RealEstate-Data-Engineering-Project-Using-Azure-DataFactory-Databricks

J Jatin DE120

Problem Statement:

Develop a pipeline that utilizes Azure Databricks for building and deploying machine learning models, and Azure Data Factory for orchestrating the entire process, including data preparation, model training, and deployment.

Project Overview:

This project leverages the "Real Estate Sales 2001-2020 State of Connecticut" dataset from Kaggle to develop an end-to-end machine learning pipeline for predicting property types as residential, commercial, or vacant land. Azure Databricks is utilized for data preprocessing, feature engineering, and training machine learning models on historical real estate sales data. Azure Data Factory orchestrates the workflow, automating data ingestion, transformation, model training, and deployment processes. By analyzing features such as sale prices, property locations, and transaction details, the pipeline predicts property types with high accuracy. This solution demonstrates the integration of Azure tools for scalable and automated real estate analytics.

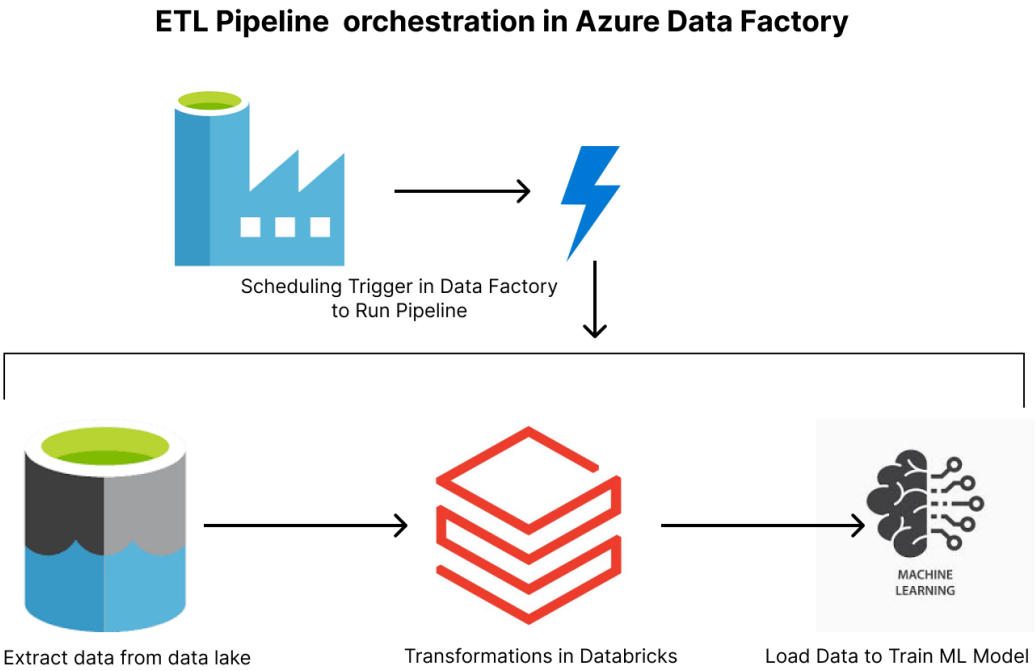
Real Estate Sales 2001-2020 State of Connecticut Dataset Overview:

This extensive dataset provides a comprehensive historical perspective on real estate sales in Connecticut, covering a span of two decades from 2001 to 2020. The dataset offers a rich repository of information about real estate transactions, including property attributes, financial details, and geographic locations. This extensive historical data is invaluable for gaining insights into long-term trends, assessing property values, and conducting in-depth market research.

Dataset Description:

1. Serial Number:
   A unique identifier for each transaction, facilitating easy tracking and reference throughout the entire 20-year.
2. List Year:
   Specifies the year to which each transaction is associated, enabling historical analysis and trend identification.
3. Date Recorded:
   The date when each transaction was officially recorded, providing a chronological view of real estate sales activity over the two decades.
4. Town:
   Identifies the specific town within Connecticut where each property is located, allowing for regional and temporal analysis.
5. Address:
   The property address for each transaction, offering a detailed inventory of properties sold.
6. Addressed Value:
   The assessed value of each property, a critical component for calculating property taxes.
7. Sale Amount:
   The actual sale price or transaction amount, serving as a primary indicator of property market value.
8. Sale Ratio:
   The ratio of the sale amount to the assessed value, useful for assessing the fairness and accuracy of property assessments over time.
9. Property Type:
   Categorizes properties into types such as Residential, Commercial, Industrial, or others, offering insights into the evolving landscape of Connecticut's real estate market.
10. Residential Type:
    Provides further classification for residential properties, including Single Family, Condo, or Two Family, reflecting the changing nature of housing trends.

Architecture Diagram



Execution Overview

This project focuses on building a machine learning pipeline that integrates **Azure Databricks** for data preparation, feature engineering, and model training, and **Azure Data Factory** to automate and orchestrate the workflow. The processed data and trained model are stored in Azure DataLake Storage Gen2, and the model is deployed in Azure DataBricks.

1. Data Preparation

1.1. Data Ingestion:

- The Dataset is downloaded from Kaggle and the raw data is stored in Azure Data Lake Storage Gen2 storage container(Bronze Layer).
- The data is ingested into a dataframe by mounting ADLS to Databricks,enabling seamless access to the container. This data is then written into Delta Tables and ADF moves this standardized data to the Silver layer.

1.2. Data Cleaning:

- Since, Raw data often contains inconsistencies, missing values, or irrelevant information, the Data in Silver Zone is subjected to necessary transformations to ensure the data is accurate, reliable, and optimized for ML model deployment.
- This Transformed Data is uploaded to Gold Layer by ADF

2. Model Deployment:

- The Data in Gold Layer is trained using Random Forest Algorithm and ADF orchestrates the model deployment subsequently.

Extraction-Transformation-Load (ETL) Pipeline:

ETL flow includes data ingestion, transformation, and integration with a Machine Learning (ML) model for predictive analysis.

1. Ingestion: From Bronze Zone to Silver Zone

- **Data Extraction**:
  - Raw data is ingested from ADLS into the **Bronze Zone** by **Mounting**. This raw data may be unstructured or semi-structured, and it is typically stored in formats such as CSV.
  - Use Databricks notebooks to perform exploratory data analysis (EDA)
- **Data Movement**:
  - The raw data from the Bronze Zone is moved to the **Silver Zone** by writing them into **Delta Tables**, where it will undergo transformations.

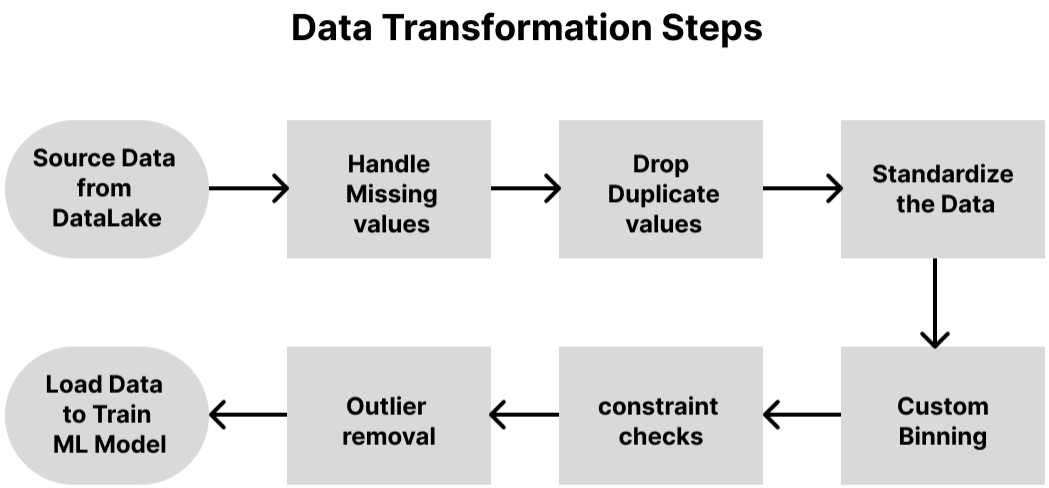
2. Transformation: From Silver Zone to Gold Zone

