FIU

Engineering & Computing

School of Computing & Information Sciences

School of Computing & Information Sciences

Spring 2019 Senior Design Project

Data Science Workflow Management System

Student: Tomas Ortega

Mentor: Dr Miguel Alonso, Steven Luis, Florida International University

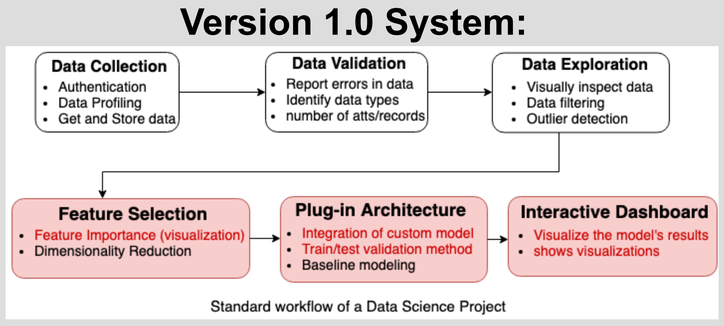
Instructor: *Dr. Masoud Sadjadi*, Florida International University

Problem

The airline industry does not currently have the capabilities to fully take advantage of the huge amounts of data generated every day to enhance customer experience, increase the efficiency of operations or make sounder data-driven decisions because they lack the tools to implement the solutions. We believe that following the latest data science tools and techniques, we can leverage valuable insights from airline data and increase revenue.

Current System

Version 1.0 System:



Standard workflow of a Data Science Project

Users of the web application are able to see and interact with graphs that contain predictions from machine learning models and move single or groups of points to change the training data. Then they can click on the train button to train the models. The user can also visualize frequency distributions involving attributes such as state of departure, market share and coupon distributions.

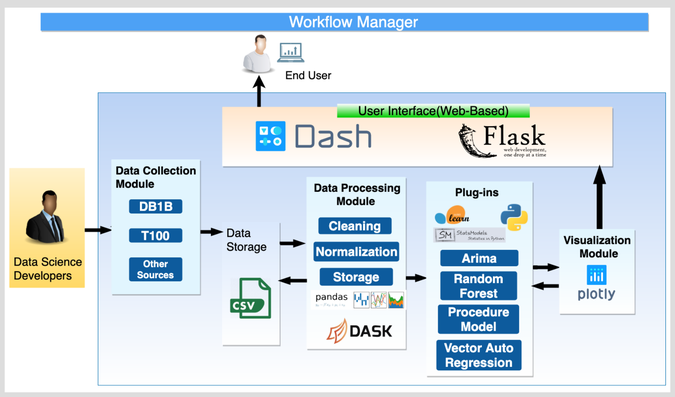
Requirements

The WFMG project requires the development of

- A **web front-end** that displays an interactive dashboard containing graphs with predictions and relevant information for the users to explore and interact with.
- A **back end** that uses the Plug-in system to create and train models, process data and manage user interactions.
- A **WFMG Module** (implemented in python) intended for providing the functionalities of custom machine learning models. This module should implement functions to load, fit, train, and save the models.

