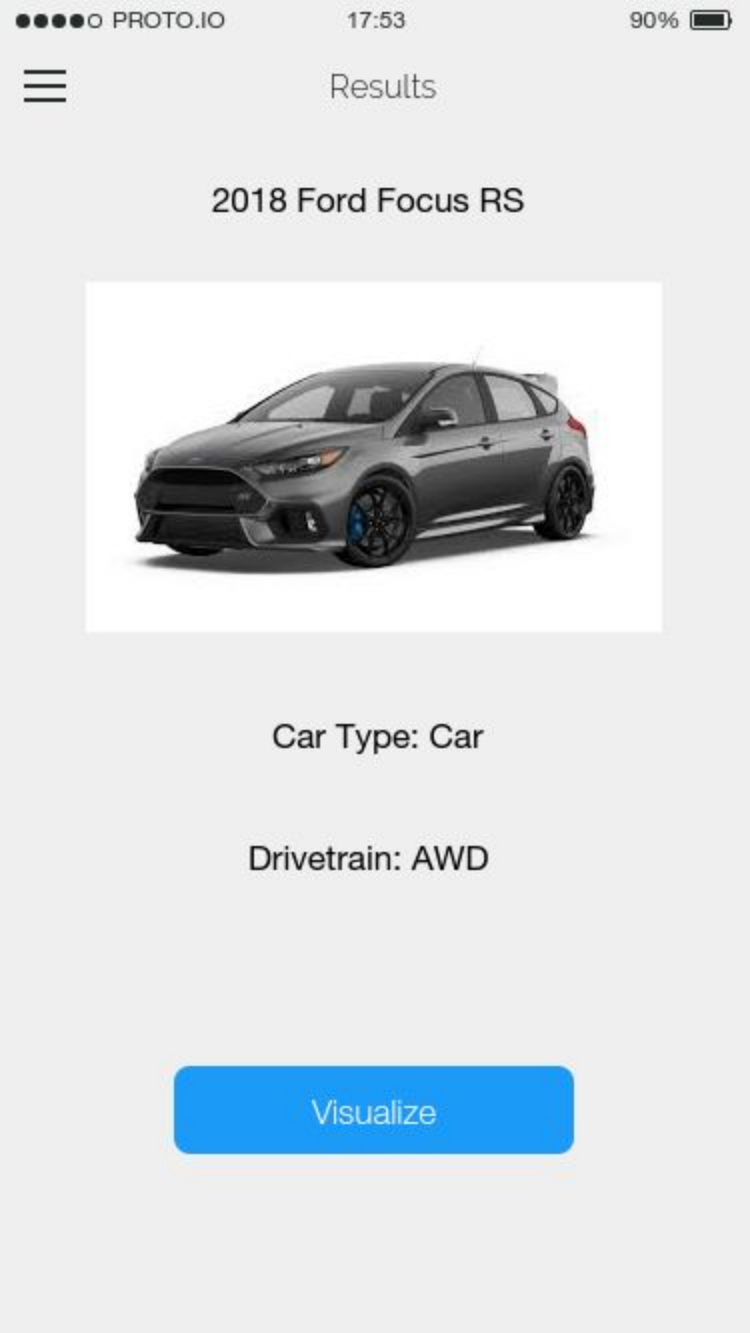


Figure 6: After a query is made, the user will be presented with a Results screen. This displays the predicted results given by the predictive models that have been generated. In this case, results for predictions on desired car model, desired car type, and desired drivetrain type are shown. The visualization button can be pressed for more detail about the prediction.