- **Data Cleaning and Preprocessing:**

  The Silver Zone serves as the stage where data undergoes cleaning and transformation. The steps include:

  - Removing duplicates and irrelevant records.
  - Handling missing values (imputation or deletion).
  - Standardizing Date Formats

- Filter out rows with negative Assessed value or Sale Amount
- Finding out Outliers in Data



- **Feature Engineering:**
    - Statistical Aggregations like mean, standard Deviation are applied.
    - Filtering out the outliers which can skew statistical analyses and machine learning models.
- Standardization:
    - Data is standardized into a format that is ready for the ML model, such as Delta tables or Silver Zone.

3. Loading Data to ML Model

After the data is cleaned and transformed into the Gold zone, it is used for training or inference with a machine learning model. Here are the steps involved:

- **Training Data for ML**:
    - The data from the **Gold Zone** (which could include structured features) is loaded into the machine learning pipeline.

- The ML model is trained to classify the Property Type with Random Forest Algorithm using tools like **Azure Databricks**.

4. Model Deployment

- Once the model is trained and validated, it needs to be deployed for production use.
- Batch processing is used to process data in bulk and run predictive Analysis.
- The performance of the deployed ML model is analysed using Accuracy

Azure Resources Used for this Project:

- Azure Data Lake Storage Gen2
- Azure Data Factory
- Azure Databricks

Project Requirements

1. Data Ingestion Requirements

- Download the Dataset from Kaggle
- Ingest it into Azure Data Lake Gen2
- Configure access to the storage account using Azure Storage account key
- Mount the Storage Container into local mount point
- Data can be read from the mounting point.

2. Data Transformation Requirements

- Filter irrelevant or incomplete records
- Handle missing values and correct data types
- Perform deduplication and standardize Date formats
- Add new columns based on existing ones for making meaningful insights
- Filter out outlier methods
- Perform statistical aggregations such as mean, standard deviation
- Store the data in some local mounting point

3. ML model Training and Deployment requirements

- Load Data into Pandas Dataframe for pre-processing.
- Split the dataset into training and testing subsets
- Apply random forest algorithm to run predictive analysis
- Evaluate the model using metrics like accuracy
- Save the trained model to a storage location

4.Scheduling Requirements.

- Use Azure Data Factory (ADF) to automate data ingestion and transformation.
- Schedule to run pipeline based on requirement
- Ability to rerun failed pipelines
- Ability to monitor pipelines

5.Data Analysis Requirements

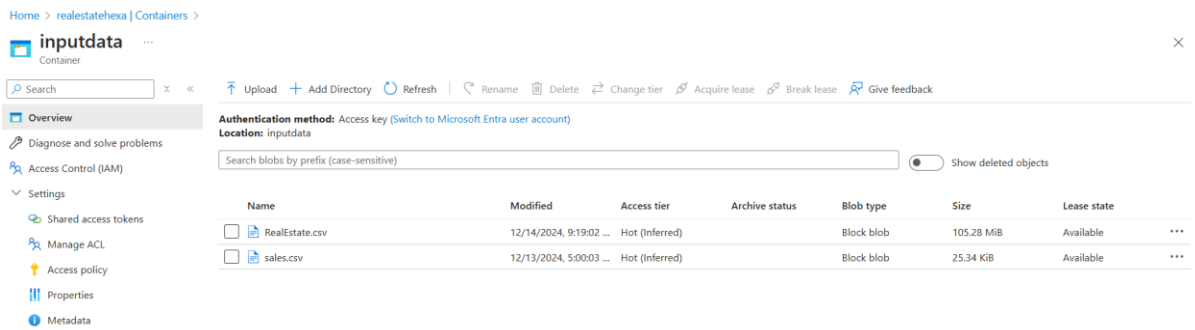
- Ability to visualize the output of notebooks.
- Ability to view the job runs both in ADF and Azure Databricks job runs

