FuelSense

Swapnil Avinash Parihar
MS Software Engineering
San Jose State University
San Jose, California, USA
swapnilavinash.parihar@sjsu.edu

Pranav Karmalkar
MS Software Engineering
San Jose State University
San Jose, California, USA
pranav.karmalkar @sjsu.edu

***Abstract*— Modern automotive companies spend large sums of money trying to meet government-regulated standards for fuel-efficiency. By creating predictive software capable of estimating the conditions of optimal function, a lot of effort and money could be saved which is typically spent on redesigning. FuelSense is an application that aims to provide the manufacturers with a tool to analyze mileage for different types of fuels and road conditions. This paper compares some of the most common machine learning algorithms including Linear regression, SVR, and Random Forest to find the most efficient algorithm suited for a small and labeled input dataset. Various visualization tools are presented to help the users effectively analyze the input dataset and compare machine learning algorithms output with the default dataset. With the help of IBM Watson, the project also aims to study the applications of chatbots as a Web navigation tool.**

Keywords*— *Regression algorithms, IBM Watson chatbot, Random Forest

I. INTRODUCTION

FuelSense deals with an input dataset that has 9 different parameters including but not, limited to distance, speed, environmental conditions in which the test car was driven for a year. The dataset provider also toggles between different fuel types for a certain period resulting in an evenly distributed labeled dataset. FuelSense as a project aims to find the correlation between all these parameters and analyze how they affect the mileage of the automotive. It makes use of random forest ensemble learning method for regression, KNN imputation to fill in the missing data and IBM cloud with Kubernetes clustering is used to deploy the tool on the cloud.

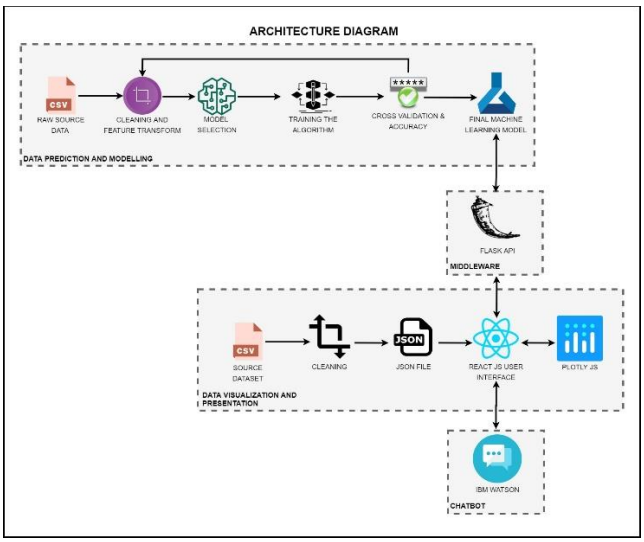

Fig. 1. Architecture diagram with Technology Stack for the project

II. TOOL DESIGN METHODOLGY

The project can be divided primarily into 3 main parts i.e. machine learning and flask API development, frontend development using react and node.js libraries and finally chatbot design using IBM Watson. Implementation details for the three parts have been discussed in this section

A. *React.js*

ReactJS has been used to design the UI for the application. Different plots like Multi-Parameter Analysis, Time Series Analysis, Outlier analysis have been displayed to let the user know about the data and understand it. The displayed plots vary from line plots to scatter plot to 5D plots. The web application is React single page application with a lot of focus on user friendliness, it is easily navigable, and the graphs are descriptive. Also, the transitions are smooth making the user feel like they are using a mobile or desktop app with no reloads of page etc. Finally, a lot of work has been done to make the code modular and easy to update with very little resources.

B. *Machine learning*

While the input dataset for the project is small and labeled it came with its own set of issues including missing data fields and incompatible data types such as string which needed to be resolved to start the machine learning model selection process. The dataset had a few rows with missing temperature value which could not be ignored owing to an already small dataset. Temperature normally follows a distribution pattern where variations in temperature have a maximum correlation with its nearest neighbors. Based on the analysis of different imputation techniques such as mean, substitution, Hot-Cold deck imputation; the k-NN algorithm was selected for data with such unique features and the value of k was computed to be 9 to minimize the influence of noise from the nearest neighbors. Moving forward, all input data types needed to be converted from strings to categories and then to integers to ensure that the machine learning algorithm can effectively understand it. The next step involved the selection of a machine learning model for data prediction. To train the model, input data was divided into an 80:20 ratio for train and test data respectively. Note that owing to the small size of the input dataset, no improvement was seen in moving from traditional machine learning models to deep learning models. Traditional machine learning techniques such as linear regression, Support Vector Regression(SVR) and, random forest regression was studied and compared. Figure 2 below shows the comparison of mean squared error for all the 3 algorithms. The random forest regression algorithm was selected as the final model owing to its low mean squared error value. The final model was then generated by taking the entire input dataset for training the algorithm to achieve maximum accuracy.

