Multivariate Statistics And Econometrics(MSE) Project

OBJECTIVE:

- We will build a predictive model to predict the sales of each product at a particular outlet.
- Also, using test hypothesis, we will compare between given variables as to how both are co-dependent.

DATA SOURCE:

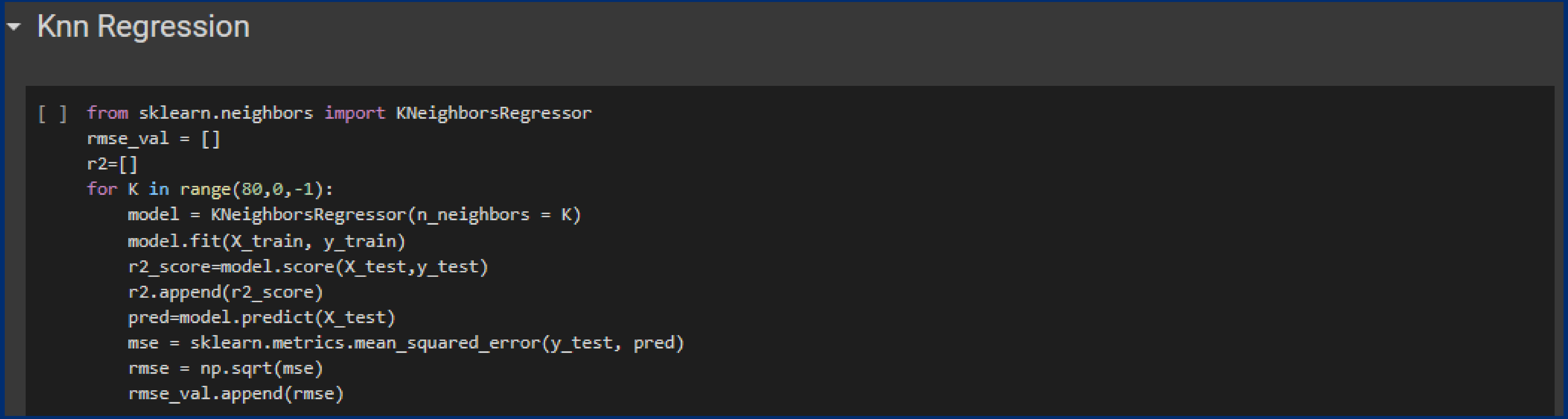
Big Mart sales data for 1559 products across 10 stores in different cities for 2013 (As collected from Kaggle).

Kaggle link:https://www.kaggle.com/datasets/brijbhushannanda1979/bigmart-sales-data?select=Train.csv.

METHODOLOGY 1:

