AUTO AI MODEL BUILDING IN IBM WATSON STUDIO

IBM-PBEL INTERNSHIP – PROJECT DOCUMENT

SUBMITTED BY – RISHIKA JALAN

BATCH – 2(ARTIFICIAL INTELLIGENCE)

COLLEGE – MODY UNIVERSITY SCIENCE AND TECHNOLOGY

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to IBM for providing me the opportunity to pursue my internship in the technology **Artificial Intelligence**. It has been an invaluable learning experience that has contributed immensely to my professional and personal development.

I am extremely grateful to my Mentor, **Mr. Samarth Amrute** for his invaluable guidance, encouragement and support during my time here. His insights and advice have pushed me to grow and learn in ways I didn’t think possible. I could not have asked for a better mentor. I would also like to thank the batch mates. Thank you for making me feel so welcomed and for patiently answering all my questions. Your passion and dedication to your work is truly inspiring.

Finally, I want to thank my professors at my college **Mody University of Science and Technology** for laying the foundation that allowed me to maximize this internship opportunity. This internship has been a defining period in my life and I could not have done it without all of your help. Thank you.

INDEX

INTRODUCTION

In this project, I used **IBM Watson Studio's AutoAI** to predict product sales across different Big Mart outlets. AutoAI automates the end-to-end machine learning process, including data preprocessing, feature engineering, model selection, and hyperparameter optimization — making it faster and easier to build high-quality predictive models.

The Big Mart Sales Prediction dataset includes historical sales data along with product details like item type, weight, price, and outlet characteristics such as size, location, and establishment year. By training an AutoAI model on this dataset, the goal is to accurately forecast sales, helping Big Mart make data-driven decisions to improve inventory management and marketing strategies.

This project showcases how automated machine learning can turn complex retail data into actionable insights, saving time and enhancing forecasting accuracy.

TOOLS USED

Services:

- IBM Watsonx.AI Studio
- IBM Watsonx.AI Runtime Service
- IBM Cloud Object Storage
- IBM Watsonx.AI AutoAI
- IBM Watsonx.AI Data Refinery

External Dataset:

Big Mart Sales Prediction

Source :

https://www.kaggle.com/datasets/shivan118/big-mart-sales-prediction-datasets

DATASET DESCRIPTION

1. Item_Identifier

- **Type:** Categorical (string)
- **Description:**
  This column contains a unique alphanumeric code assigned to each product (e.g., FDA15, DRC01). It acts as a primary identifier for items and is used to differentiate products, even those belonging to the same broader category. While the code itself does not directly carry numerical meaning, it can be helpful when merging data or creating derived features.

---

2. Item_Weight

- **Type:** Numerical (float)
- **Description:**
  Represents the weight of each product in kilograms. This attribute provides a physical characteristic of the product, which may influence logistics, pricing, and sales patterns. Some products have missing values in this column, which might need to be addressed through data cleaning (e.g., imputing missing weights using median or mean values by item type).

---

3. Item_Fat_Content

- **Type:** Categorical (string)
- **Description:**
  Indicates the fat content category of the product, such as "Low Fat" and "Regular." However, the data includes inconsistent labels like "low fat", "LF", and "reg" that refer to the same category. Proper preprocessing requires standardizing these labels to ensure meaningful analysis. Fat content can influence consumer preference and sales, especially for food and beverage items.

---

4. Item_Visibility

- **Type:** Numerical (float)
- **Description:**
  Represents the proportion of the total display area in the store that is allocated to a particular product. This is expressed as a percentage (typically as a decimal, e.g., 0.016). Higher visibility can positively impact product sales, as products displayed more prominently are likely to catch shoppers' attention. Extremely low or zero values might indicate missing or incorrect data.

---

5. Item_Type

- **Type:** Categorical (string)
- **Description:**
  Broad category or classification of products, such as Dairy, Soft Drinks, Meat, Fruits and Vegetables, Household, Baking Goods, etc. This column helps in grouping similar items and is useful for feature engineering. Typically, there are around 16–17 unique categories, offering a balanced view of the product range.

---

6. Item_MRP

- **Type:** Numerical (float)
- **Description:**
  Stands for Maximum Retail Price, which is the highest price at which the product is sold to consumers. This attribute is crucial, as pricing strongly affects consumer purchasing behavior and, consequently, sales volume. The distribution of MRP can reveal product pricing strategies and segmentation.

---

7. Outlet_Identifier

- **Type:** Categorical (string)
- **Description:**
  Unique alphanumeric code for each retail outlet (e.g., OUT049, OUT018).

This serves to identify data coming from specific stores and is particularly important when analyzing outlet-level trends or building models that consider outlet characteristics.

---

8. Outlet_Establishment_Year

- **Type:** Numerical (integer)
- **Description:**
  The year in which the outlet was opened (e.g., 1999, 2009). Although this field is numeric, it often makes sense to derive a new feature from it, such as the **age of the outlet** (current year minus establishment year), which may provide more intuitive insights about how long the outlet has been operating and its potential effect on customer loyalty and sales.

---

9. Outlet_Size

- **Type:** Categorical (string)
- **Description:**
  Describes the physical size of the store, with categories like "Small", "Medium", and "High". Store size might reflect capacity, product variety, and expected customer traffic. Note that this column has missing values, which should be addressed through imputation or other data cleaning methods to avoid bias in analysis.

---

10. Outlet_Location_Type

- **Type:** Categorical (string)
- **Description:**
  Indicates the type of city where the outlet is located: "Tier 1", "Tier 2", or "Tier 3". Typically, Tier 1 cities are larger metropolitan areas, while Tier 3 cities are smaller towns. This column can capture differences in customer behavior, purchasing power, and competition across different urban settings.

---

11. Outlet_Type

- **Type:** Categorical (string)
- **Description:**
  Describes the format of the retail outlet, such as "Supermarket Type1", "Supermarket Type2", "Supermarket Type3", or "Grocery Store". These categories represent differences in business models, product variety, and scale of operations, which can significantly impact sales performance.

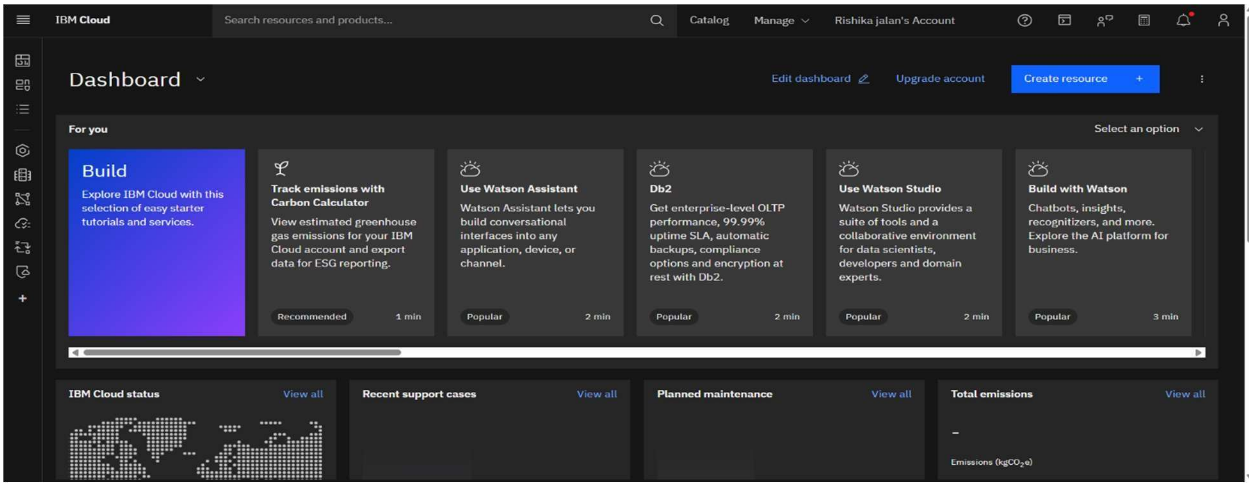
---

12. Item_Outlet_Sales

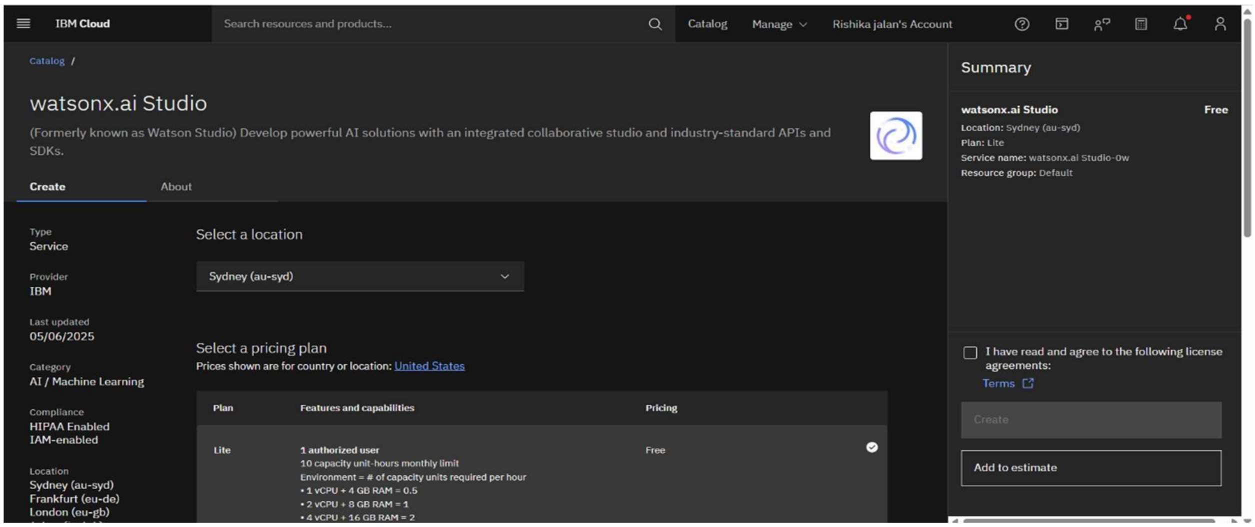
- **Type:** Numerical (float)
- **Description:**
  The target variable of the dataset, representing the total sales of each product in a particular outlet over a specified period (often monthly or quarterly). This column is measured in monetary units and is used as the dependent variable for regression models aiming to predict future sales.