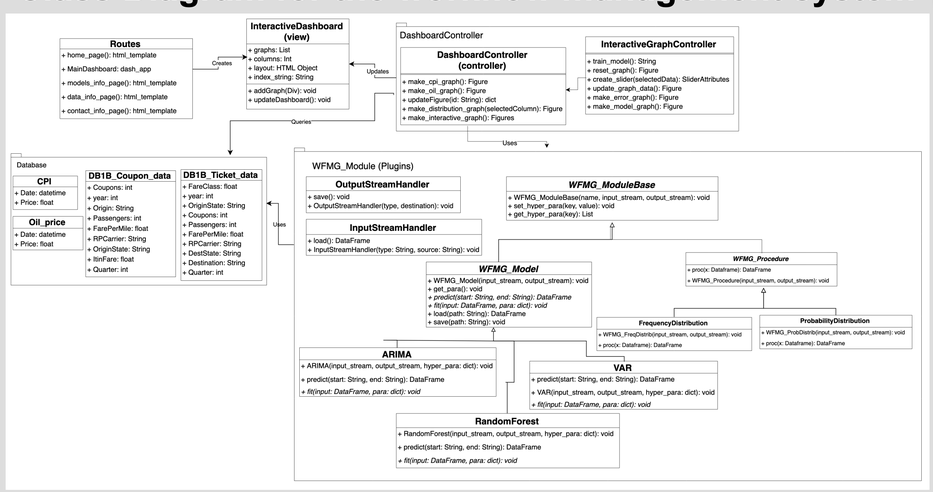
System Design

Architecture diagram



The Workflow Manager web application utilizes the **Model View Controller (MVC)** design pattern to provide different levels of abstraction to the developers working on new plugins or working on the view components. We used **plotly** for the visualizations because it provides interactive components in a web environment.

Class Diagram for the workflow management system

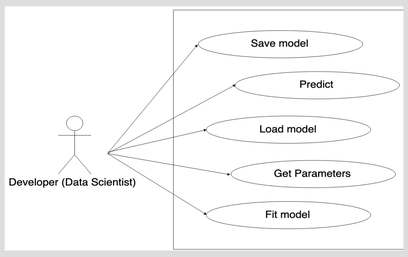


We created a system for Plug-ins using OOP concepts so that developers working on a data Science project can add new plugins into the pipeline so that they can be used in the visualizations or for data processing. The wfmg_model class and all its subclasses are currently being used to create the prediction data and to retrain the models.

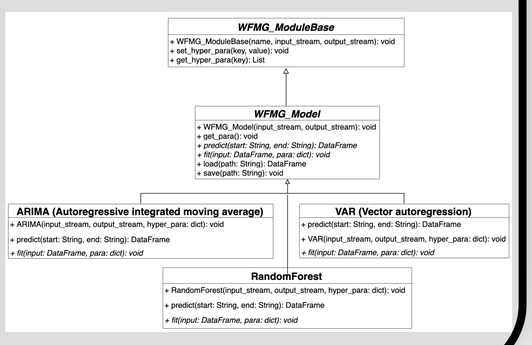
Object Design

Design of the **wfmg_model** python module for the plug-in architecture

- This module provides functions for **loading**, **fitting** and **saving** machine learning models. It also provides a function to generate new predictions from the visualiation module..

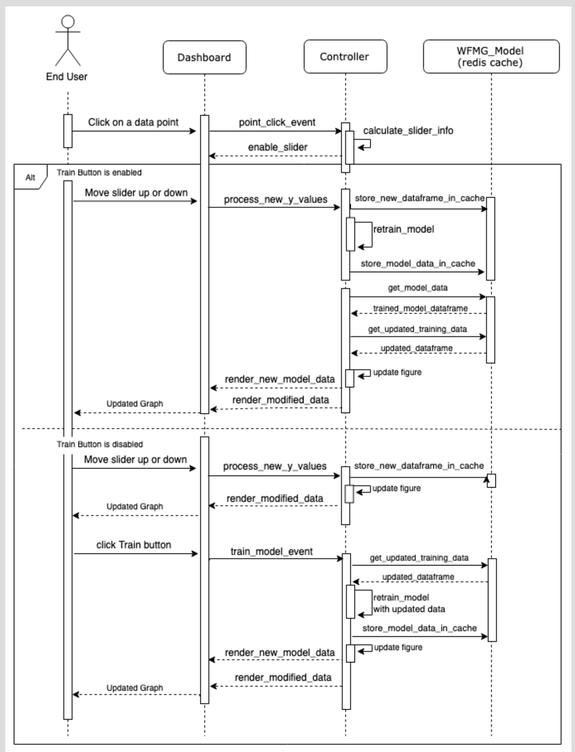


- The **WFMG** module contains interfaces that other plug-ins can implement to be accepted by the visualization module.



