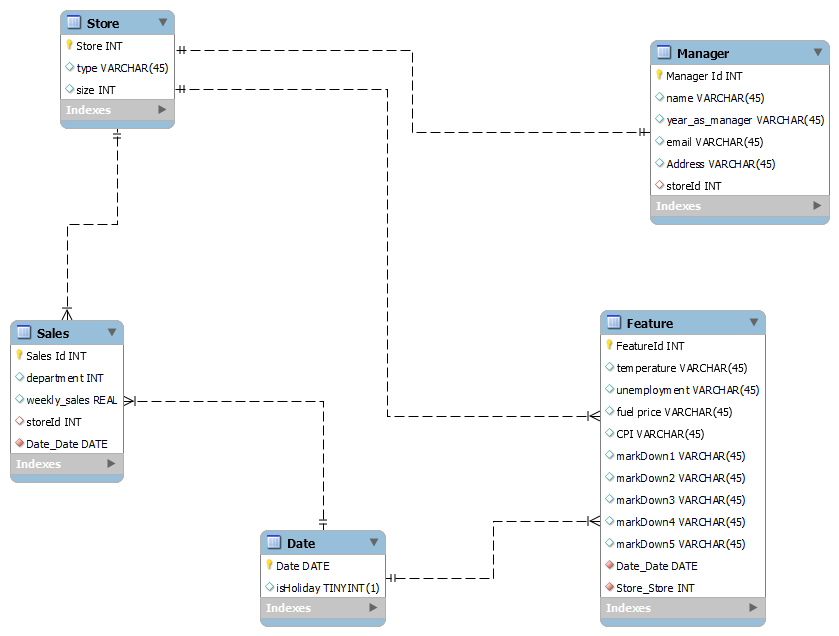
Course Work:

1. Entity Relation Diagram:

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of the relationships that exist between people, objects, places, concepts, or events in an information technology (IT) system.

The model in the figure below represents the relationship between Store Info, Store Data, Sales Data, and Features Data.



Design Choices:

Before considering my design choices, these guidelines were followed

* Determining the purpose of my dataset.
* Finding and organizing the information required
* Dividing the information into tables.
* Turn information items into tables.
* Specify primary keys.
* Set up the table relationships.
* Refine your data.
* Apply normalization rule.
* Separation of manager information and using store id to link the information.
* Remove of dates and making it a table where it has a one to many relation between sales and features table respectively.

1. Columns Names:

* Manager Schema:

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Attribute |
| Manager Id | Int | Primary Key. Not null |
| Store | Int | Secondary key. Not null. |
| Manager | String |  |
| Years\_as\_manager | Int |  |
| Email | String |  |
| Address | String |  |

* Store Schema:

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Attribute |
| Store | Int | Secondary key. Not null. |
| type | String |  |
| size | Int |  |

* Features Schema:

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Attribute |
| Feature Id | Int | Primary Key.  Not null. |
| Store | Int | Secondary key. Not null. |
| Date | Date |  |
| temperature | Double |  |
| Markdown1 | String |  |
| Markdown2 | String |  |
| Markdown3 | String |  |
| Markdown4 | String |  |
| Markdown5 | String |  |
| Cpi | Double |  |
| Unemployment | Double |  |

* Sales Schema:

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Attribute |
| Sale Id | Int | Primary Key.  Not null. |
| Store | Int | Secondary key. Not null. |
| dept | String |  |
| date | Int |  |
| Weekly sales | String |  |

* Date Schema

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Attribute |
| Date | Date | Primary Key. Not null |
| IsHoliday | Boolean |  |

1. SQL schema for all tables:

* Stores Table:

CREATE TABLE IF NOT EXISTS stores(

store INTEGER PRIMARY KEY,

type TEXT,

size INTEGER

)

* Managers Table:

CREATE TABLE IF NOT EXISTS managers(

managerId INTEGER PRIMARY KEY,

manager TEXT,

years\_as\_manager INTEGER,

email TEXT,

address TEXT,

store INTEGER,

FOREIGN KEY (store) REFERENCES stores(store)

)

* Sales Table:

CREATE TABLE IF NOT EXISTS sales(

saleId INTEGER PRIMARY KEY,

store INTEGER,

dept INTEGER,

weekly\_sales INTEGER,

date DATE,

FOREIGN KEY (store) REFERENCES stores(store),

FOREIGN KEY (date) REFERENCES dates(date)

)

* Features Table:

CREATE TABLE IF NOT EXISTS features(

featureId INTEGER PRIMARY KEY,

store INT,

temperature REAL,

fuel\_price REAL,

markdown1 Text,

markdown2 Text,

markdown3 Text,

markdown4 Text,

markdown5 Text,

date DATE,

cpi REAL,

unemployment REAL,

FOREIGN KEY (store) REFERENCES stores(store),

FOREIGN KEY (date) REFERENCES dates(date)

)

* Dates Table:

CREATE TABLE IF NOT EXISTS dates(

date DATE PRIMARY KEY,

isHoliday BOOLEAN

)

1. Data Cleaning:

In a SQL database, a schema is a list of logical structures of data. A database user owns the schema, which has the same name as the database.