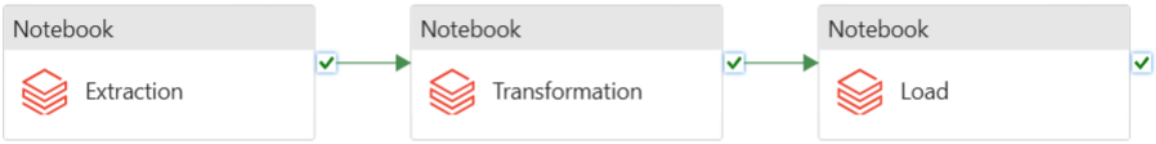
6. Additional Requirements

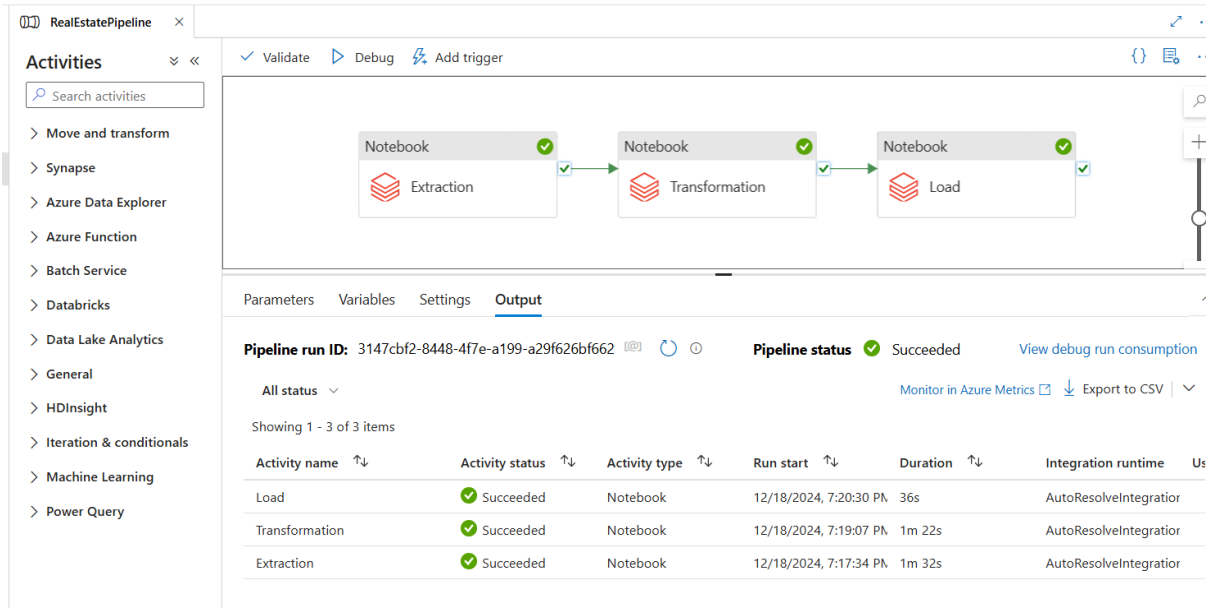
- Perform Exploratory Data Analysis to uncover patterns, detect anomalies in the dataset.