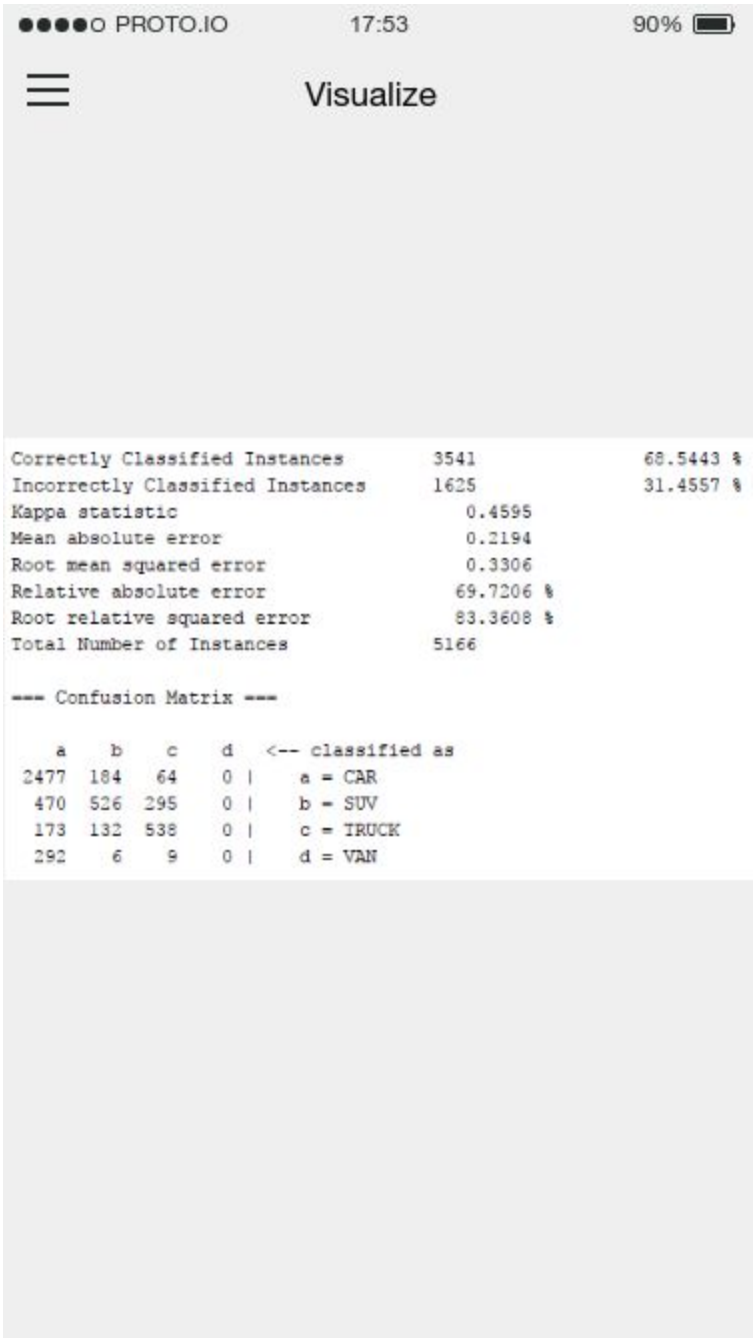


Figure 7: A simple prototype of the visualization screen. Due to stakeholder feedback, this is the screen in whose design we are least confident. Continued experimentation will follow during implementation to find the best way in which to provide human-readable visualization and statistics about the prediction process for those who do not have a data science background.