From the entity relation model, A SQL schema can be drafted out from each of the principal blocks as follows:

* The data cleaning process used to clean data is the filling of missing values from the store\_info.csv and the stores\_data-set.csv dataset.
* Data Cleaning by filling the missing values for the store\_info.csv dataset:  
  store = pd.read\_csv('store\_info.csv')

store.describe(include = "all")  
store.fillna({'Manager': 'George Parker'}, limit = 1, inplace = True)

* Data Cleaning by filling the missing values for the store\_data-set.csv dataset:  
  store\_data = pd.read\_csv('stores\_data-set.csv')

Store\_data.describe(include = "all")

Data Cleaning Notes:

1. The data cleaning procedure was implemented using python library - Pandas
2. The fillna() function was used in filling the missing values in both datasets
3. The missing values was filled one at a time using the limit parameter
4. The inplace parameter was used so that the changes in the dataset can take effect permanently.
5. In the store\_info dataset, the values used for filling the missing data were gotten by using values from the either the mail given or from other data sources like Kaggle.
6. In the store\_data-set dataset, the values used for filling the missing data were gotten from other data sources like Kaggle.
7. Working GUI:

* An email validation before updating details of a specific store.

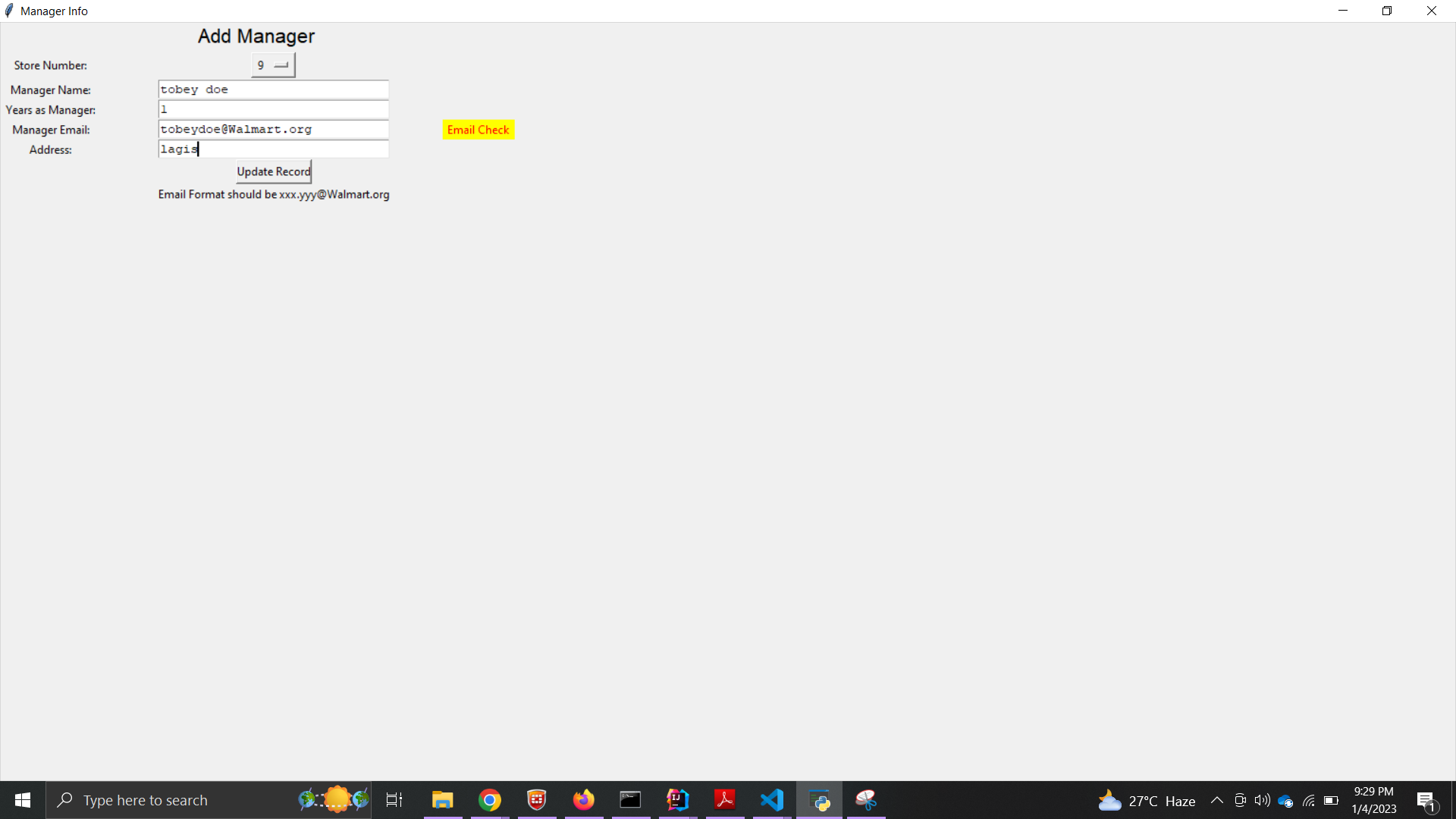


Fig 1. Working GUI for updating details of a specific store.

* Calculation of mean store size when a type of store is picked.

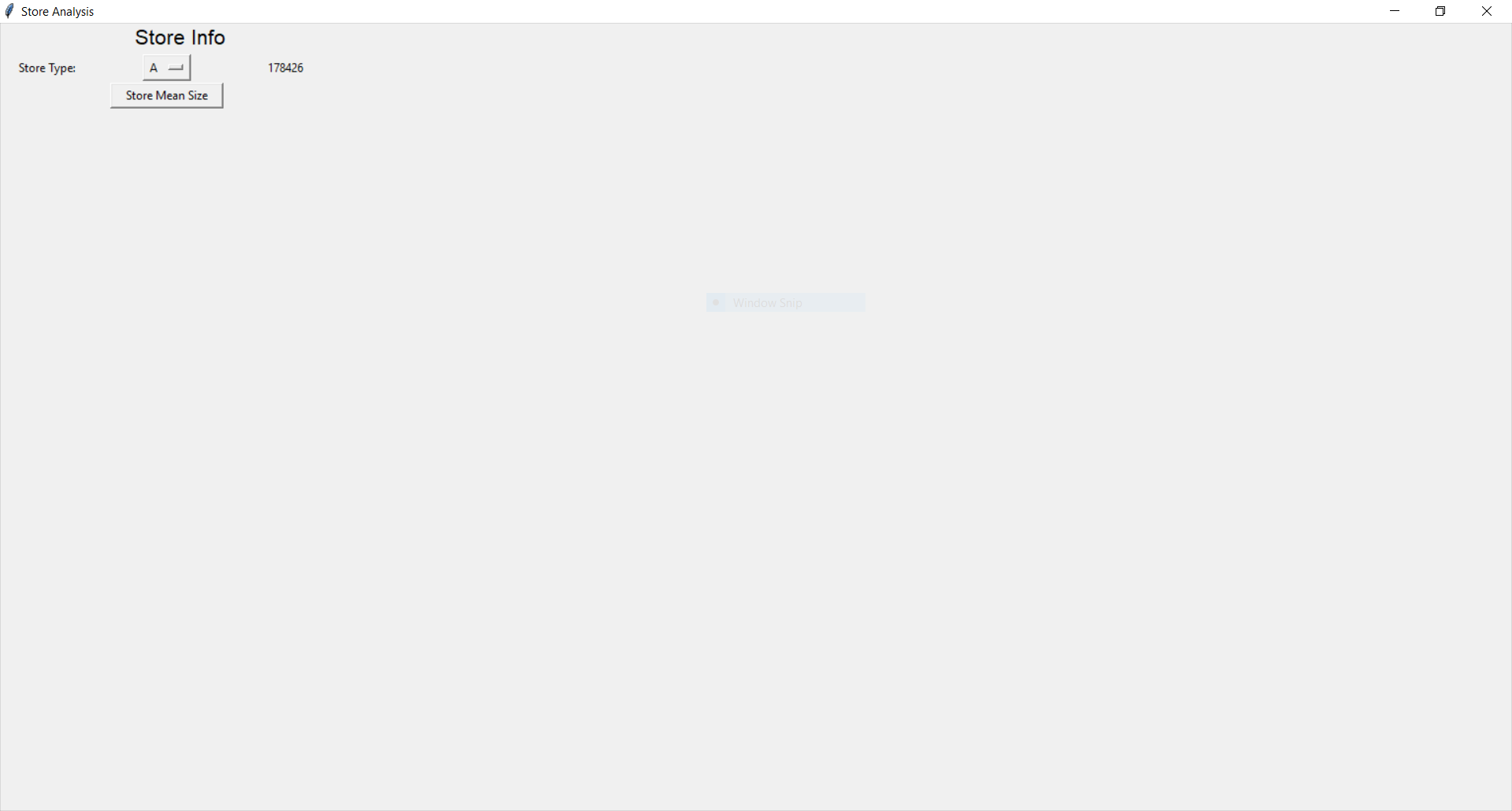


Fig 2. Working GUI for calculating mean store size of a specific store type.