SETUP PROCESS

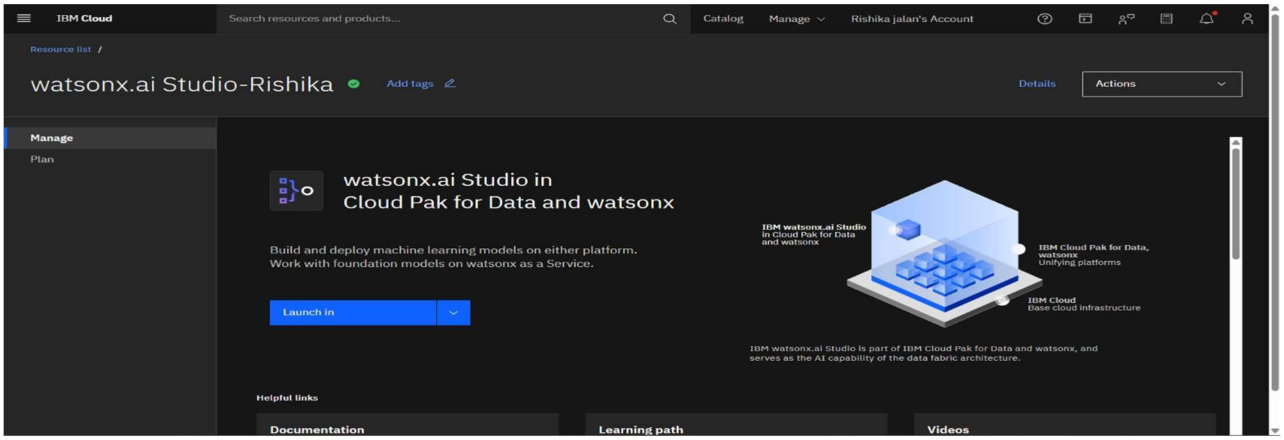
STEP 1: Login to IBM Cloud account



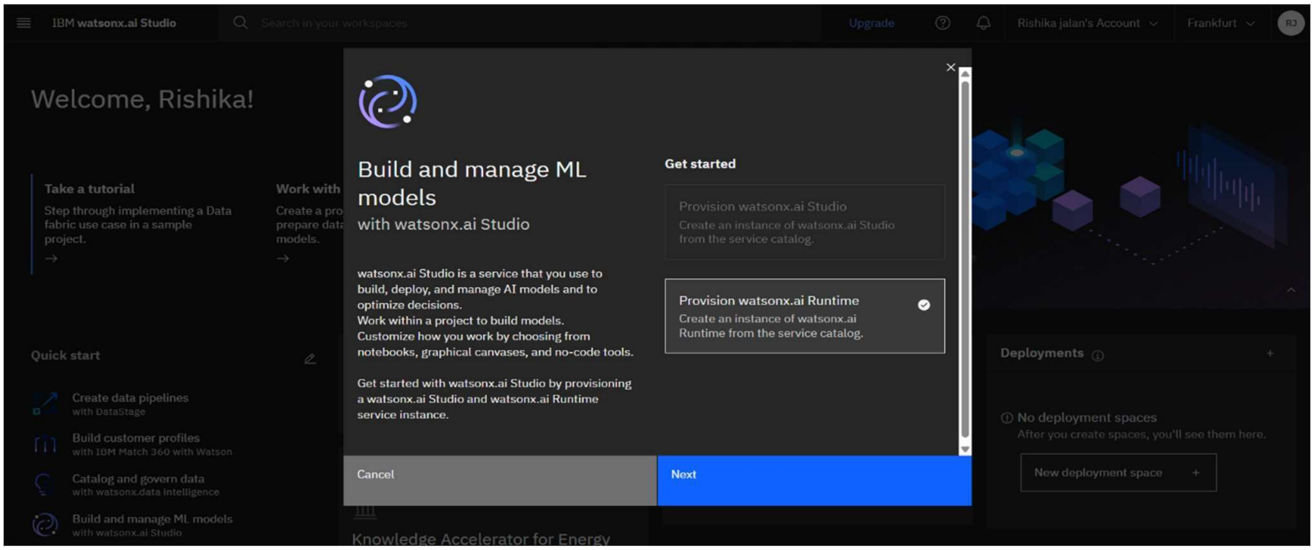
STEP 2: Creating Watsonx.ai Studio Service Instance



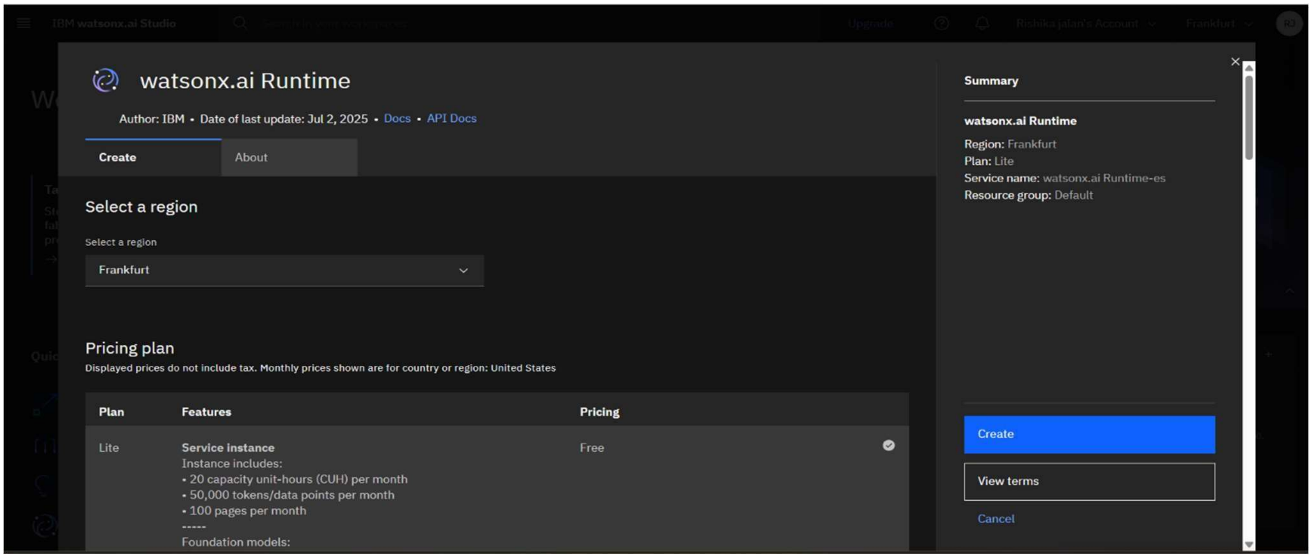
STEP 3: Launching Watsonx.ai Studio – Cloud Pak for Data and Watson



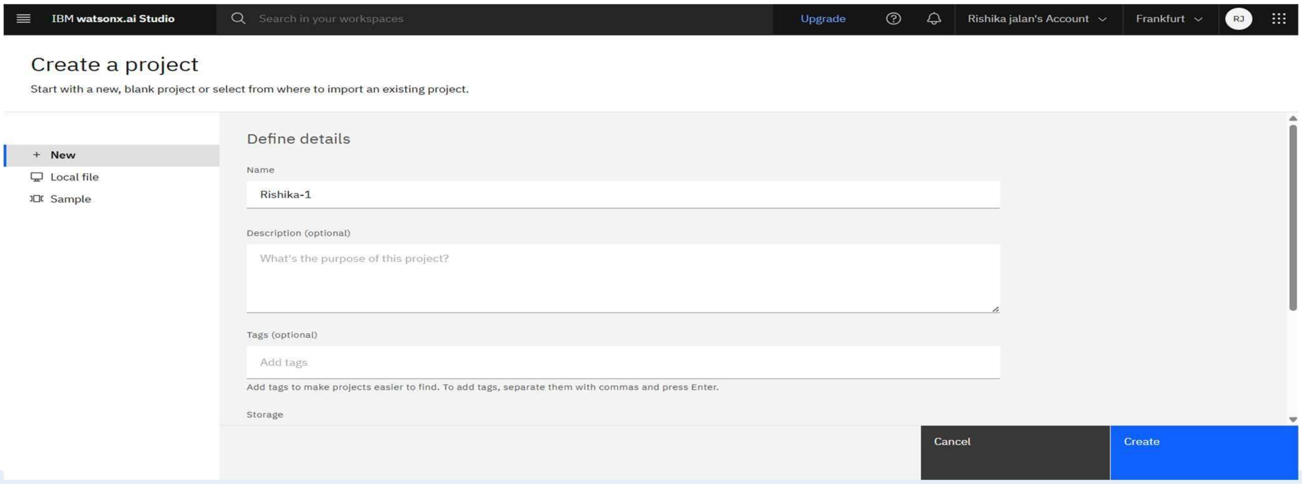
STEP 4: Watsonx.ai Studio Launched



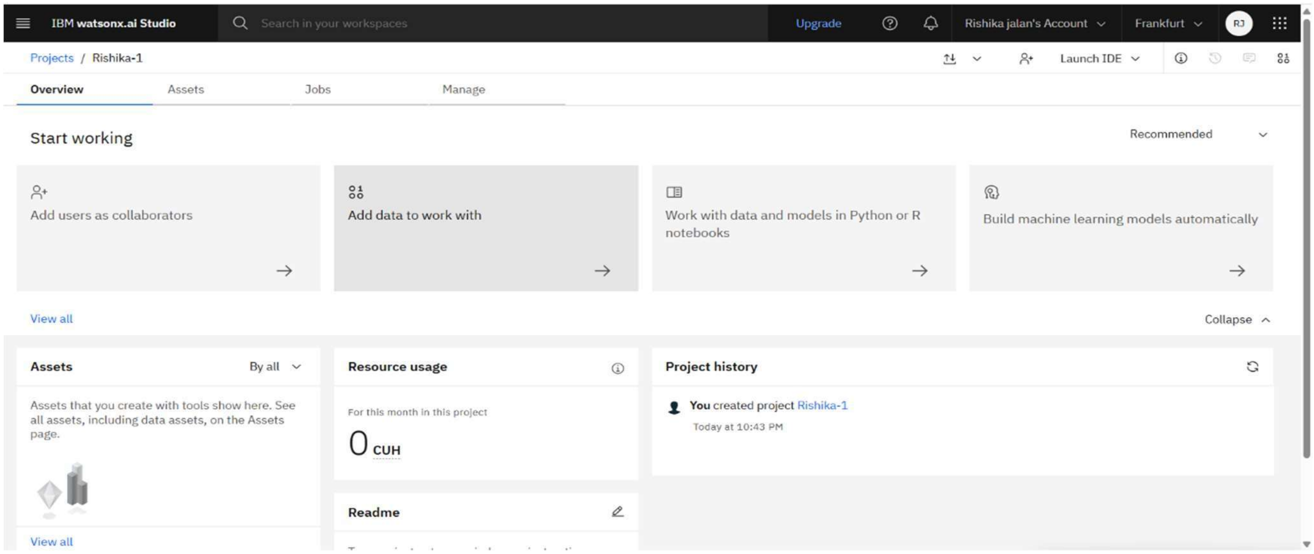
STEP 5: Creating Watsonx.ai Studio Runtime service instance