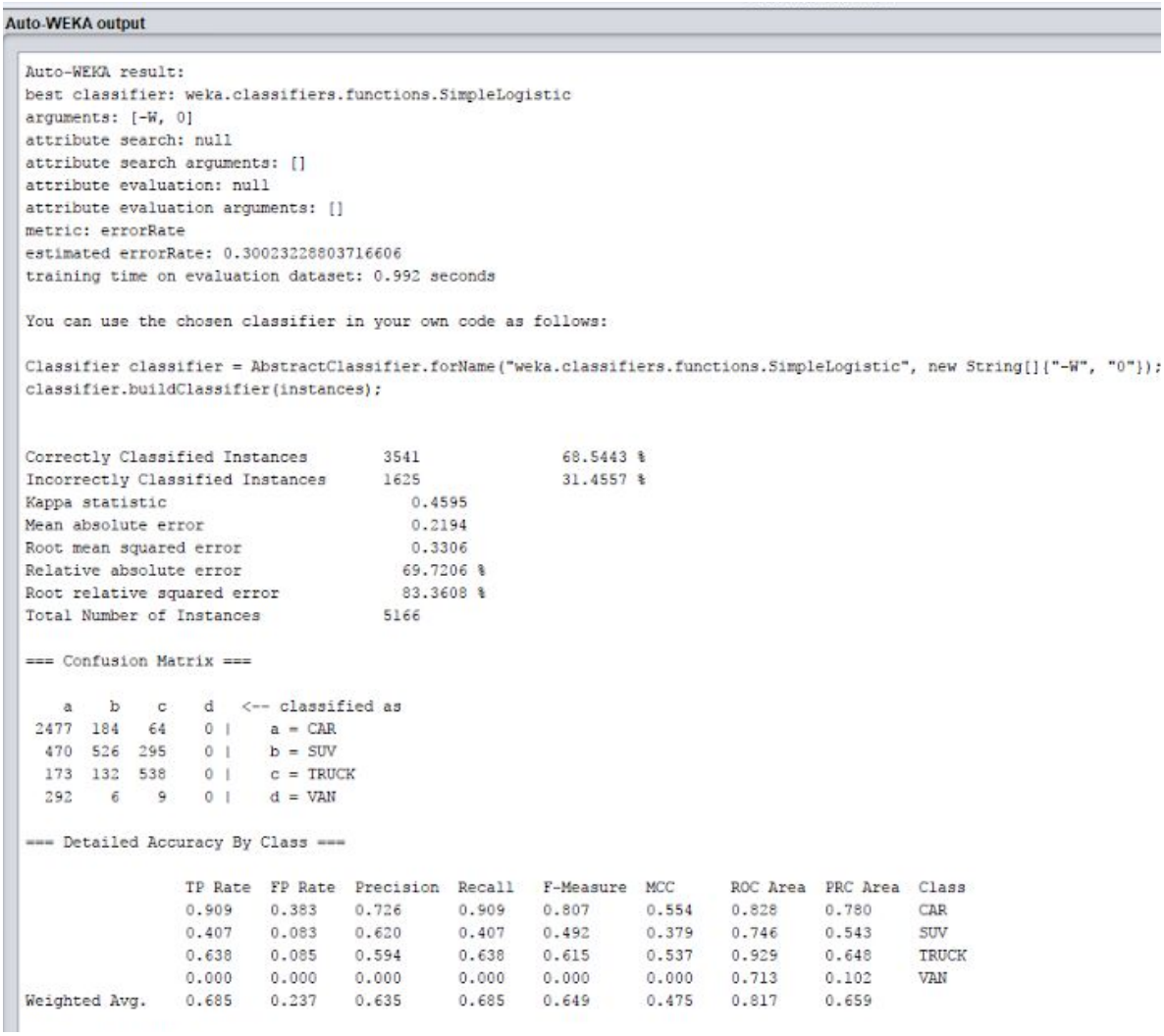
Figure 8: The query history screen allows the client who is using the Drivetrain app to quickly and easily retrieve the results of past queries so that their predictions can be accessed once more. This screen features query dates along with data input so that the desired query can be easily located.

```
@relation task2

@attribute 'Home Total Value' numeric
@attribute 'Estimated Current Home Value' numeric
@attribute 'CAPE: Educ: ISPSA' numeric
@attribute 'Ethnic Insight - Grouping Code' {K,O,Z,A,L,N,G,E,C,J,B,I,D,F,H}
@attribute 'CAPE: HHSize: HH: Average Household Size' numeric
@attribute 'Estimated Monthly Mortgage Pay - Amount' numeric
@attribute 'CAPE: Typ: HH: \% Married Couple Family' numeric
@attribute 'Person Number' numeric
@attribute 'Down payment \%' numeric
@attribute 'CAPE: Density: Persons Per HH for Pop in HH' numeric
@attribute 'Est. Household Income V5' {F,J,G,C,B,K,E,A,D,H,L,I}
@attribute 'Vehicle Type' {CAR,SUV,TRUCK,VAN}
@attribute 'Person Title of Respect' {MS,MR,MRS,MISS,REV,DR,RABI}
@attribute 'Home Purchase Price' numeric
@attribute 'Home Base Square Footage' numeric
@attribute 'Rural Urban County Size Code' numeric
@attribute 'Person Exact Age' numeric
@attribute 'Dwelling Unit Size' {A,F,I,E,C,G,H,D,B}
@attribute 'Children: Age 16-18 Score V3' numeric
@attribute 'Drive Type' {RWD,AWD,FWD,4X4,4WD,4X2}
@attribute 'CAPE: Age: Pop: Median Age' numeric
@attribute 'CAPE: Age: Pop: \% 18-99+' numeric
@attribute 'Dwelling Type' {S,A,P,M}
@attribute 'CAPE: Educ: Pop25+: Median Education Attained' numeric
@attribute 'County Name' {MONTGOMERY,BERGEN,HARFORD,POLK,MIAMI-DADE,STARK,CUMBERLAND,HALIFAX,CU

@data
141,191294,3536,K,273,581,604,2,0,273,F,CAR,MS,99,15,1,50,A,3,RWD,4100,751,S,122,MONTGOMERY