* Since the records from the sales\_data-set.csv file were too much, we had to select within the range of 50 dates to make the graph more visible.

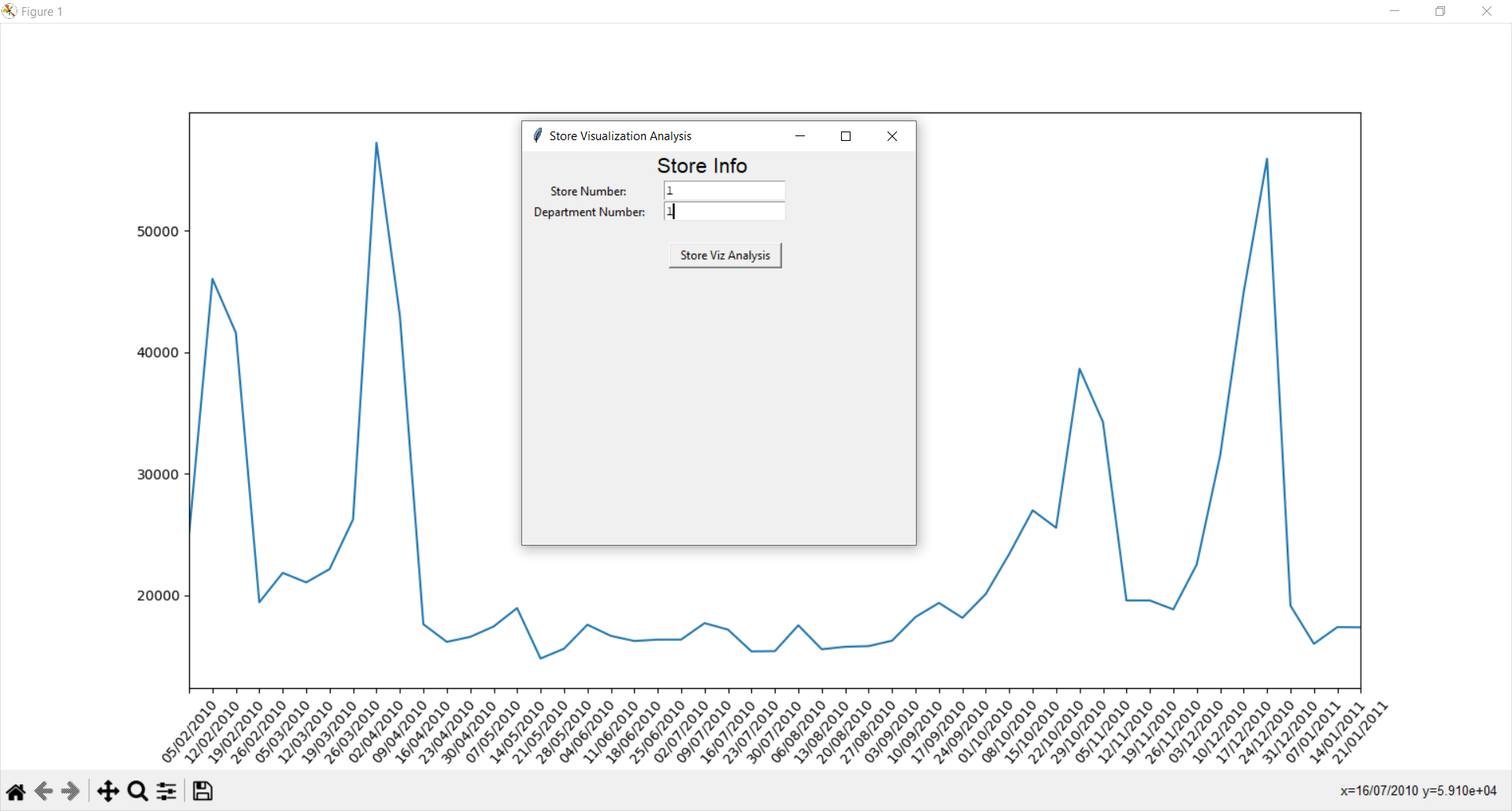


Fig 3. The graph of sales versus data for a specific store and department GUI.

1. How AWS can be used in the project: