Fuel Price Predictor Documentation

This document explains the functionality and background processes of the Fuel Price Predictor application as well as improvements that could be made in the future.

1. Necessary installations

Python 2.7 and some Python libraries are necessary to run this application.

1.1. Install Python2.7

Python 2.7 can be installed from this website https://www.python.org/downloads/release/python-2711/

1.2. Install libraries

Install pip from here https://pip.pypa.io/en/stable/installing/ . Pip is used to install the libraries.

You will need to pip install the following:

SciPy - https://www.scipy.org/install.html
StatsModels - http://statsmodels.sourceforge.net/devel/install.html

You may need other libraries. Run the PredictionModelARMA.py file from the command line to find which libraries you are missing.

2. Run FuelPricePredictor.exe

In the FuelPricePredictor folder, double click on FuelPricePredictor.exe to run the program.

2.1. User interaction

The application window has a number box in which the user must input the current price per litre of fuel. It is assumed the user is at a service station where they can see the current price of fuel.

The user must then either hit the PETROL or DIESEL button corresponding to the fuel they have the price information for. Petrol refers to 91 octane which is the most commonly used fuel. The user will then see a coloured word, either BUY or WAIT, on the window which tells the user to either buy or not buy the fuel due to the price. The colour of the word indicates the significance of the price difference to the predicted price in order to give more information to the user in an intuitive way.

The user can then hit the INFO button at the bottom right of the window to see more information about why they should buy or wait. The INFO window gives a quick explanation of the given message, for example the green BUY message's explanation is "Current price is significantly low." The current price and predicted prices are also shown on this window so the user can see how different they are. A graph of the recent historical change in the retail price of fuel is displayed at the bottom of this window giving the user an indication of the current trend in fuel prices. This graph will be different if diesel is selected than if petrol is selected. The user can press the BACK button to return to the starting window.

2.2. Background processes

2.2.1. Python file

When the .exe is run a command line window will come up as the Python code is executed before the application window opens. This Python code loads in the data from the .csv file located in the

same folder and uses an autoregressive–moving-average (ARMA) model to predict the current retail price per litre, and next week's price, of 91 octane petrol and diesel. These predictions are saved to a predictions.txt file which is then read by the application. The Python code also creates and saves two graphs. These are what are shown on the INFO screen of the application.

2.2.2. BUY or WAIT message

When the application is running and the PETROL or DIESEL button is hit the application tests the current fuel price against the normal variance for the predicted fuel price for the current week. Normal variance is assumed to be ±4 cents per litre, which was an estimation of the normal fluctuations in fuel prices between different service stations. The difference between the current price and normal variance of the predicted price is used to determine the message, and colour of the message, that is displayed to the user. If the current price is within predicted normal variance for the current week then the predicted price for the next week is also used to determine the message. The message colours are modelled on the traffic light colours in order to be as intuitive as possible for the user. The possible scenarios for price difference are explained below:

Scenario	Outcome	Reason
Current price is **above** predicted normal variance for current week	**Red WAIT** message displayed	Actual current fuel price is **significantly higher** than predicted current fuel price. User **should definitely wait** to buy fuel as it is currently expensive
Current price is **below** predicted normal variance for current week	**Green BUY** message displayed	Actual current fuel price is **significantly lower** than predicted current fuel price. User **should definitely buy** fuel as it is currently cheap
Current price is **within** predicted normal variance for the current week and **above** predicted normal variance for the next week	**Yellow WAIT** message displayed	Actual current fuel price is **normal** compared to predicted current fuel price **but significantly higher** than the next week's predicted fuel price. User **should wait, if possible**, until next week to buy fuel as although the current price is normal the price of fuel will likely reduce by the next week.
Current price is **within** predicted normal variance for the current week and **below** predicted normal variance for the next week	**Yellow BUY** message displayed	Actual current fuel price is **normal** compared to predicted current fuel price **but significantly lower** than the next week's predicted fuel price. User **should buy** as the current price is normal and the price of fuel will likely increase by the next week.
Current price is **within** predicted normal variance for the current and next weeks	**Black BUY** message displayed	Actual current fuel price is **normal** compared to predicted current fuel price and **normal** compared to next week's predicted fuel price. User **could buy** as the current price is normal and will likely be at a similar price next week.

3. Future improvement of application to overcome current limitations

3.1. Location based, daily information

Due to the scarcity of data on fuel prices in New Zealand the prediction model uses weekly national averages of fuel prices. This means the prediction model returns price predictions that can be wildly inaccurate compared to local daily prices of fuel. We recommend that the application be expanded to save location based daily fuel price information to a database. This would allow the application to

be updated to give more accurate predictions for day to day fuel prices in the area local to the user, as well as the ability for the application to notify users of cheaper fuel near their current location.

3.2. Preferences

The application could be updated to include preferences to be set when the user first opens the application and modifiable at any time. They could include options as described below.

3.2.1. Fuel types

The application can currently only predict fuel prices for 91 octane petrol and diesel. Having fuel-type as a selectable preference would allow the user to select a specific fuel type, say 98 octane petrol, and the predictions of fuel prices will be based upon data gathered on that fuel type. For the best results this should be implemented with the location based information database so that the data for these different fuel types can be stored.

3.2.2. Classes

The application currently uses a normal variance range of ±4 cents per litre. This does not take in to account how much fuel the user may purchase and with what frequency they purchase it. Having selectable classes (heavy, light, truck/transportation vehicle etc.) in the preferences would allow the normal variance range to be modified with regards to frequency of re-fuelling and fuel-usage per day. For example, a truck may use significantly more fuel per day than a light vehicle so smaller changes in fuel prices will have a more significant impact on the money they spend per week, therefore the normal variance range should be reduced.

3.3. Improved prediction model

The prediction model can be updated with additional functions to improve accuracy of predictions. Additions to the model could include web-scraping for world events that may affect import price of fuel and incorporating those effects into the model.