**Student Name:** Denise ‘Orah’ Kittrell

**Project Name:** DevNet

**Date:** 03-08-2015

**CPDM Capstone Final Project Requirements**

1. **PROCESS(ES)**
   1. The first process is a recommender system. A recommender system provides users with an item based on a predictive analytic model, think Netflix’s movies recommendations. The recommender system in DevNet will automatically provide RSS feed recommendations based on user profile data.
   2. The second process is a dashboard. This dashboard will display useful statistics on DevNet’s user base.
   3. The third process is a You Tube like Media Service called DevTV which allows for uploading video content, managing the content, and viewing the videos within

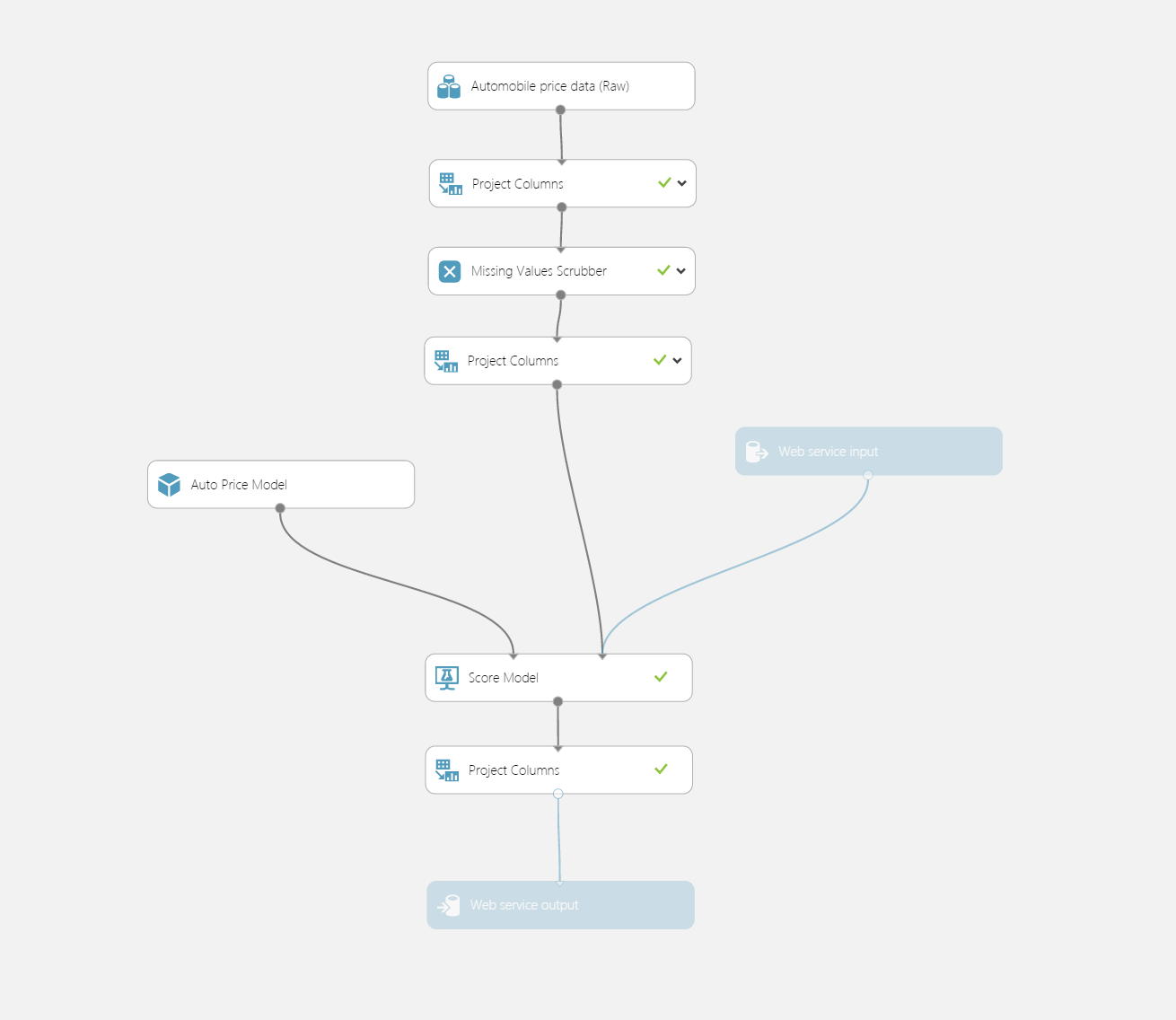
DevNet with a 3D effect.

**II. PROCESS(ES) STEPS**

* 1. **Recommender System**
     1. The Data inputs will be parameters into the web service. These parameters will be based on the user profile. For example user group affiliation, favorite IDE, age, sex, income, etc.
     2. The Output will be the prediction value returned from the web service, in this case a programming language or technology.
     3. The following tasks are performed to go from the input to the output:
        1. First SQL Azure database is used in the cloud on Azure to provide training and test data for model. Access to Database is through the Reader module in Azure ML.

note: Below example is using a built in dataset instead for Automobile data but this is just for demonstration purposes only. Custom data from Azure SQL Database will be used for DevNet application.

* + - 1. Next, using Azure ML(Machine Learning), we create a new workspace and begin creating our model, we select our data source, filter and clean the data, select columns as features, choose algorithm, score the data into training and test data, do some post-filtering, and finally publish model as a web service. Below is a screenshot from a tutorial on Azure ML I did using Automobile dataset in Azure ML:



c. At the application layer, we consume this web service to provide prediction based on user profile data points. Using the Automobile example from before, in Visual Studio we must code the model, the web service call from the controller, and the view.