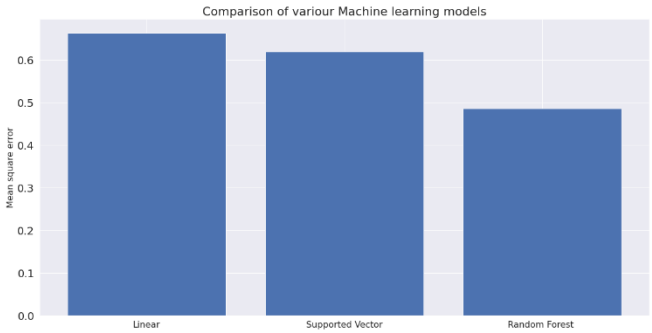

Fig. 2. Machine learning regression algorithm comparison

C. *IBM Watson Chatbot*

As chatbots are becoming a large part of our day to day life, this paper dives into its application as a tool for web navigation and enterprise user assistance. IBM Watson chatbot has 3 main building blocks i.e. intents, entities, and dialogs. Intents are goals that the chatbot is expected to meet which include greeting the user, providing information about the project, providing definitions of raw data and their units, and helping them navigate the website. For intents like greeting the user, it is a straightforward choice between a set of predefined responses that the bot can select from. Alternatively, for intents such as providing definitions and units for data variables chatbot first needs to know which variable the user is talking about. These "variables" or nouns for the projects are defined in the entity building block of the chatbot. For example, variables such as distance, consume and data outliers together form an entity for the "definition intent". Based on the variable the user is talking about the next step involves creating dialogs. Suppose the user wants to know "what is consume?" or "what are the units for 'consume' variable?". Both the questions can be answered by providing a detailed definition of the parameter like 'It indicates the mileage of the vehicle measured in l /100Km.' Thus with the help of a clear definition of scope including goals and intents, a chat-bot can become an intelligent conversational agent. The last phase involves employing reinforce learning techniques to improve the chatbot's accuracy of detecting the correct intent.

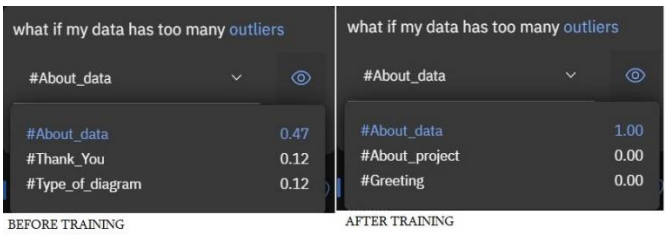

Fig. 3. Testing and training IBM Watson

III. Technology Stack

The project technology stack includes React and Plotly.js for frontend development, Python - scikit-learn, NumPy, Pandas, PyPI for machine learning implementation, Flask as a microweb frame work to act as a bridge between frontend and Machine Learning Model and finally IBM Cloud, Docker and Kubernetes for cloud deployment.

IV. Future Enhancements

FuelSense project functionality can be further enhanced by creating a support tool to automatically feed different input datasets using the webtool. This will enable the user to compare test data for more than 1 care model simultaneously. Adding Advanced ML analysis functionality to IBM Watson chatbot can also enhance the end-user experience.

V. Conclusion

The paper demonstrates techniques to develop data visualization and analytic tools that can enhance and streamline data sharing processing for an automotive design enterprise. Additionally, it sheds light on the applications of IBM Watson as a web-navigation and enterprise user assistance tool.

Acknowledgment

Our deep gratitude to Professor Rakesh Rajan for his guidance and useful critiques during the development of this project.

GitHub Project Repository

https://github.com/SJSUSpring2020-CMPE272/FuelSense

References

[1] A. Wagener, Car fuel consumption, Hamburg, Germany,2018. Accessed on: Jan. 28, 2020. [Online]. Available: https://www.kaggle.com/anderas/car-consume#measurements.csv

[2] B. S. Raschka and V. Mirjalili, Python machine learning, 2nd edition: Packt Publishing Ltd., 2017.

[3] C. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 1st edition: O'Reilly Media, Inc., May 2014.