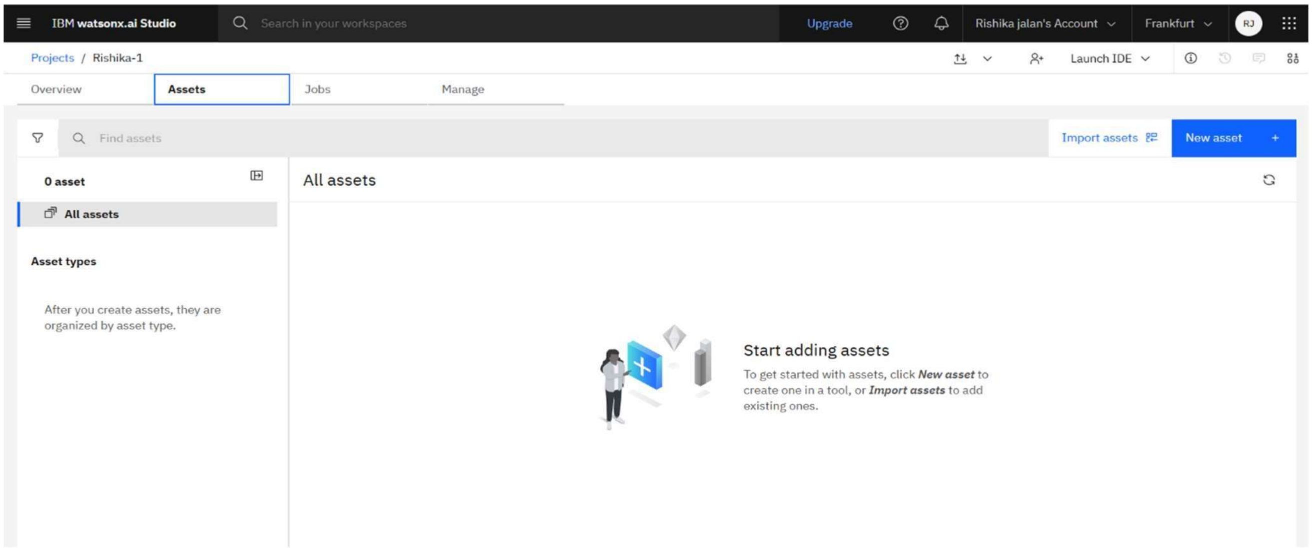
STEP 6: Creating a new project with associated cloud object storage



STEP 7: Overview of the watsonx.ai studio project

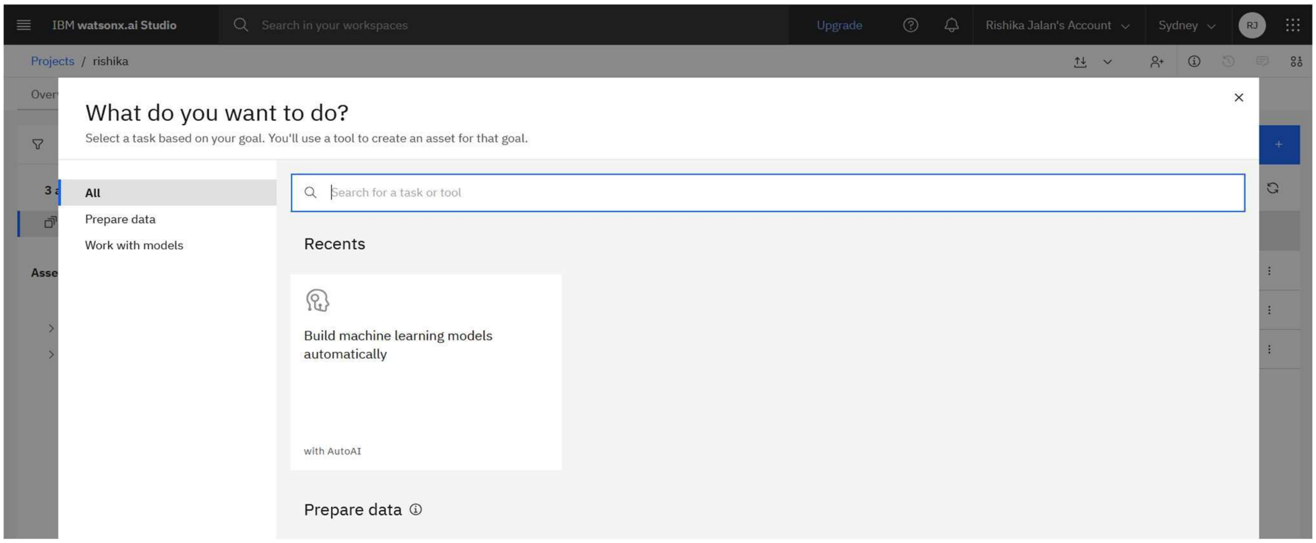


STEP 8: Assets of watsonx.ai Studio project

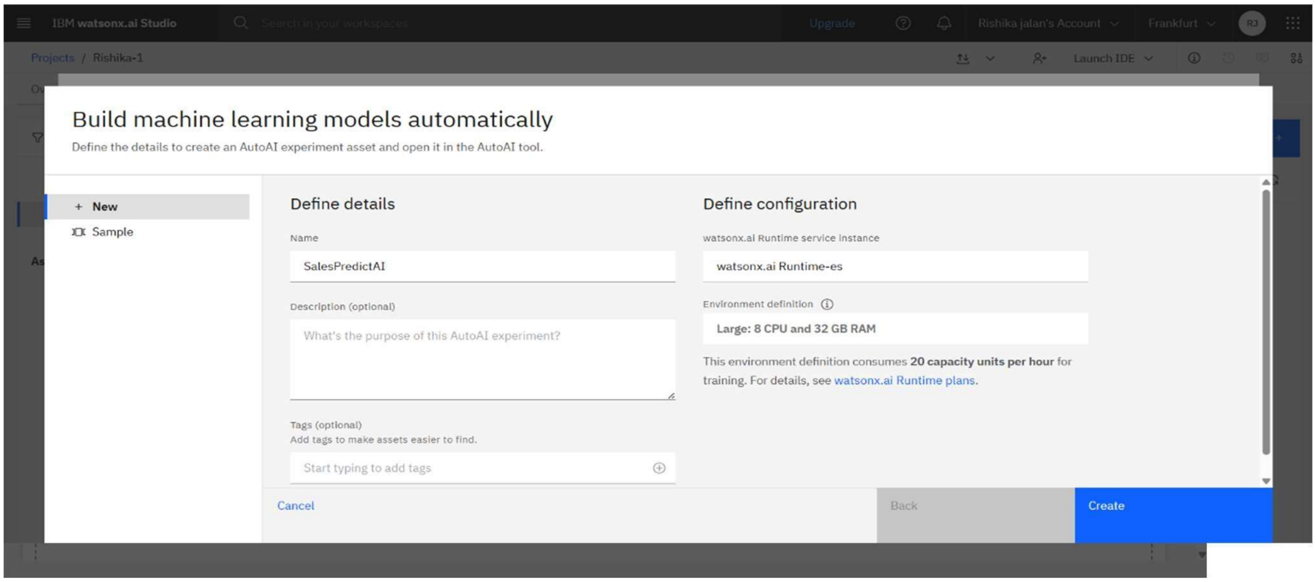


MODEL CREATION USING AUTOAI

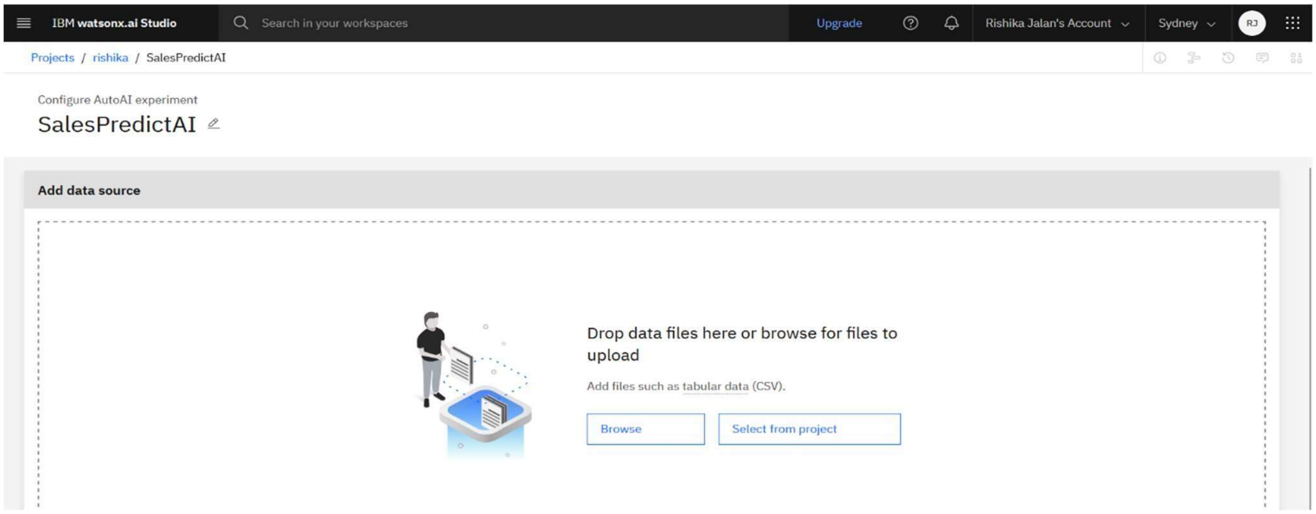
STEP 1: Selecting new asset as build machine learning model automatically