Implementation

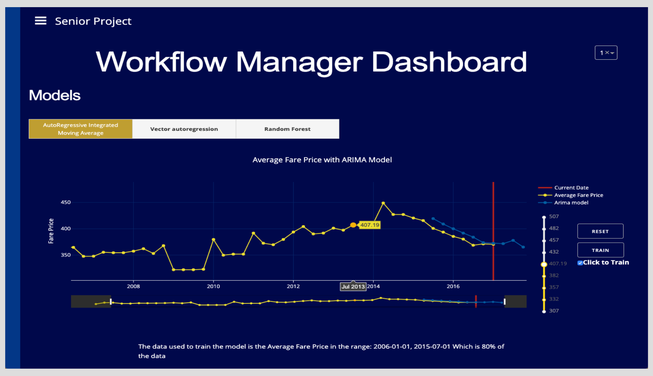
Sequence diagram for retraining the model when points are moved. It also includes the steps for updating the data used in the visualizations.



- We used methods provided by the dash framework to dynamically update the components of the UI to allow the user to smoothly interact with the dashboard visualizations
- The error graph is also updated every time the user makes a change to the graph.

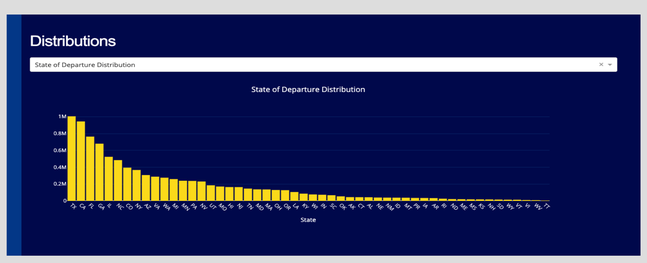
Screenshots

Visualizations from the dashboard page:



Fare Price Visualization and Model Prediction

Random Forest Feature Importance graph and Model Error



Frequency and Probability distributions plots

Verification

We used selenium to automate part of the testing process and performed unit and integration tests

- Sample unit tests.
- This particular test description is for the user story “movable data points” with ID “WFMG-34”

Test Case ID: WFMG-34-UT01	Sunny Day
Purpose	Validate that users are able to move individual data points using the slider on the right side of the graph and that the system is able to react to the point movement event.
Preconditions	The user must have the dashboard page loaded on the web browser.
Expected Result	The end user should be able to click one of the points in the data set used to train the model and move it vertically using the slider. When the user lifts the mouse, the graph should update and move the point to the desire position.
Actual Result	The results are as expected, the points of the graph that are part of the training data can be moved by the end user and the graph is updated when the user lets go of the slide.
Status	Passed

- These tests are used to verify the performance of the feature under different conditions.

Test Case ID: WFMG-34-UT02	Rainy Day
Purpose	Validate that users are able to move individual data points using the slider on the right side of the graph only if a point is selected.
Preconditions	The user must have the dashboard page loaded on the web browser and wait until the graphs are visible.
Expected Result	The end user should not be able to click on the slider until a valid point is selected. After a point is selected, the slider must show the current value of the point and a range of (+100, -100) from the current point for the user to choose a new value. If 0 points are selected, then disable the slider and ignore clicks on the slider.
Actual Result	The results are as expected, the selected point of the graph that are click training data can be moved by the end user and the graph is updated wh the slide.
Status	Passed

Summary

The WFMG project version 1.0 is now able to display the visualizations for the prediction data from several multivariate and univariate machine learning models. The Plug-in architecture enables developers to include new models or swap the current models for new ones.

The end user can interact with the machine learning models by modifying the training data and visualizing the results immediately

Acknowledgement

The material presented in this poster is based upon the work supported by the FIU AIRLab Team and the Farelogix Team. I am thankful for the help that I received from my group member Serge Metellus.