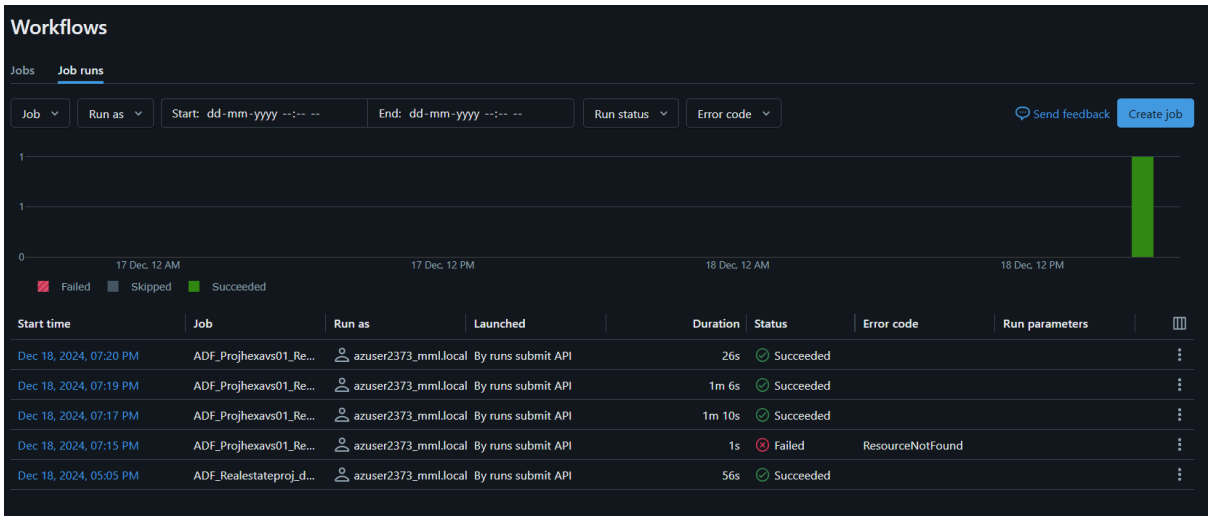
Results & Analysis:

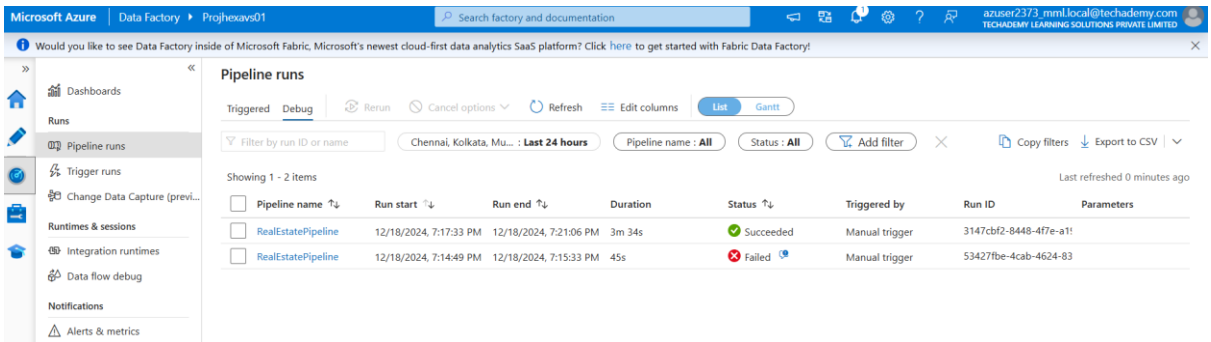
Pipeline Orchestration Results




Notebook
Extraction
Notebook
Transformation
Notebook
Load


RealEstatePipeline
Activities
Search activities
Move and transform
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
HDInsight
Iteration & conditionals
Machine Learning
Power Query
Validate
Debug
Add trigger
Notebook
Extraction
Notebook
Transformation
Notebook
Load
Parameters
Variables
Settings
Output
Pipeline run ID: 3147cbf2-8448-4f7e-a199-a29f626bf662
Pipeline status Succeeded
View debug run consumption
All status
Monitor in Azure Metrics
Export to CSV
Showing 1 - 3 of 3 items
Activity name
Activity status
Activity type
Run start
Duration
Integration runtime
Load
Succeeded
Notebook
12/18/2024, 7:20:30 PM
36s
AutoResolveIntegration
Transformation
Succeeded
Notebook
12/18/2024, 7:19:07 PM
1m 22s
AutoResolveIntegration
Extraction
Succeeded
Notebook
12/18/2024, 7:17:34 PM
1m 32s
AutoResolveIntegration