* + - * 1. Automobile Model



ii. In Controller – Consume Web Service returning deserialized JSON object



iii. View to display prediction results, in this case price of automobile



d. Finally, based on predictive value, in DevNet application we will use Programming language instead of Automobile price. It will be used to recommend a relevant RSS feed to user.

B. **Dashboard**

* + 1. The input data here will be the results returned from a web service from Azure ML containing the relevant statistics.
    2. The output will be some sort of visualization displaying the statistics in DevNet.
    3. The following tasks are performed to go from the input to the output:
       1. First once the predictive analytics model is created, besides the web service described above we will create an additional web service only this time returning various statistics.
       2. This web service will be consumed in DevNet application.
       3. The results will be parsed and displayed in a graphical manner using d3.js.

C. **DevTV**

* + 1. The input data will be registered user uploaded videos.
    2. The output data will be streaming the videos from Azure Media Service into the DevNet application.
    3. The following tasks are performed to go from the input to the output:
       1. First, we will create a new Azure Media Service
       2. Next, we will need the Nuget package WindowsAzure.MediaServices.
       3. Then, we will create in ASP.NET MVC our model and generate the CRUD pages using ASP.NET MVC Scaffolding.
       4. Then we will create the controller with the authorize annotation so that all actions can be executed only when a user is logged in.
       5. We will then proceed to create the view.
       6. We will need to upload media files in chunks since there is a limitation of Request size imposed by IIS. To accomplish this task, we will use the JavaScript file media-upload.js.
       7. Now we need to create a connection string to our Blob Storage where the videos will live. The account name and key are provided when we added our Media Service.
       8. Once the upload is working we will have a save button, which will save the title and id of the video to our database.
       9. Finally, we will stream the videos and display them using 3D effects by making use of three.js framework and css3d.

1. **PROCESS(ES) ENVSISION**
   1. **Recommender System**
      1. This system will kick in after a new user registers to DevNet for the first time.
         1. The web service is automatically triggered upon successful user registration.
         2. The user is then provided with a recommended Programming related RSS feed.
         3. They may decline if they choose, otherwise this feed will be on the site whenever they log in.
      2. Efficiencies of the Recommender System will be convenience to the user and enhanced user experience.
      3. While there aren’t really any tangible benefits in the form of a financial benefit, it can be argued that time is money. Also, the user will benefit from reading technical articles of interest to them without the extra time spent searching the net.
   2. **Dashboard**
      1. The dashboard will be in its own area of the site linked on the navigation menu. The look and feel will be custom in that it will reflect both the theme of the site and display the data in a visually appealing yet artistic way by means of the JavaScript library d3.js, which is commonly used for data visualization.
      2. Having various metrics crucial to the purpose of the site in one place provides quick access for users whether registered to the site or not. The main target user here that may benefit most is the business user. By getting a sense of what languages developers are using as well as interesting statistics pulled from the registered users aids business decisions when it comes to targeting new talent or planning which technologies to use in their own development shops.
      3. Here we have the potential to save businesses money in the following areas:
         1. Hiring costs will lower due to finding talent through DevNet or knowing what to look for in terms of filling their developer positions.
         2. Reduced costs by making better technology forecasts thanks to the provided metrics in the Dashboard. This way businesses won’t risk investing in unpopular technology in their areas which can either make it expensive or time consuming to fill positions. Or even worse, invest in technology that ends up having a short lifetime often due to it no longer being actively developed or supported.
         3. They will also be more informed about the latest changes if they are already committed to being a .NET or Java Shop as an example.
         4. As far as precise dollar amounts saved we can walk through a possible scenario as follows:

“Mr. Smith is the CIO of CompuWorld and is looking to migrate his company’s web applications from Java to .NET. Mr. Smith checks out DevNet’s Dashboard Portal and sees that ASP.NET MVC C# is where it’s at amongst developers. He decides to make this move and hire some new .NET senior programmers to aid in the transition as well as fill some junior roles with developers he met on the site. Mr. Smith knows they will work out great since they posted many knowledge base videos on ASP.NET MVC and so clearly know their stuff. Usually Mr. Smith would have a larger downtime while his existing Java developers retrain but thanks to hiring knowledgeable .NET devs combined with the experience of the existing Java developer’s knowledge of the company’s existing infrastructure, what would have taken 6 months to a year to transition, now only takes 3 to 6 months. This could very easily save the company $500,000 to $1million dollars, even with the added expense of new .NET licensing and additional developers.”

e. Yet another point, the chances of having a bad hire are diminished due to being able to “check out” the devs on DevNet beforehand. This results in more savings and will continue for years to come having such good talent working at CompuWorld.

* 1. **DevTV**
     1. DevTV will be in a 3D environment unlike most 2D galleries. Again, it will echo the established dark theme of the site. The unconventional look and feel will distinguish the site from the rest. The WoW factor will draw in more devs, many of whom grew up playing 3D video games.
     2. DevTV will provide a central place for devs to upload video content providing tutorials and demos. This is in the spirit of open source, where the developers openly share their knowledge free of charge. It also saves time from having to dig through a bunch of irrelevant videos on places like You Tube.
     3. The tangible dollars works both ways here, both for the developer and the future employer. The developer could benefit financially by being noticed through the videos he or she uploads which may result in a job. position or an upgrade from their current job. Employers will obviously benefit since this will save them some of the hassle of lame interviews that hog up time and don’t always result in good talent.