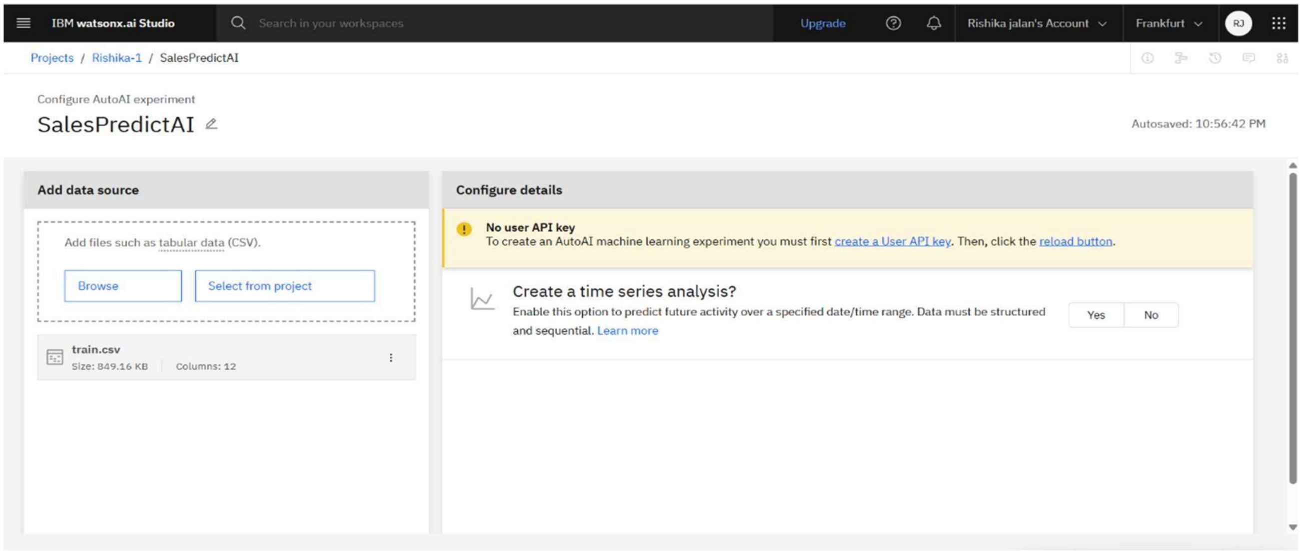
STEP 2: Connecting watsonx.ai runtime service instance



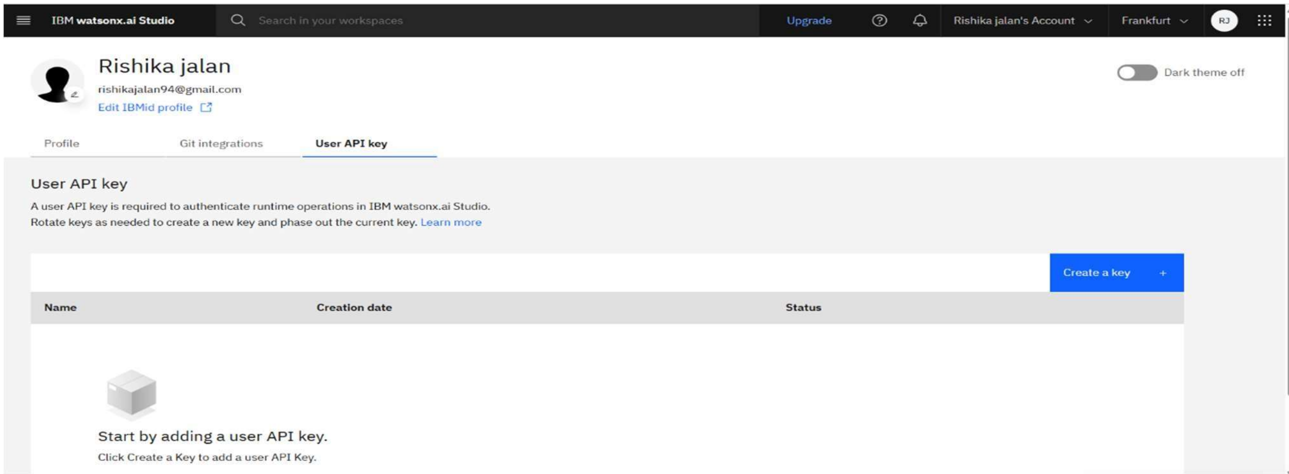
STEP 3: Importing Dataset from local files though Browse



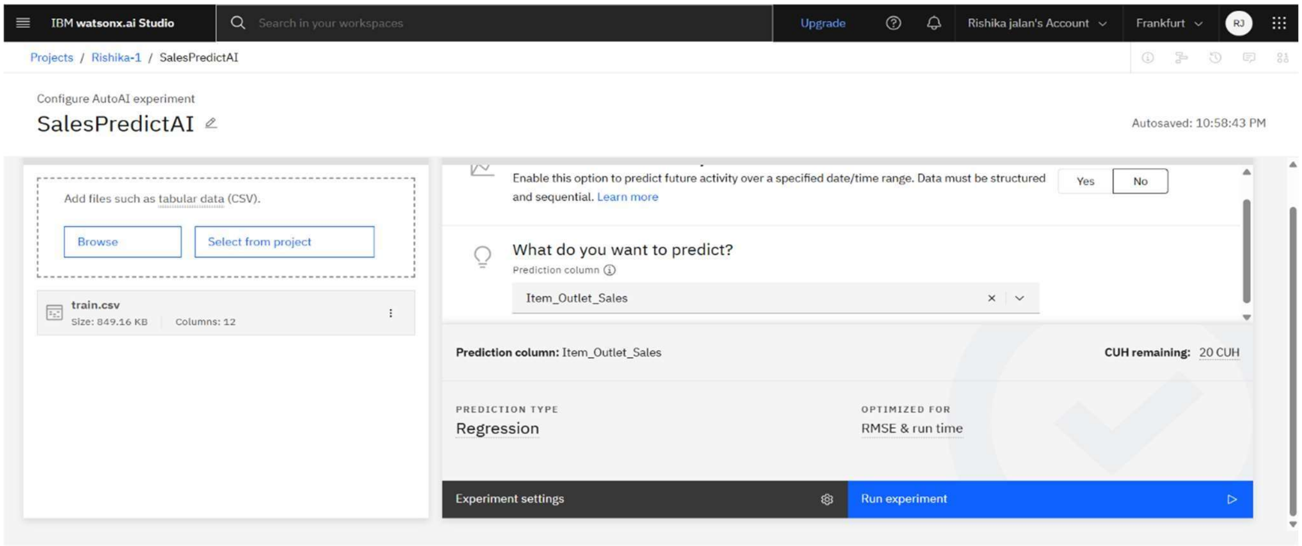
STEP 4: Configure details after fetching training dataset