Workflows
Jobs
Job runs
Job
Run as
Start: dd-mm-yyyy --:-- --
End: dd-mm-yyyy --:-- --
Run status
Error code
Send feedback
Create job
17 Dec, 12 AM
17 Dec, 12 PM
18 Dec, 12 AM
18 Dec, 12 PM
Failed
Skipped
Succeeded
Start time
Job
Run as
Launched
Duration
Status
Error code
Run parameters
Dec 18, 2024, 07:20 PM
ADF_Projhexavs01_Re...
azuser2373_mml.local
By runs submit API
26s
Succeeded
Dec 18, 2024, 07:19 PM
ADF_Projhexavs01_Re...
azuser2373_mml.local
By runs submit API
1m 6s
Succeeded
Dec 18, 2024, 07:17 PM
ADF_Projhexavs01_Re...
azuser2373_mml.local
By runs submit API
1m 10s
Succeeded
Dec 18, 2024, 07:15 PM
ADF_Projhexavs01_Re...
azuser2373_mml.local
By runs submit API
1s
Failed
ResourceNotFound
Dec 18, 2024, 05:05 PM
ADF_Realestateproj_d...
azuser2373_mml.local
By runs submit API
56s
Succeeded

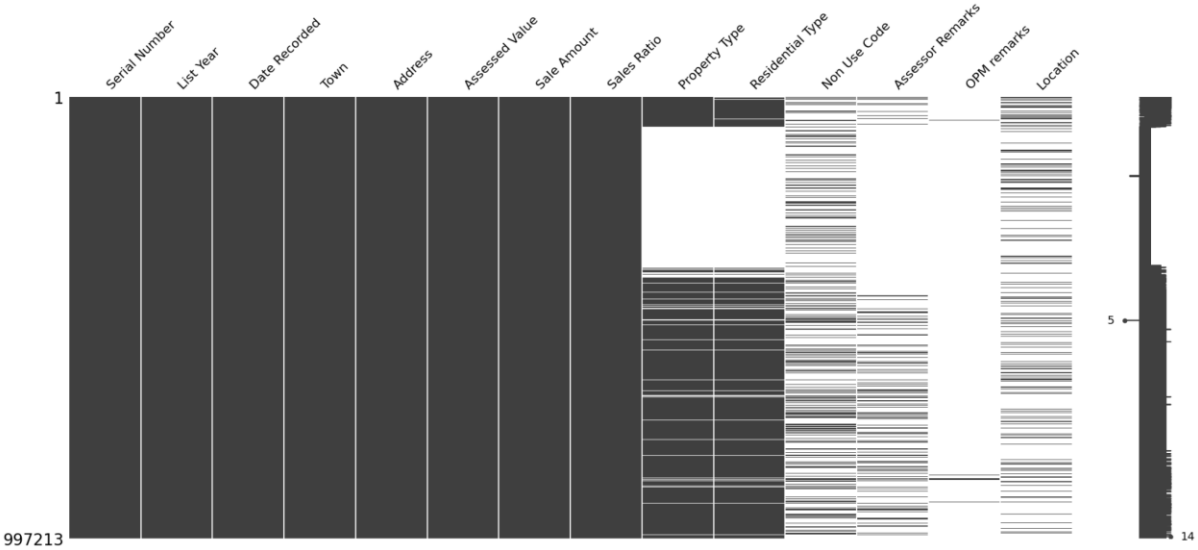

Microsoft Azure
Data Factory
Projhexavs01
Search factory and documentation
azuser2373_mml.local@techademy.com
Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform? Click here to get started with Fabric Data Factory!
Dashboards
Runs
Pipeline runs
Trigger runs
Change Data Capture (previ...
Runtimes & sessions
Integration runtimes
Data flow debug
Notifications
Alerts & metrics
Pipeline runs
Triggered
Debug
Rerun
Cancel options
Refresh
Edit columns
List
Gantt
Filter by run ID or name
Chennai, Kolkata, Mu... : Last 24 hours
Pipeline name : All
Status : All
Add filter
Copy filters
Export to CSV
Showing 1 - 2 items
Last refreshed 0 minutes ago
Pipeline name
Run start
Run end
Duration
Status
Triggered by
Run ID
Parameters
RealEstatePipeline
12/18/2024, 7:17:33 PM
12/18/2024, 7:21:06 PM
3m 34s
Succeeded
Manual trigger
3147cbf2-8448-4f7e-a1!
RealEstatePipeline
12/18/2024, 7:14:49 PM
12/18/2024, 7:15:33 PM
45s
Failed
Manual trigger
53427fbe-4cab-4624-83

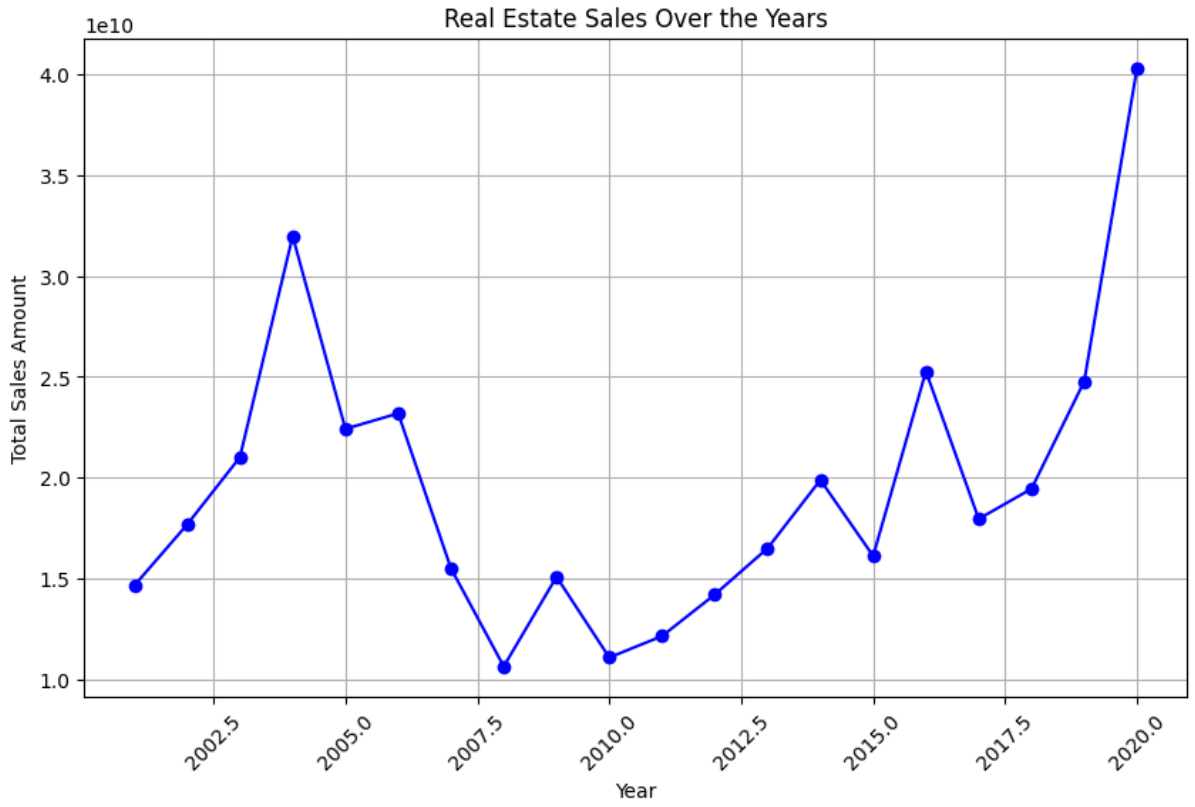
ML model Deployment Results

```
predictions: pyspark.sql.dataframe.DataFrame
Test Accuracy: 0.6882716049382716
```

```
Model loaded successfully!
+-------------+----------+--------------------+
|Property_Type|prediction|            features|
+-------------+----------+--------------------+
|Single Family|       0.0|[2013.0,170890.0,...|
|Single Family|       0.0|[2017.0,35840.0,6...|
|Single Family|       1.0|[2017.0,46725.0,9...|
|Single Family|       1.0|[2017.0,81970.0,9...|
|        Condo|       0.0|[2017.0,101170.0,...|
|Single Family|       0.0|[2017.0,117150.0,...|
|  Four Family|       0.0|[2017.0,161490.0,...|
|Single Family|       0.0|[2017.0,116935.0,...|
|Single Family|       0.0|[2017.0,52360.0,4...|
|        Condo|       0.0|[2017.0,17220.0,2...|
|Single Family|       0.0|[2017.0,23870.0,3...|
|   Two Family|       1.0|[2017.0,64610.0,1...|
|        Condo|       0.0|[2017.0,75990.0,7...|
| Three Family|       1.0|[2017.0,45570.0,1...|
|Single Family|       0.0|[2018.0,161840.0,...|
|Single Family|       0.0|[2018.0,14000.0,3...|
|Single Family|       0.0|[2019.0,96630.0,2...|
```

EDA Analysis Results





Tasks performed:

- Built a solution architecture for a data engineering solution using Azure Databricks, Azure Data Lake Gen2 and Azure Data Factory.
- Created and used Azure Databricks service and the architecture of Databricks within Azure.
- Worked with Databricks notebooks and used Databricks utilities, magic commands, etc.
- Passed data between notebooks as well as created notebook workflows.
- Created, configured, and monitored Databricks clusters, cluster pools, and jobs.
- Mounted Azure Storage in Databricks using secret keys.
- Worked with Databricks Tables, Databricks File System (DBFS), etc.
- Used Delta Lake to implement a solution using Lakehouse architecture.
- Developed a Machine Learning (ML) model - RandomForest classification.
- Loaded data into ML model to train by using that data
- Evaluated Model Performance by evaluation metrics like accuracy, precision.

Spark (Only PySpark and SQL)

- Spark architecture, Data Sources API, and Dataframe API.
- PySpark - Ingested CSV into the data lake as parquet files/ tables.
- PySpark - Transformations such as Filter, Simple Aggregations, GroupBy, Window functions etc.
- PySpark - Created global and temporary views.
- Spark SQL - Created databases, tables, and views.
- Spark SQL - Transformations such as Filter, Join, Simple Aggregations, GroupBy, etc.
- Spark SQL - Created local and temporary views.
- Implemented full refresh and incremental load patterns using partitions.

Delta Lake

- Performed Read, Write, Update, Delete, and Merge to delta lake using both PySpark as well as SQL.
- History, Time Travel, and Vacuum.

Azure Data Factory

- Created pipelines to execute Databricks notebooks.
- Designed robust pipelines to deal with unexpected scenarios such as missing files.
- Created dependencies between activities as well as pipelines.
- Scheduled the pipelines using data factory triggers to execute at regular intervals.
- Monitored the triggers/ pipelines to check for errors/ outputs.

About the Project:

Folders:

- Pynb - folder contains extraction, transformation, load, ML model training and exploratory data analysis notebooks in ipynb format.

- Html - folder contains extraction, transformation, load, ML model training and exploratory data analysis notebooks in html format

Technologies/Tools Used:

- Pyspark
- Spark SQL
- Delta Lake
- Azure Databricks
- Azure Data Factory
- Azure Data Lake Storage Gen2
- Feature Engineering
- Machine Learning

Report Prepared by,

DE115 - Divya Sree Murali

DE120 - Jatin J

DE138 - Sivaprakash V