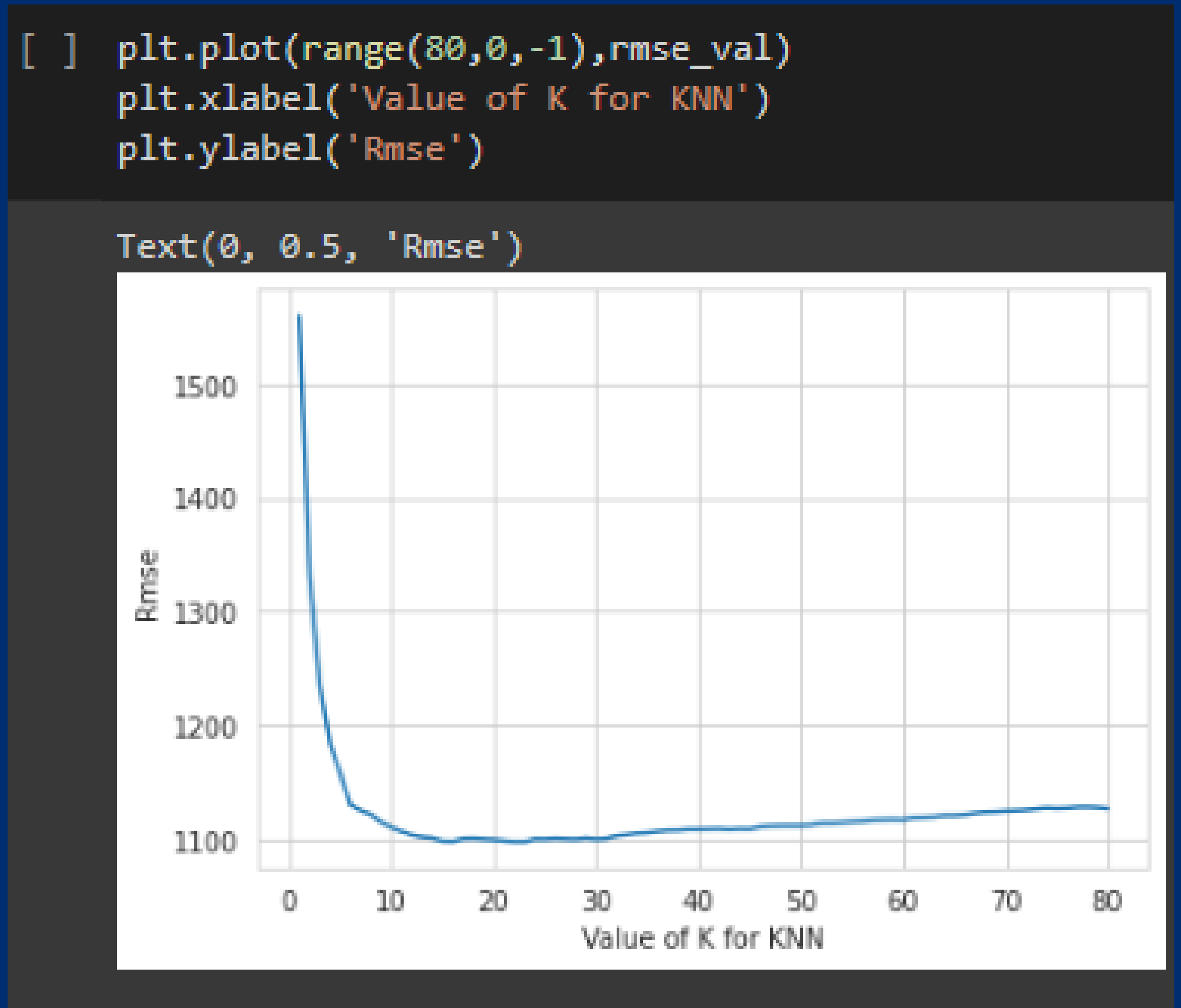
We have primarily analyzed the data structure, done a few basic EDA(Exploratory Data Analysis) on the given dataset, preprocessed the same, trained and tested the dataset, applied **Machine Learning models** such as KNN, Decision Tree and Random Forest, so as to predict the sales of each product at a particular outlet.

KNN REGRESSION:



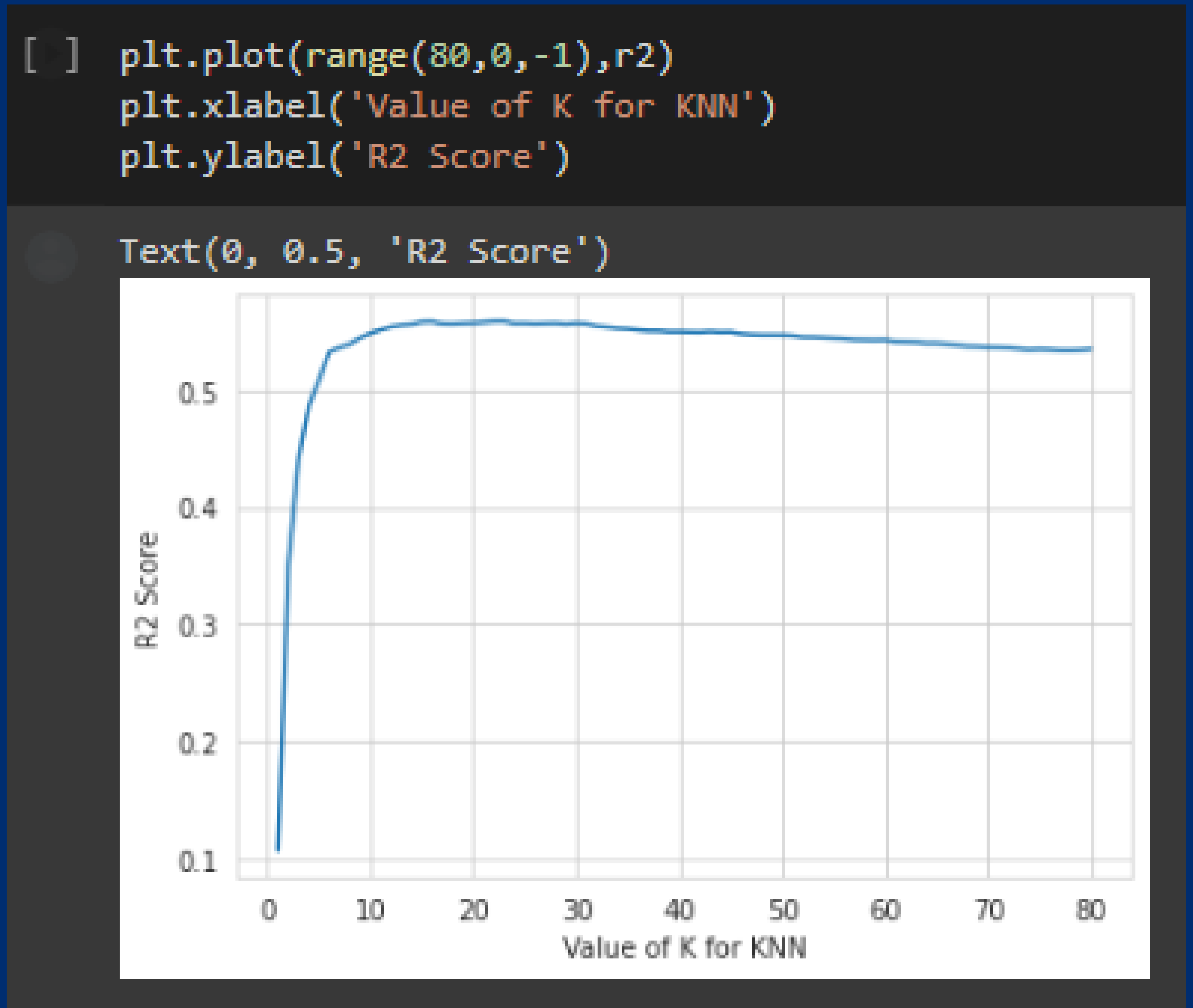
Knn Regression

```
[ ] from sklearn.neighbors import KNeighborsRegressor
    rmse_val = []
    r2=[]
    for K in range(80,0,-1):
        model = KNeighborsRegressor(n_neighbors = K)
        model.fit(X_train, y_train)
        r2_score=model.score(X_test,y_test)
        r2.append(r2_score)
        pred=model.predict(X_test)
        mse = sklearn.metrics.mean_squared_error(y_test, pred)
        rmse = np.sqrt(mse)
        rmse_val.append(rmse)
```



```
[ ] plt.plot(range(80,0,-1),rmse_val)
    plt.xlabel('Value of K for KNN')
    plt.ylabel('Rmse')
```