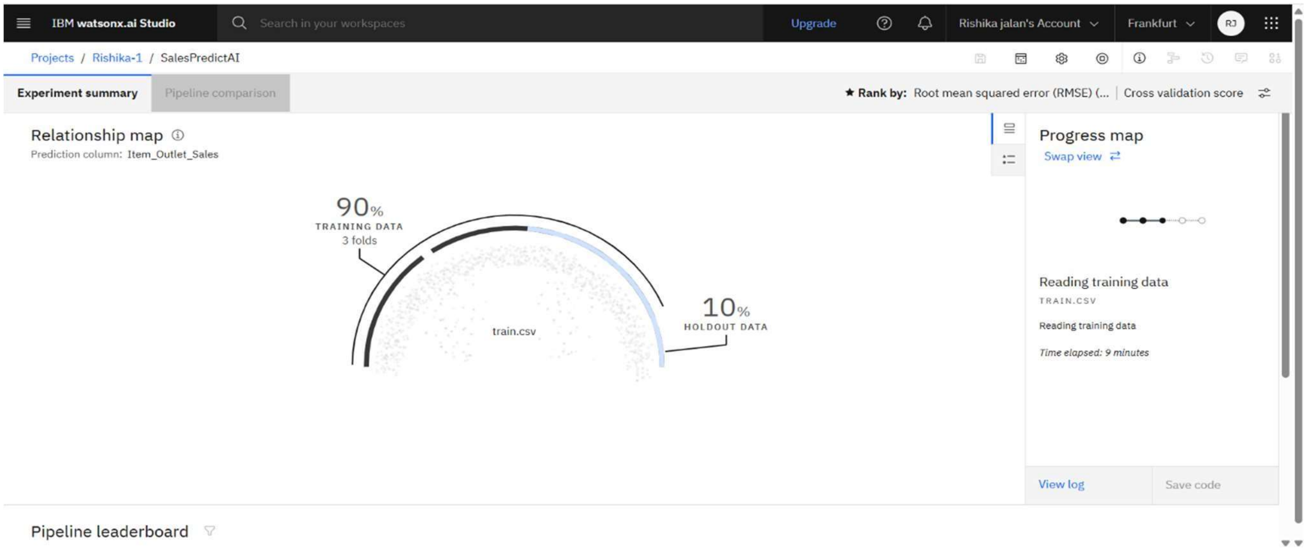
STEP 5: Creating User API Key



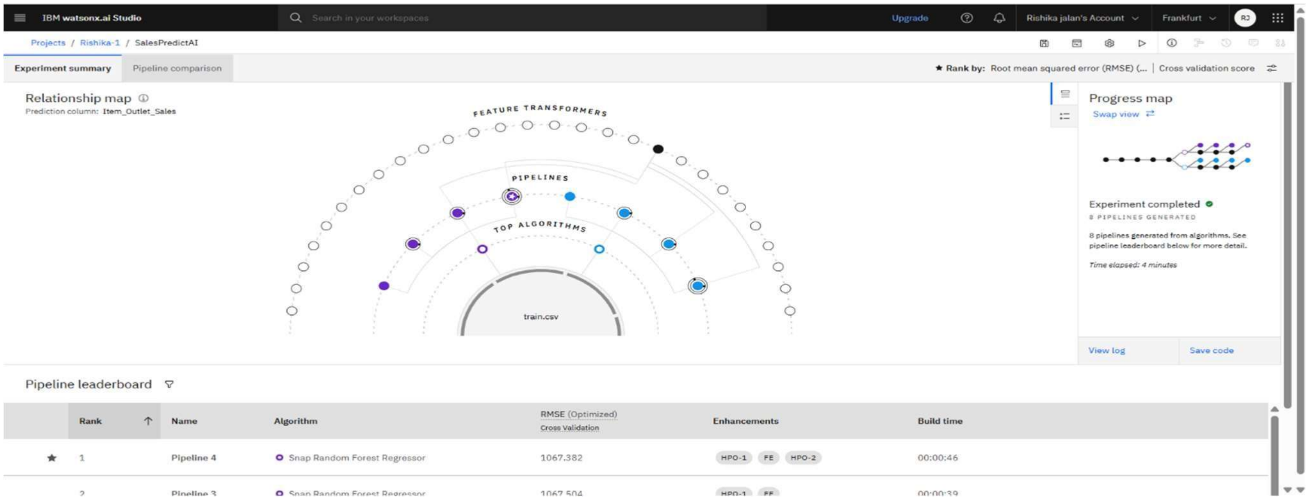
STEP 6: Selecting features to be predicted with type



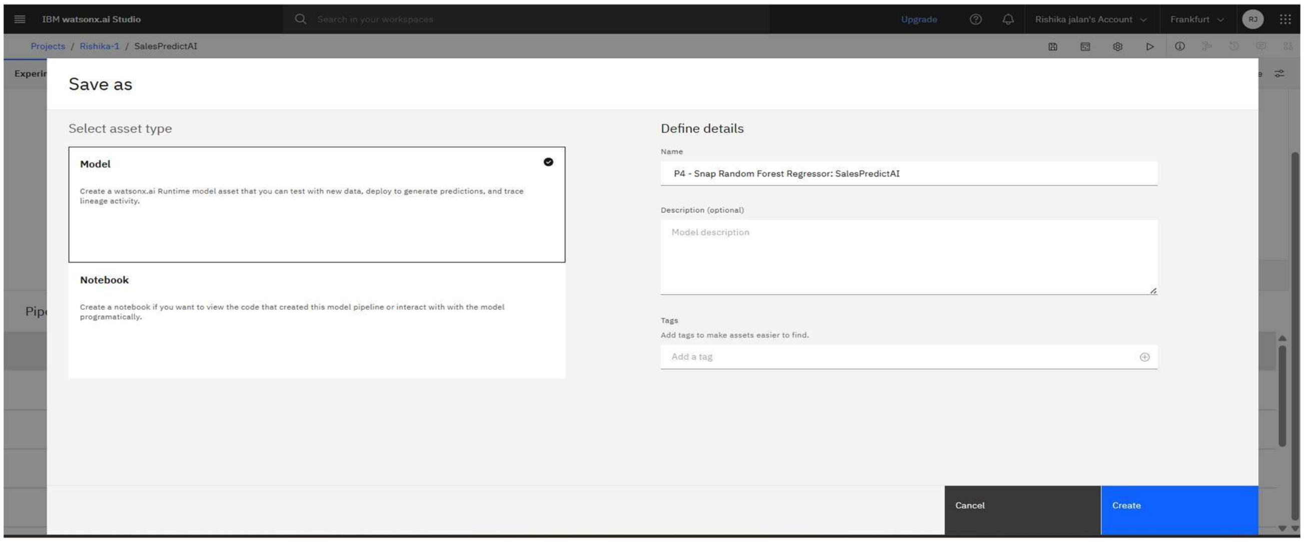
STEP 7: Reading training data in train.csv



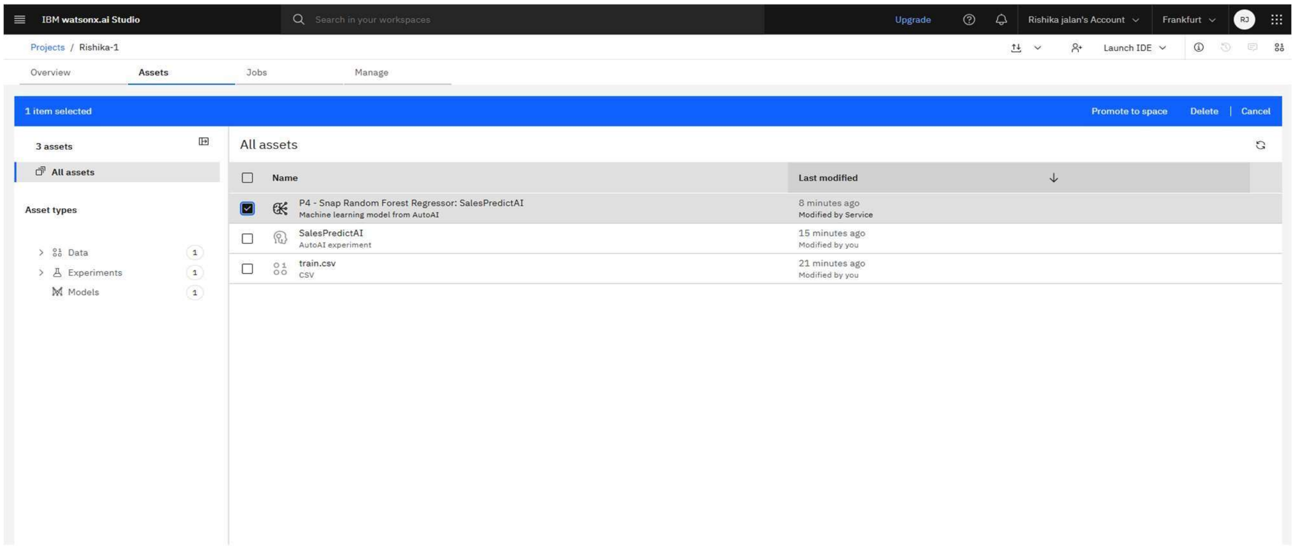
STEP 8: Building pipeline automatically with AutoAI



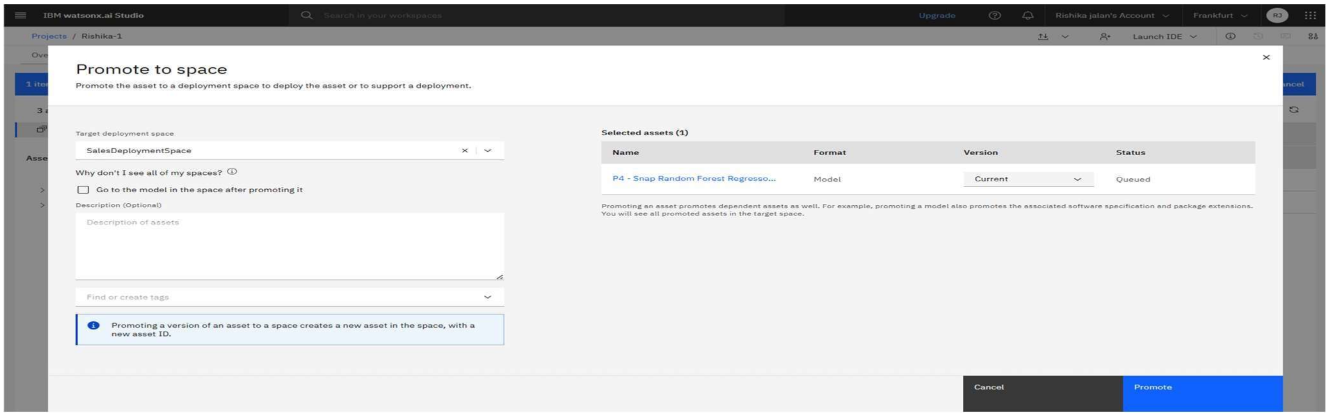
STEP 9: Saving the best Model according to Pipeline Leaderboard – Snap Random Forest Regression