1348,1950594,4372,O,314,7288,768,2,23,314,J,SUV,MR,1050,?,1,77,A,8,AWD,4400,742,S,149,BERGEN
120,142802,3899,Z,235,755,459,1,2,235,F,CAR,MR,85,10,1,61,F,1,FWD,3400,780,A,127,HARFORD
117,132203,3006,K,226,0,228,1,0,226,G,TRUCK,MR,?,?,2,46,A,3,RWD,2800,703,S,119,POLK
219,278759,3063,O,347,611,615,1,22,347,C,SUV,MR,88,?,1,78,A,1,FWD,4200,824,S,117,MIAMI-DADE
64,61065,3105,K,229,0,453,1,0,229,B,VAN,MRS,?,9,2,82,A,1,FWD,4400,800,S,117,STARK
?,?,3634,K,243,0,495,3,0,243,K,CAR,MR,?,?,3,66,I,6,RWD,4400,796,A,119,CUMBERLAND
19,?,3209,A,232,0,469,3,0,232,F,CAR,MS,?,?,6,23,A,6,FWD,4600,791,S,117,HALIFAX
127,137180,3470,K,272,530,677,1,0,272,B,CAR,MS,?,10,1,64,A,6,FWD,4200,785,S,122,CUYAHOGA
?,?,3085,K,187,0,318,1,0,187,E,TRUCK,MR,?,?,3,35,I,2,RWD,5000,875,A,118,ALLEN
?,?,3754,K,244,0,405,1,0,244,A,CAR,MS,?,?,2,66,F,3,FWD,3300,731,A,125,WASHINGTON
?,?,4260,L,201,0,337,2,0,201,C,CAR,MS,?,?,1,?,E,1,FWD,3900,849,A,141,'LOS ANGELES'
187,221681,3634,O,290,1160,734,1,0,290,F,CAR,MR,?,13,3,53,A,5,FWD,4200,750,S,126,PUEBLO
49,86998,2755,O,368,419,613,1,0,368,E,CAR,MR,?,9,1,49,A,21,FWD,2800,646,S,113,LIBERTY
51,81774,3154,K,258,0,681,2,0,258,D,VAN,MRS,?,13,2,75,A,1,FWD,4900,830,S,118,MAHONING
?,?,2930,K,255,0,577,2,0,255,F,CAR,MRS,?,?,2,57,A,2,FWD,4100,802,S,117,LAWRENCE
?,?,3144,O,276,0,571,2,0,276,B,CAR,?,?,?,2,?,C,1,FWD,3300,688,A,119,HIDALGO
59,101178,2744,K,317,0,413,1,0,317,D,SUV,MR,?,8,1,?,A,1,FWD,3400,727,S,116,'PALM BEACH'
?,?,3571,K,226,0,469,2,0,226,E,SUV,MS,?,?,6,?,A,4,4X4,3400,833,S,132,LAWRENCE
?,?,3934,A,169,0,190,2,0,169,B,CAR,MS,?,?,2,?,E,3,FWD,4200,832,A,122,GUILFORD
375,418065,4055,K,242,1436,558,1,0,242,H,SUV,MS,?,13,3,53,A,7,4WD,4400,807,S,132,'ST MARYS'
131,?,3157,K,251,0,501,3,0,251,F,CAR,MR,?,12,4,74,A,1,RWD,4200,789,S,117,HURON
```

Figure 9: A representative image of the processed subset of the used car dataset selected for model generation. This file is in .arff format for use with the Weka datamining platform. The selected attributes for prediction are denoted by the @attribute tag, and the data instances with these attributes follow the @data tag. Missing values are denoted by a ? in the .arff format.



Auto-WEKA output

```
Auto-WEKA result:
best classifier: weka.classifiers.functions.SimpleLogistic
arguments: [-W, 0]
attribute search: null
attribute search arguments: []
attribute evaluation: null
attribute evaluation arguments: []
metric: errorRate
estimated errorRate: 0.30023228803716606
training time on evaluation dataset: 0.992 seconds

You can use the chosen classifier in your own code as follows:

Classifier classifier = AbstractClassifier.forName("weka.classifiers.functions.SimpleLogistic", new String[]{"-W", "0"});
classifier.buildClassifier(instances);


Correctly Classified Instances        3541               68.5443 %
Incorrectly Classified Instances      1625               31.4557 %
Kappa statistic                          0.4595
Mean absolute error                      0.2194
Root mean squared error                  0.3306
Relative absolute error                 69.7206 %
Root relative squared error             83.3608 %
Total Number of Instances            5166

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 2477  184   64    0 |    a = CAR
  470  526  295    0 |    b = SUV
  173  132  538    0 |    c = TRUCK
  292    6    9    0 |    d = VAN

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.909    0.383    0.726      0.909    0.807      0.554    0.828     0.780     CAR
                 0.407    0.083    0.620      0.407    0.492      0.379    0.746     0.543     SUV
                 0.638    0.085    0.594      0.638    0.615      0.537    0.929     0.648     TRUCK
                 0.000    0.000    0.000      0.000    0.000      0.000    0.713     0.102     VAN
Weighted Avg.    0.685    0.237    0.635      0.685    0.649      0.475    0.817     0.659
```

Figure 10: This image shows output from the Weka datamining platform GUI after model generation for the dataset from Figure 9 had been completed. Weka conducts a process called hyperparameter optimization to select the best possible model and model parameters for a given dataset. The model achieves approximately 70% accuracy, and we are confident there is significant room for improvement. This model can be used to make predictions when given new data instances.

Prototype Evaluation

Stephen Williams (Industry Sponsor, Marketing Evolution)

Stephen Williams gave us mostly positive feedback, and said he was pleased with the prototype's functionality. Initially, we did not have a query button on the home screen, so Stephen suggested that one should be added. That way, a user can easily figure out what to do and where to go for core app functionality without having to look in the menu. Stephen's feedback was mostly focused on the app prototype, which we worked to make as realistic as possible. Overall, Stephen said that the team was on the right track and that the app's functionality was likely to satisfy users. Of course, this was said for a prototype, and he voiced his assumption that the final app would have complete functionality and more polished user interface design.

