Task A: Big Data – Flights Delays

Introduction

The purpose of this case study is to explore new and improved travel opportunities that allow travel agencies to better predict flight delays based on historical data of such events. Margie's Travel (MT) is a business travel concierge services provider that seeks to differentiate itself and create value for its corporate clients by leveraging its current data assets to deliver new insights that give it a competitive edge. In order to do the aforementioned, Margie's Travel seeks ways to use machine learning and predictive analytics to help consumers make the best travel decisions based on potential delays. This approach is planned to be tested internally by giving customer service representatives access to a prototype web-based tool, allowing the rep to visualise expected delays for a particular customer's preferred departure airport. In addition, current and anticipated weather information is obtained from a third-party provider and also from the United States Department of Transportation (USDOT).

MT plans on implementing this pilot project to investigate whether historical weather data and flight delay data can be fused together to build a solution wherein it can be helpful in creating a risk assessment for optimising travel for their customers. The idea is to test out an application that their agents can utilise as a means to help with the customer's decision-making process when it comes to risk assessment while travelling. This solution aims to be provided to Margie's Travel's premium tier business travellers, giving them an opportunity to assess the flight delay risk when booking their travels.

To accomplish building a prototype solution, Microsoft Azure tools were used for the various tasks, namely, Azure Databricks, Data Factory, Storage Containers (Spark container) and Power BI. The latter was used to provide a visualisation of the solution.

Azure Databricks is an Apache Spark based platform for predictive analytics. The purpose of this tool is to provide a notebook like environment for users to build and deploy machine learning models for better scalability. Data sources such as Azure Blob Storage, SQL Database etc are also supported. Databricks provides a Spark cluster that is auto-scalable and can streamline pipelines to ensure ease of use and makes managing the load of the machine learning model. Streamlined pipelines make integration with other tools easy for simplifying feature selection, data preprocessing and training of the model.

Azure Data Factory is useful for managing data pipelines for transforming data across various cloud services and data stores, i.e., data pipelines are created that can move data from on-premise sources to cloud sources. Pipeline runs can be triggered on a schedule, as well for ease of use. Integration between other services such as Databricks is also provided to combine building models and providing analytics solutions. Custom transformations are also accomplished in Data Factory with the use of advanced tools. Data transformation such as sorting, filtering et al is made easy.

Power BI is a data visualisation tool to gain insights from the input data. It can also integrate with other Azure tools such as Databricks for dashboard creation. Power BI has the option to also connect with other Microsoft tools such as Excel, Azure SQL Databases etc as data sources for gleaning useful information. Like the tools above, Power BI also provides a platform for data transformation and preprocessing for visual analysis of the data. It is a valuable tool for exploratory data analysis and allows users the opportunity to collaborate which is helpful in organisations.

Solution Implementation

Creating the Databricks cluster:

Databricks clusters are used to run various workflows, being used as automated jobs. The purpose of creating this cluster is to include various configurations and resources for data analytics and machine learning jobs. The creation of a cluster is done by setting up the access key and storage account details within the Spark config input area. This gives the user access to the storage account and the required files for uploading.

Load sample data:

After creating the cluster, the next step is to upload the sample data from the Data Explorer tab. The data are CSV files needing formatting to be accessed by the databricks cluster and hence the names of each file are formatted to align with the formatting rules. Once the data has been loaded, related machine learning libraries are imported to enable the cluster to train, deploy and evaluate the machine learning model.

The first notebook relates to data preprocessing and transformations, while the second is to train the machine learning model. The algorithm chosen for this particular case study is Decision Tree Classifier. There are multiple classification algorithms that can be used to achieve the best possible model. The third notebook is used to deploy the model to production after serving it.

Setup the Data Factory and Data Factory Pipeline:

Azure Data Factory is a data integration service that is cloud-based for pipeline management, including creating and scheduling them for transforming and transferring data across the cloud and other data stores. Data integration through data factory is highly scalable and reliable. Runtime Integration is a tool provided by data factory for data integration in various network environments, including self-hosted nodes (used for this case study). The chosen self-hosted node performs data transfer and enables data flow across the various Azure services (including Databricks), used to send operations to external resources.

During the creation of the environment, an authentication key is required to register the Runtime Integration and that is taken from the data factory's newly created node. This is helpful for configuring the former with the data factory. After the connection has been established, the pipeline can be used to move data from the on-premise server to the Blob Storage that was created during the pipeline creation.

To achieve the above, a copy pipeline has to be created and the connection between the on-prem server and the blob storage is tested to make sure they run successfully. The file system and the blob storage both use the local files to get the data for the copy pipeline. Once the necessary credentials and connections have been established, the pipeline is deployed for the next step, which is to perform the scoring of the notebook that was previously trained.

Operationalise ML Scoring:

To operationalise ML scoring, a connection between the data factory pipeline and the databricks notebook has to be established. Essentially, to operationalise ML scoring is to implement and integrate a previously trained machine learning model in production. This is done for solving an ML problem such as prediction or classification on any new data. The copy pipeline contains a section known as Activities which can connect to Databricks notebooks. This is run on Spark jobs present in the databricks cluster in data factory

pipelines. A copy activity is already present for the copy pipeline and all that's needed is to configure the notebook activity to connect to the ML scoring notebook from databricks. Once that has been established and configured, the two activities will be available at pipeline runtime. The structure is as follows – the copy activity first processes and creates the copied file in the storage account and the notebook activity connects the cluster and the ML notebook. After publishing the setup to the server, the workflow can be manually triggered to perform ML scoring. The user must make sure that the cluster is running otherwise the pipeline run will fail.

Summarise the Data:

Summarising the data is required to be able to perform data visualisations on it. The scored data that is generated from the pipeline is then summarised as one table with a target variable that aggregates the anticipated delays for each origin airport, along with adding the coordinates to it.

Visualisation of Data:

Power BI is a powerful tool for data exploration and data visualisations. In order to get the right data, the server to be accessed in Power BI has to be integrated with the cluster that was created and a connection has to be established using a JDBC URL. Built-in visualisations such as maps, charts etc can be utilised to gain insights from the data. After summarising the data, it can be seen that the flight delays are over a period of one month. To get a complete synopsis of the data, visualisations such as treemaps, maps and stacked bar are used. Hierarchical data is visualised well using treemaps which gives a nested look for ease of readability. From the visuals, it can be seen that Atlanta's airport has the highest number of flight delays and the map visual gives the user an accurate bird's eye view of the coordinates. Utilising bubble size sorts the data and tells the user which airports have the most number of delays.

Evaluation

The outcome of the project is to predict flight delays for each origin airport. To achieve this, binary classification is done and a classification algorithm chosen for the purpose. Classification is used to predict class labels for datasets without labels. Algorithms such as Decision Trees, Naïve Bayes etc are popular classification algorithms.

Decision Trees are commonly used for machine learning model building and training because of the ease of understanding and training. They are also a good choice for binary classification because they are easy to interpret (Patel et al, 2018). For the case study given here, decision trees are useful in selecting the most important features for causes of flight delays such as weather, departure time etc. This knowledge helps the travel company to mitigate flight delays by giving a sliding window for travel booking for the airports with most number of delays. When it comes to handling non-linear and complex relationships, decision trees are adept at it. The relationship between flight delays and weather condition is one example. Being an imbalanced dataset, it is a good idea to use decision trees to prioritise the minority class. This model splits the data into subsets until a stopping criterion is reached. If the company uses this model, it can be used to guide customers to make alternate travel arrangements in the case of significant delays.

Another machine learning model that is viable for binary classification of flight delays is gradient boosting. It is an ensemble learning model that is used to combine multiple weak learning models (decision trees etc) which boosts accuracy and provides robustness of the predictions. It has similar

characteristics to decision trees, but having multiple models helps it choose the best and increase the accuracy of determining flight delays.

Reference

Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. *International Journal of Computer Sciences and Engineering*, *6*(10), 74-78.

Appendix

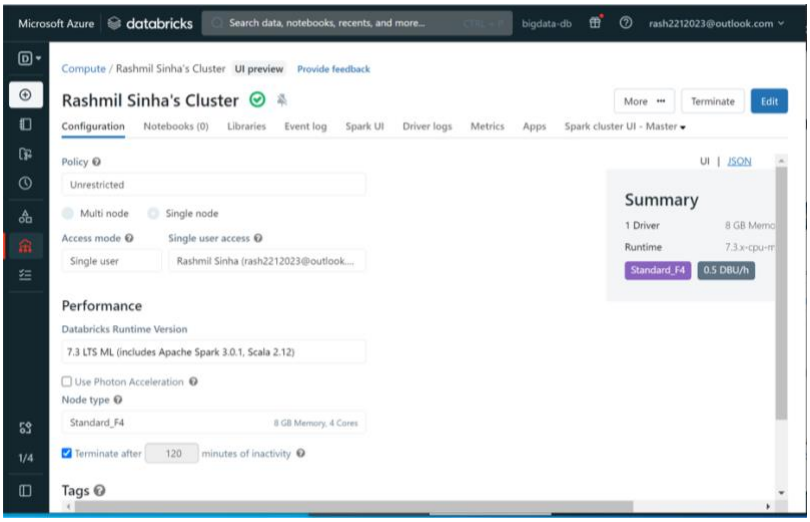

Figure 1 - Cluster in Databricks

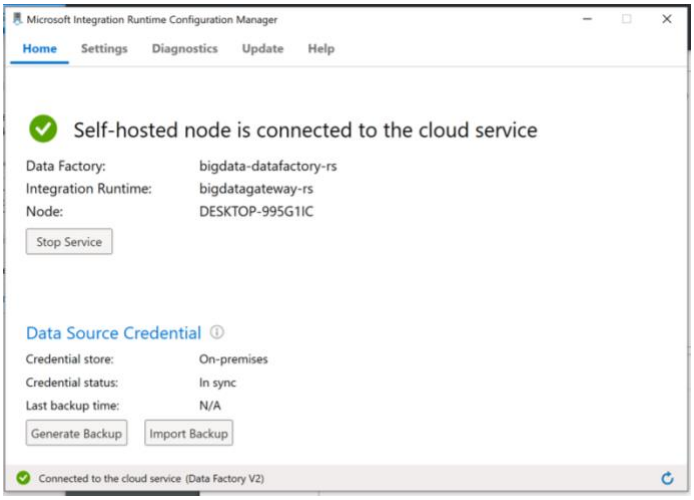

Figure 2 - Runtime Integration successful

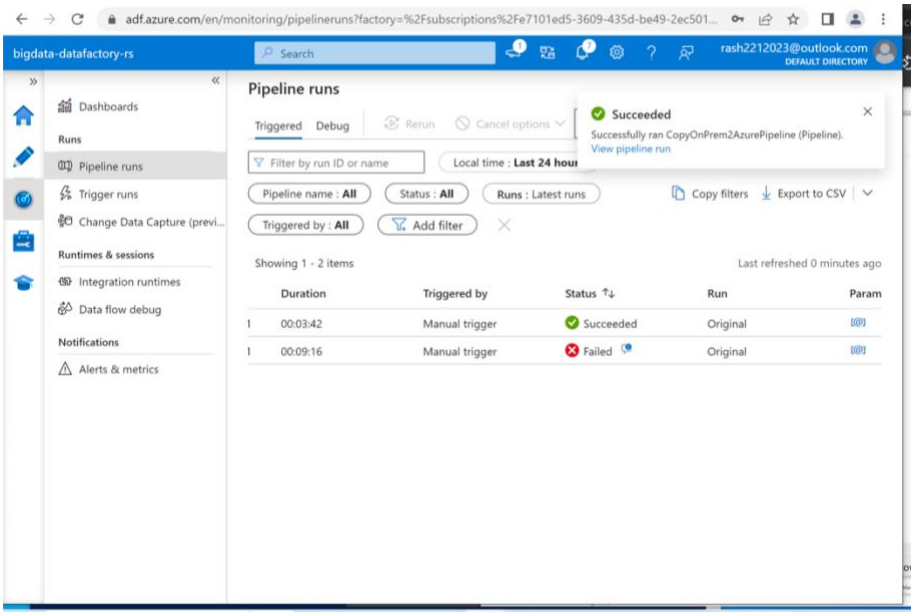

Figure 3 - Successful Copy Pipeline run

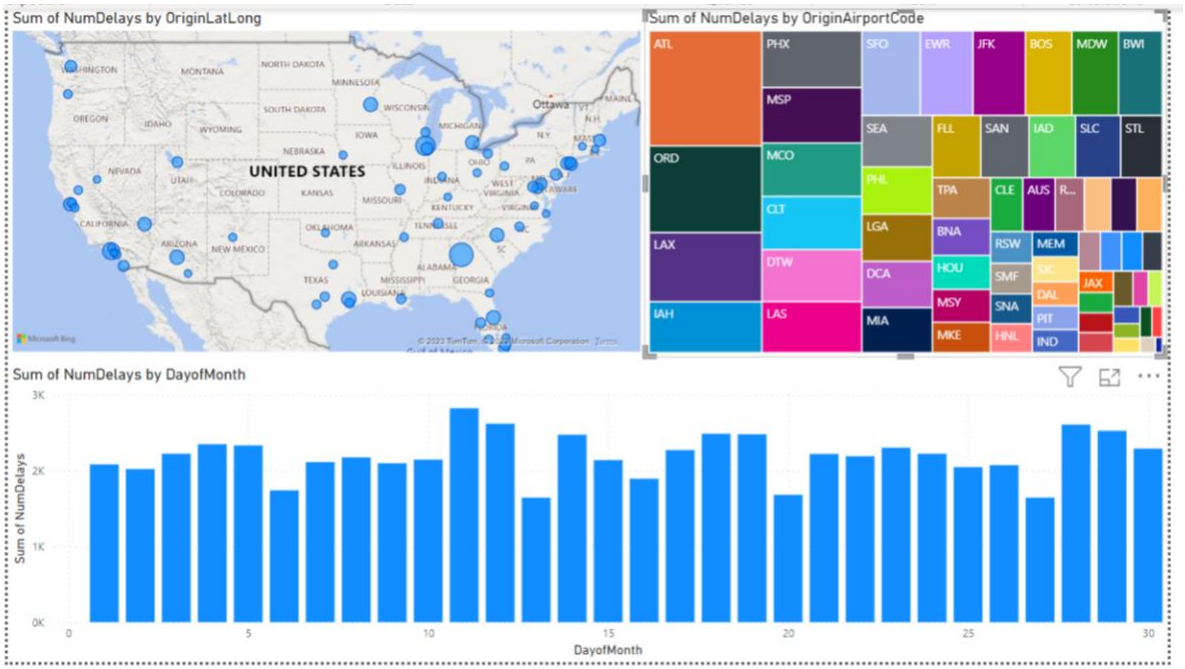

Figure 4 - Power BI visualisation