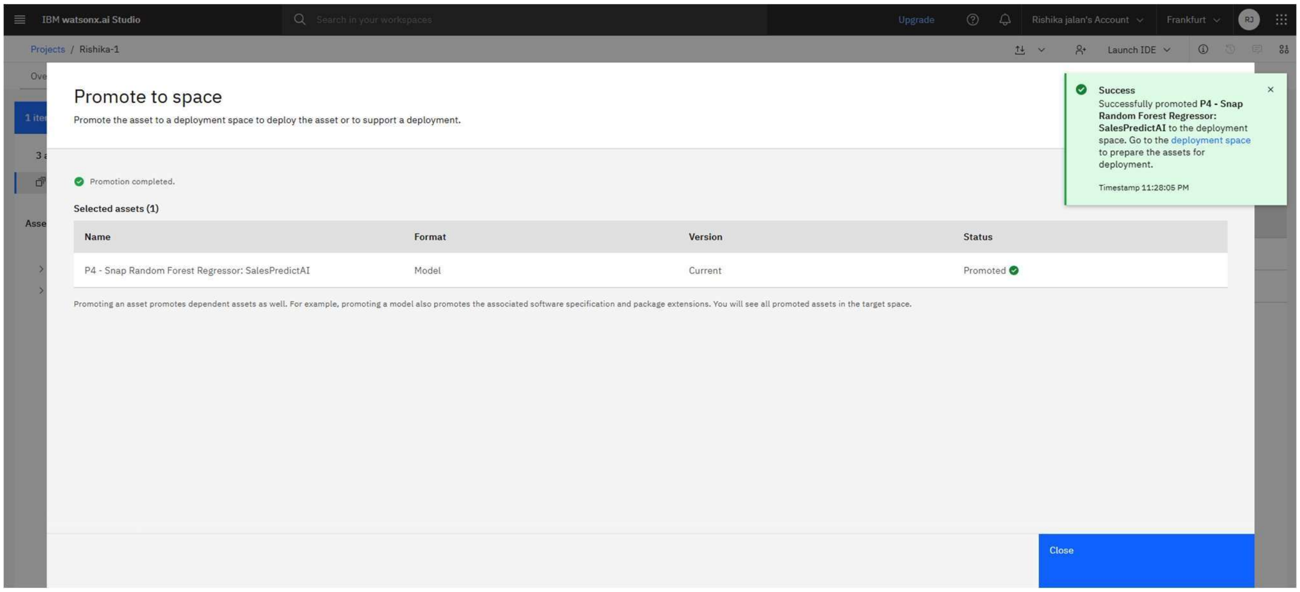
STEP 10: Promote to space



STEP 11: Promoting to space using deployment space

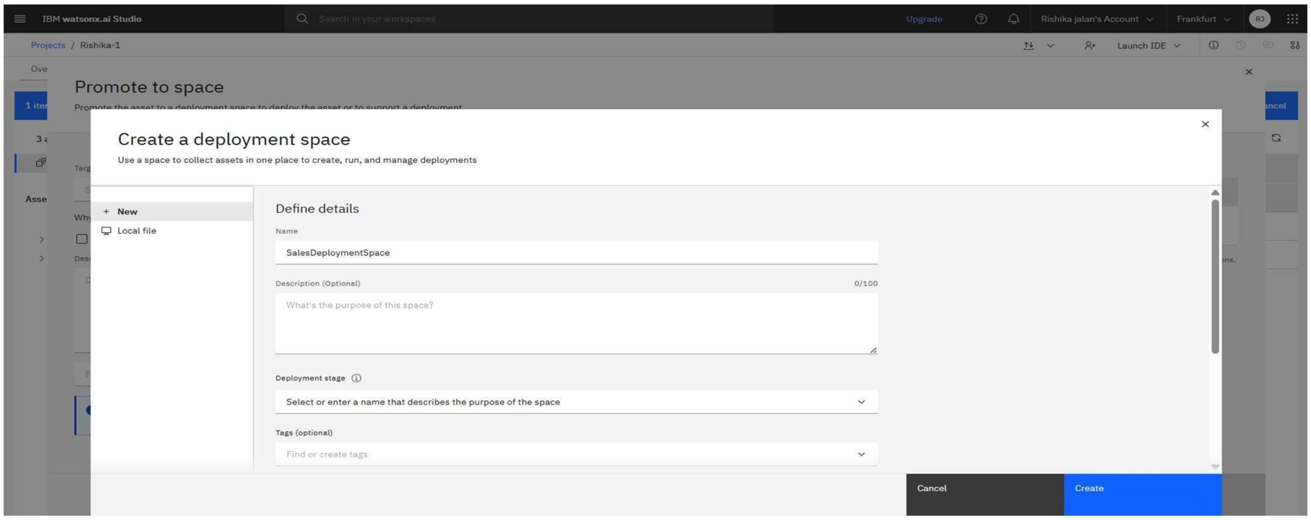


STEP 12: Promoted to space

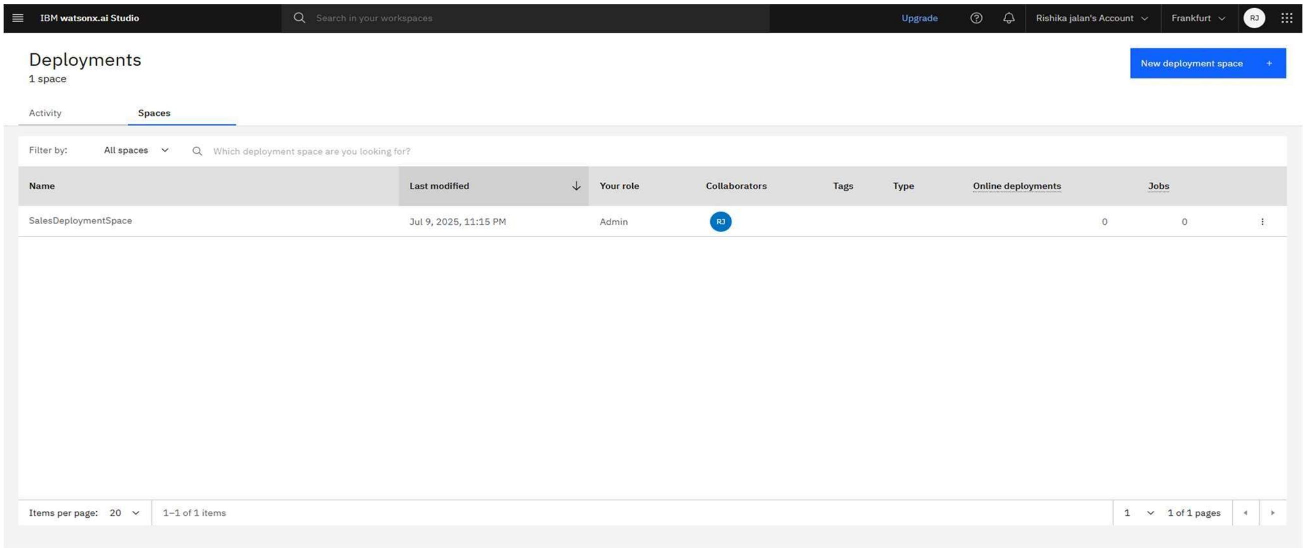


MODEL DEPLOYMENT

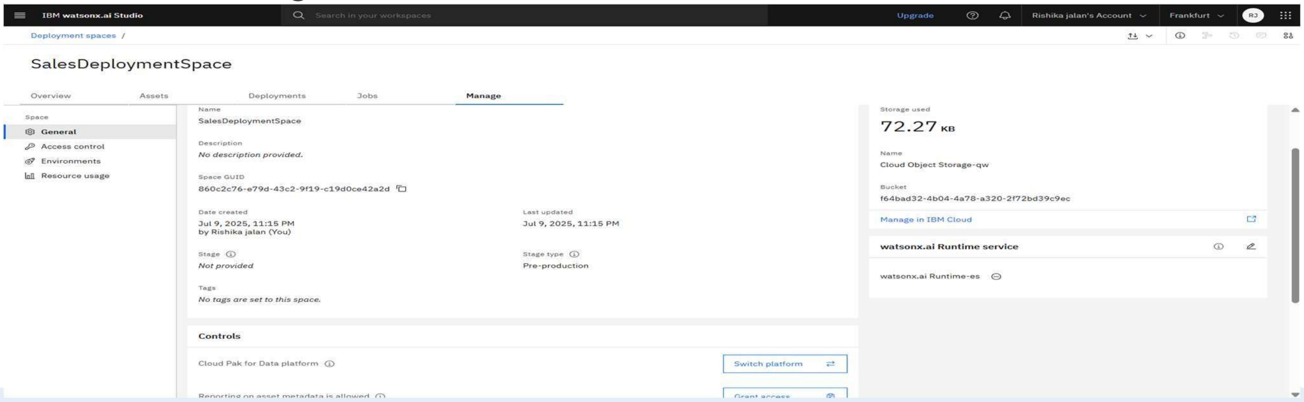
STEP 1: Creating a Deployment Space



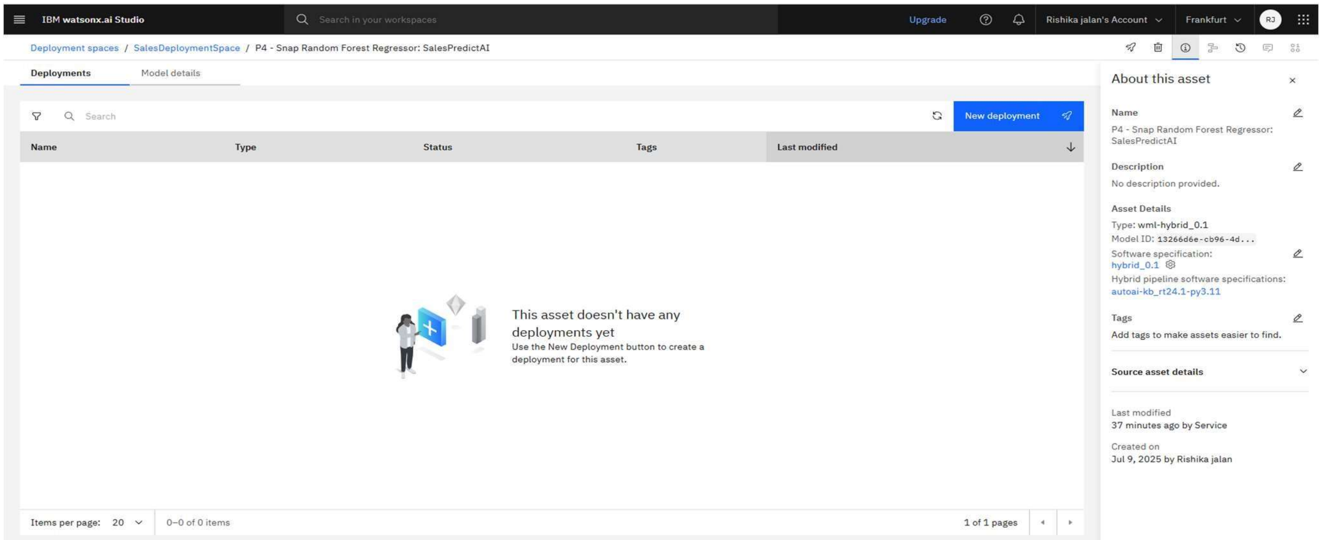
STEP 2: Deployment Space – SalesDeploymentSpace



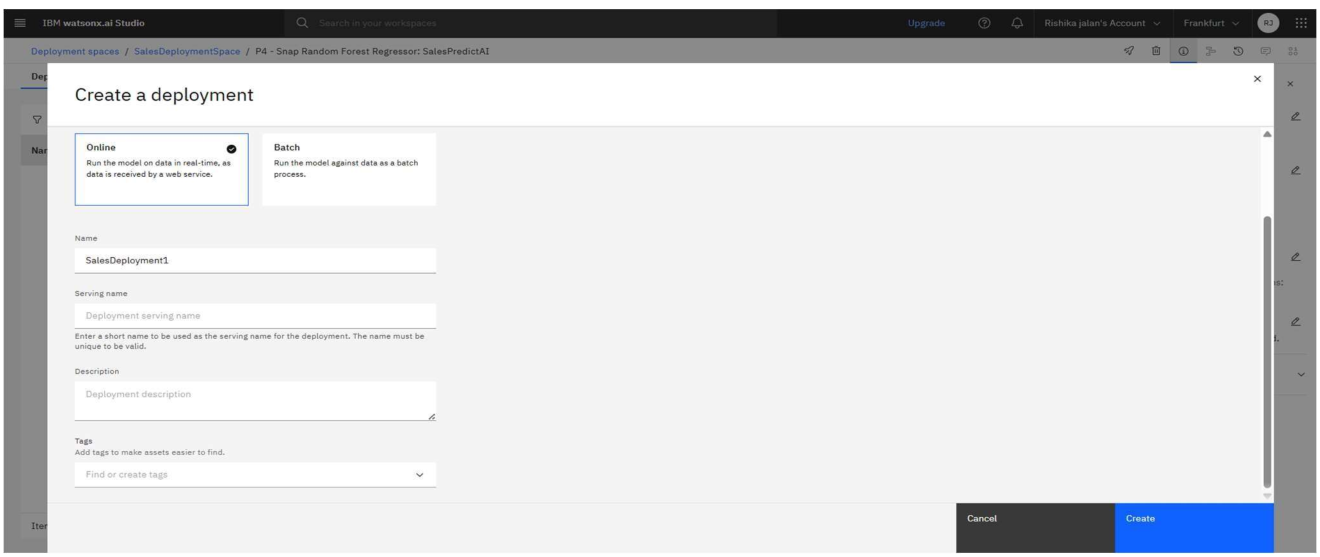
STEP 3: Connecting watsonx.ai runtime service



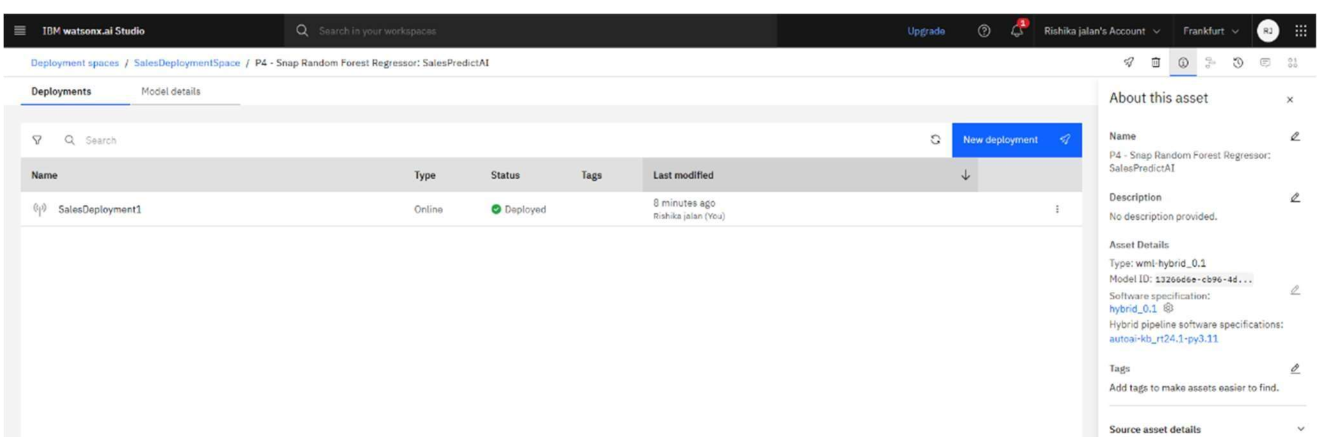
STEP 4: New deployment



STEP 5: Creating an online deployment

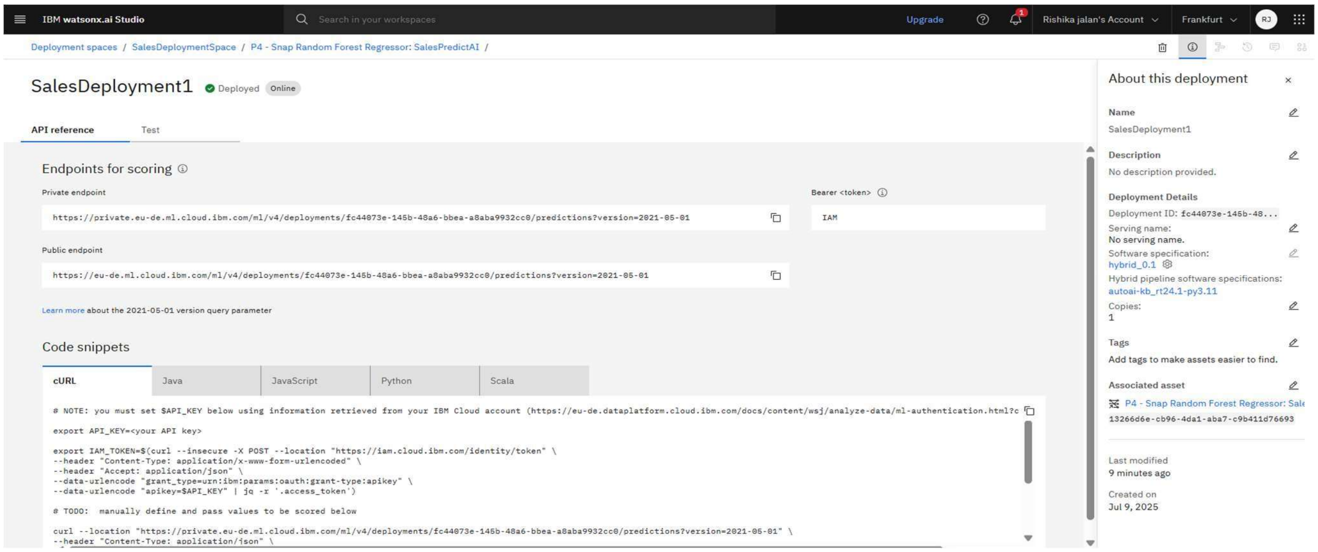


STEP 6: Deployed model by creating new Online deployment

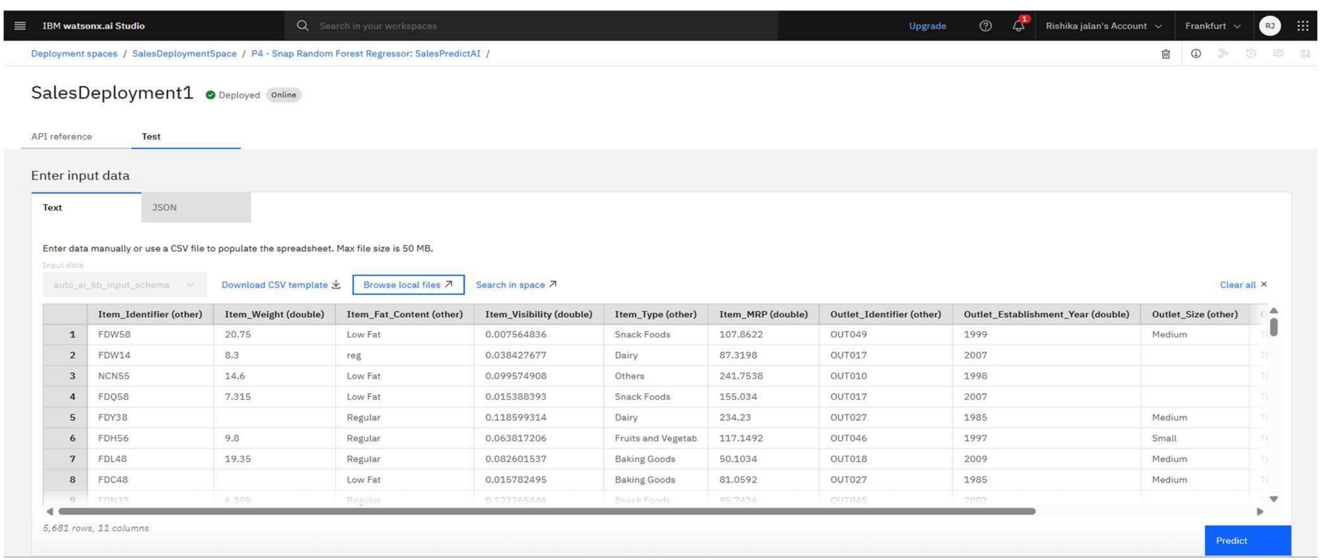


MODEL TESTING

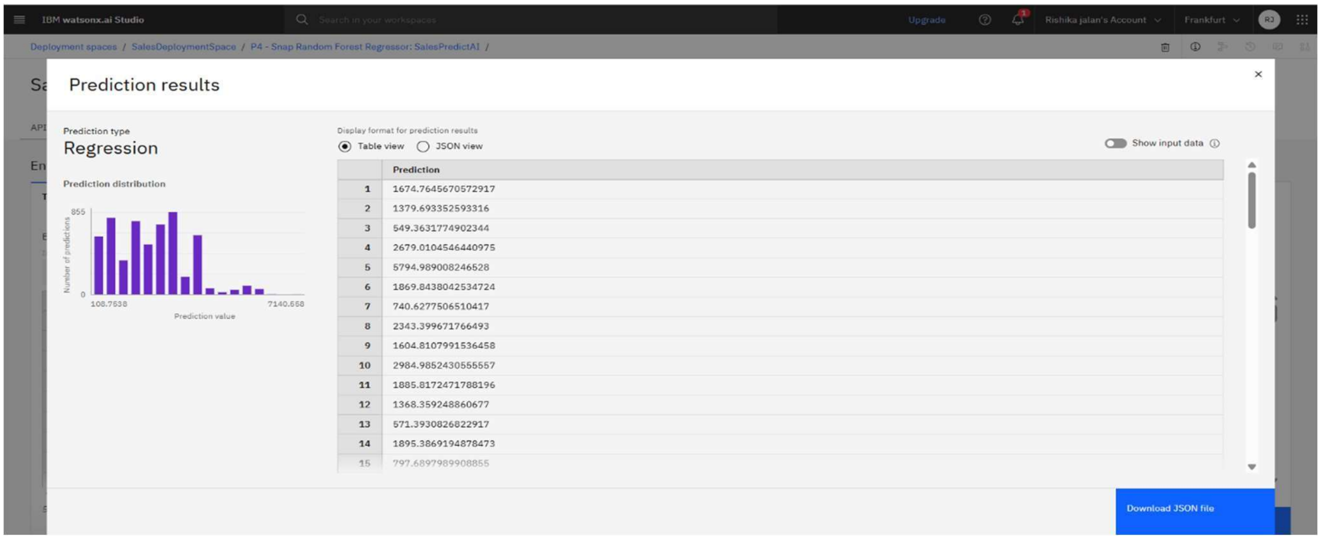
STEP 1: SalesDeployment1 API reference for testing