As far as choice of data mining software, Stephen suggested the AWS machine learning platform since it allows for API endpoints that could connect with the app's backend with minimal difficulty. He also emphasized that Dr. Yang would be better suited to answer questions regarding data mining platforms and obtaining high quality results. This led us to consult with Dr. Yang on choice of datamining platform, for which Weka was ultimately used.

Lei Yang (Faculty Adviser, UNR)

Dr. Yang provided feedback with a different focus from that which was offered by Stephen. Dr. Yang strongly emphasized the accuracy of our algorithm and focused mainly on the model generation and prediction aspects of the prototype, with less concern for our UI design. However, he did question our choice of data to feature in the visualization tab of the client-side UI, prompting us to iterate on its design to find a higher standard of quality.

In terms of attribute selection and data quality, Dr. Yang encouraged us to conduct further experimentation and testing by adding or removing attributes from consideration by the model building algorithm. He emphasized that the measure of quality for a predictive model is its accuracy, encouraging us to reach for higher accuracy rates during the app development process.

Dr. Yang also gave us interesting feedback on the challenge of visualizing prediction models and in a useful, human-readable way. While we can easily generate statistical data about a trained predictive model, this data is largely of use to a data scientist and adds little value to the app from a layman's perspective. Dr. Yang explained that this was a challenge for the field in general and encouraged us to do further experimentation in this area.

Demo Prototype

We will demonstrate our prototype to instructors for the course at 2:00 PM on December 12th.

Changes Needed to Software Design

After receiving prototype evaluation from our stakeholders, we added more navigation features to improve the flow of the app. Initially we only used the Slide-In Menu for navigation, but after testing and feedback we decided to add Query and Query History to the Main Menu to streamline these selections. We also removed a graph whose meaning was unclear from the Visualize section, since we were advised that it would not be useful. To improve visualization, we added relevant output metrics from Auto-Weka. Visualization is an area for which we are continuing to conduct experimentation and collect feedback on how best to convey human-readable information about the prediction process to users without a data science background.

When developing the app prototype it quickly became apparent that Visualization was not needed on the Main Menu as previously specified on the APP GUI activity diagram. Our initial idea of choosing to Visualize the current result didn't make sense once we started using the prototype. However, we still need the option to visualize the results so we chose to remove Visualize from the Main Menu, and link it to the Results page. From the APP GUI diagram, this allows navigation to Visualize from both Query and Query History.

After developing our proof-of-concept model in Weka, we observed that it would be straightforward to predict on several additional attributes of the dataset besides the initial one we chose, specific car model. This prompted us to add predictions on car type (i.e. van, truck, car, SUV) and drivetrain type (i.e. full-wheel drive, front-wheel drive, rear-wheel drive) to the results screen.

Additionally, we are still in the process of selecting a scalable and powerful datamining platform for use in final implementation. Rather than use Weka as in this prototype, we are likely to use the Scikit-learn or Orange platforms in Python for implementation.

Contributions of Team Members

Figure 11 shows a table of primary team member tasks and time worked on each task.

Team Member	Task/Contribution	Time Worked
Patrick	Proofreading, editing	1 hour
	Abstract	30 minutes
	Introduction	1 hour
	Prediction/query system implementation	4 hours
Mile	Prototype objectives	1 hour
	Prediction/query system implementation	1.5 hours
	Stakeholders feedback	1 hour
Eric	App prototype Development	5 hours
Guang	Paper design and editing	1 hour
	Meetings and team coordination	2 hours

Figure 11: Table of group member contributions.

In addition to time spent working on sections individually, the team conducted several meetings to coordinate, discuss, and ensure the quality of the document. In addition to stakeholder meetings and the prototype presentation to the teaching staff, the team also met with a member of the teaching staff, Devrin Lee, to discuss questions and concerns with a focus on use case selection for prototyping. These meetings lasted approximately 5 hours in total.