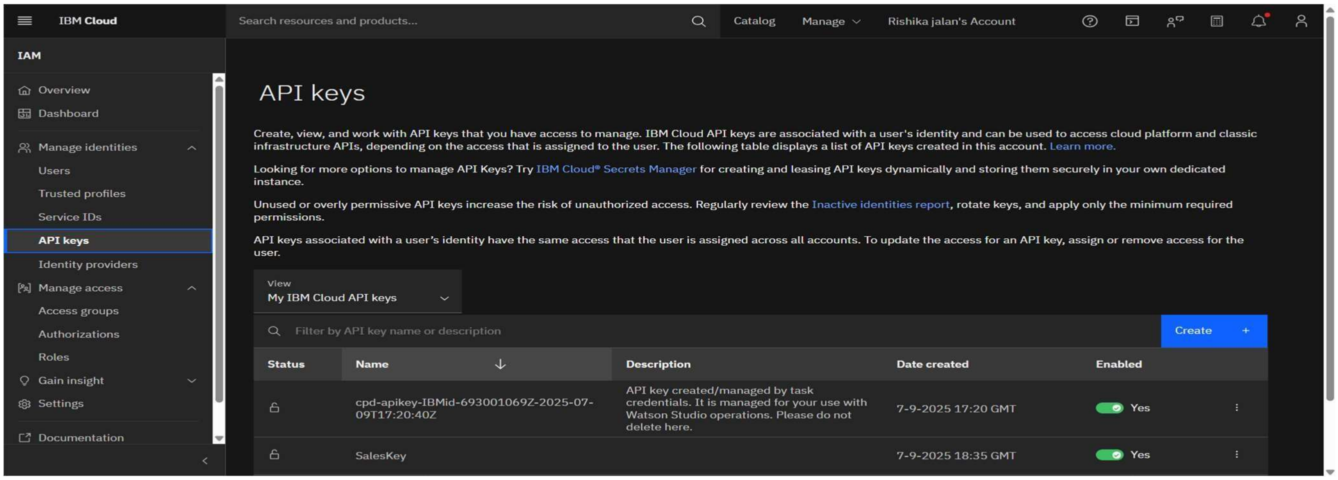
STEP 2: Importing Testing Dataset from local file to test the Model



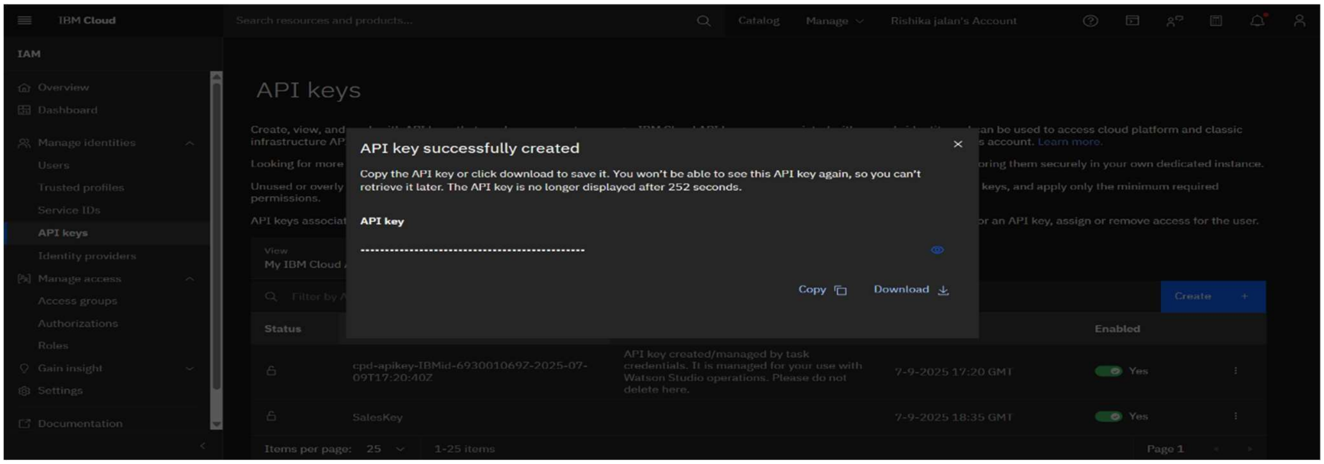
STEP 3: Predicted Result of the test dataset



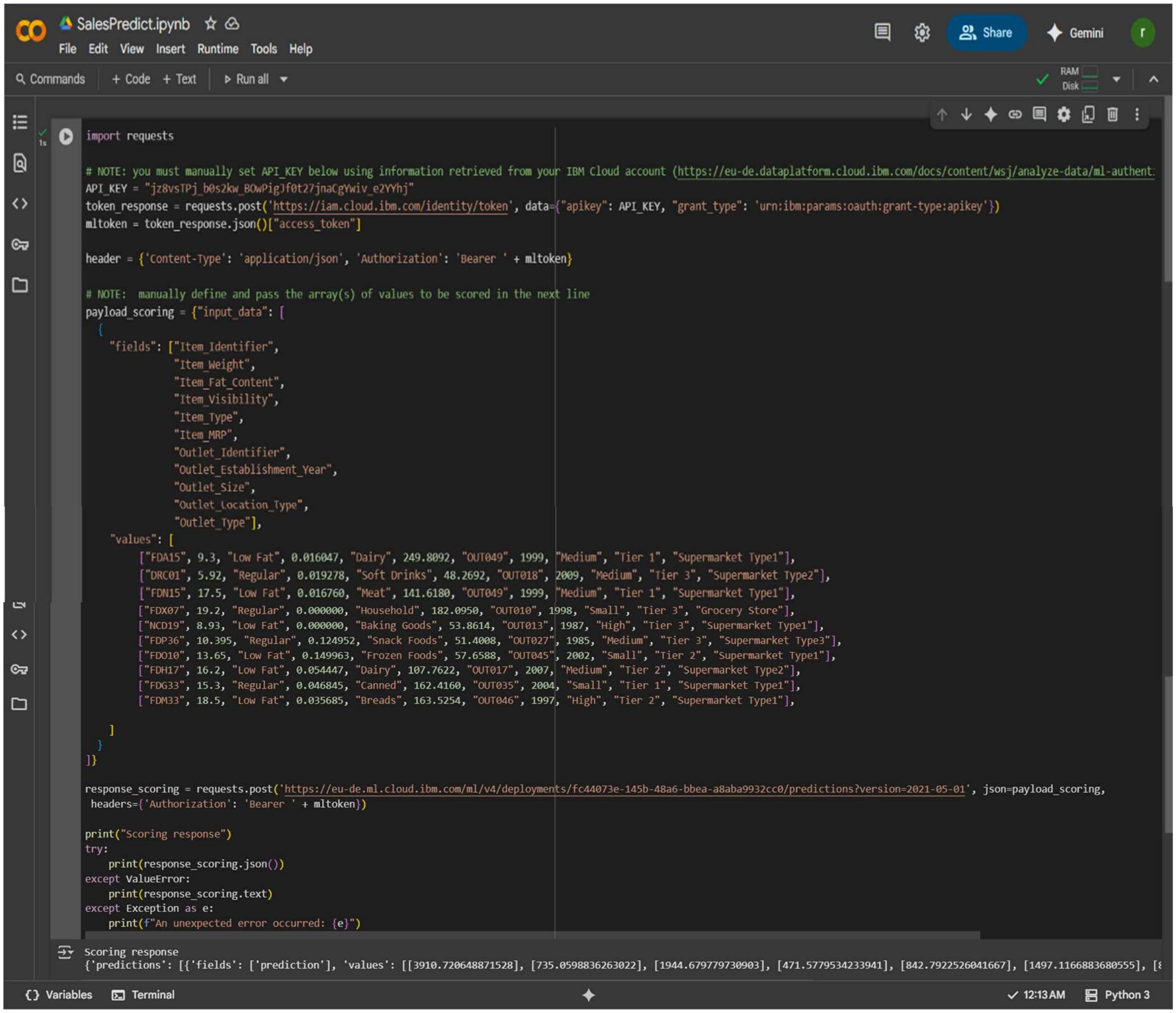
STEP 4: Create API Keys



STEP 5: API Key created successfully



STEP 6: Using API key and public endpoints to predict the results for some sample data on Google Colab Notebook



CONCLUSION

In this project, I leveraged **IBM Watson Studio's AutoAI** to develop a predictive model for forecasting product sales across Big Mart outlets. AutoAI simplified the machine learning workflow by automating key steps such as data preprocessing, feature engineering, and model optimization, which significantly reduced development time and complexity.

By training on the **Big Mart Sales Prediction dataset**, the model provided useful insights into how product attributes and outlet characteristics affect sales. These predictions can help Big Mart improve inventory planning, optimize stock levels, and make data-driven marketing decisions.

Overall, this project highlights the power of **AutoAI** in turning complex retail data into actionable business strategies. Moving forward, further improvements could include integrating more recent sales data, exploring additional features, and deploying the model as an API to support real-time decision-making.