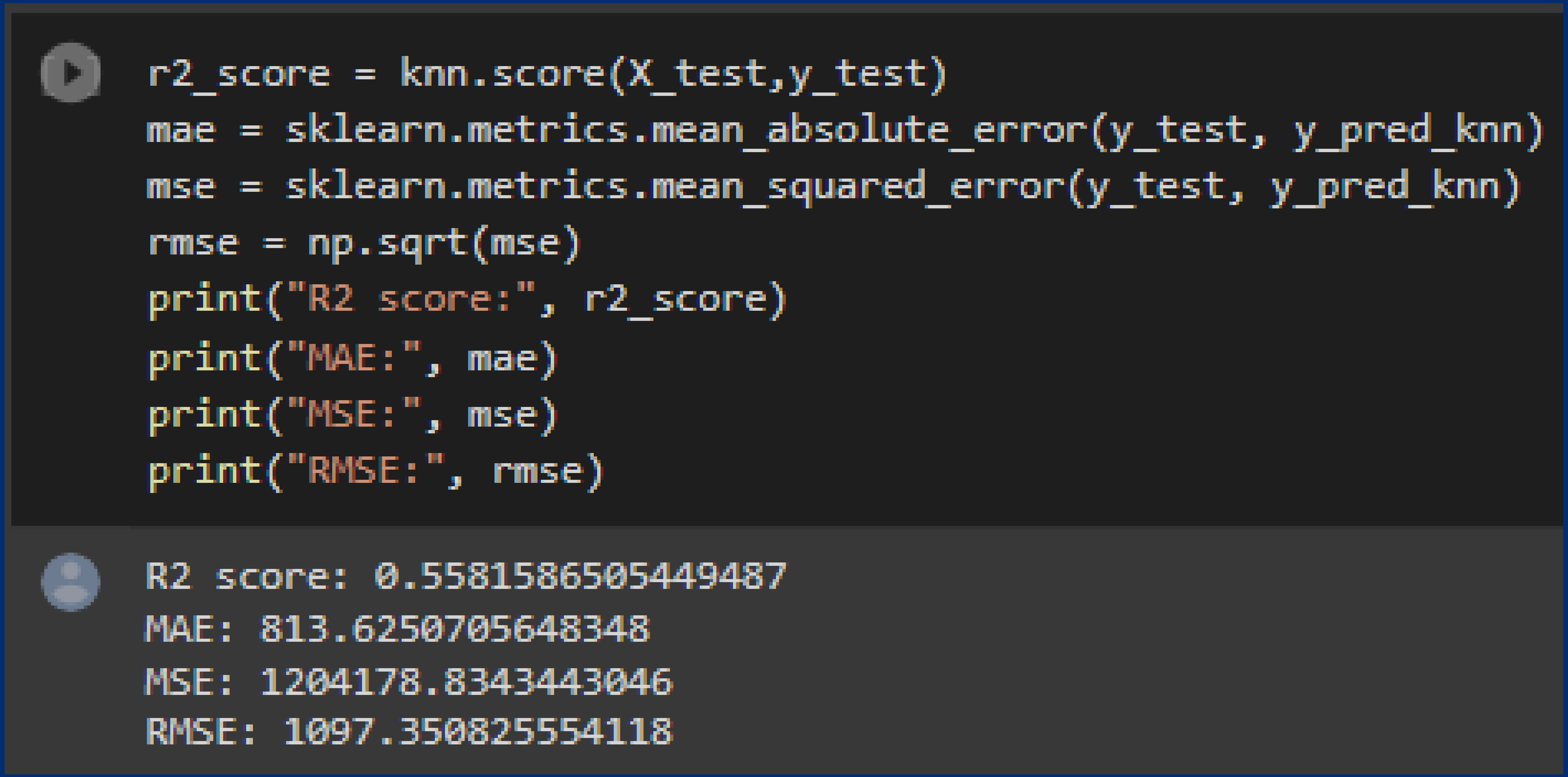
```
Text(0, 0.5, 'Rmse')
```



```
[ ] plt.plot(range(80,0,-1),r2)
    plt.xlabel('Value of K for KNN')
    plt.ylabel('R2 Score')
```

```
Text(0, 0.5, 'R2 Score')
```

Analysis and Result:

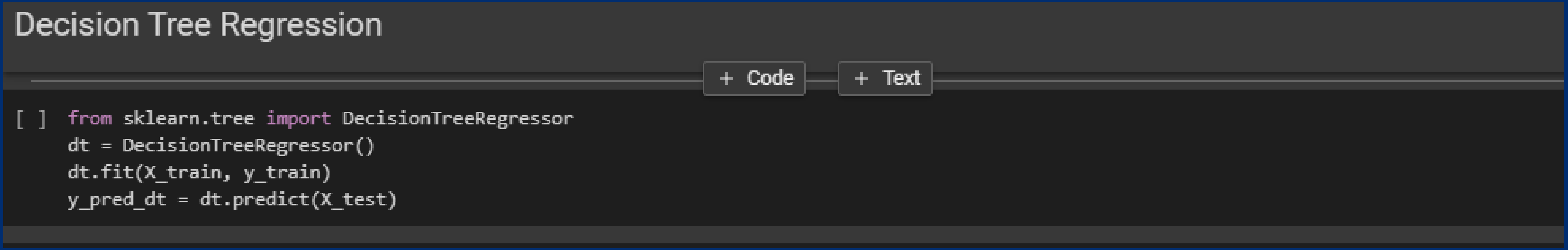


```
r2_score = knn.score(X_test,y_test)
mae = sklearn.metrics.mean_absolute_error(y_test, y_pred_knn)
mse = sklearn.metrics.mean_squared_error(y_test, y_pred_knn)
rmse = np.sqrt(mse)
print("R2 score:", r2_score)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
```

```
R2 score: 0.5581586505449487
MAE: 813.6250705648348
MSE: 1204178.8343443046
RMSE: 1097.350825554118
```

DECISION TREE REGRESSION:

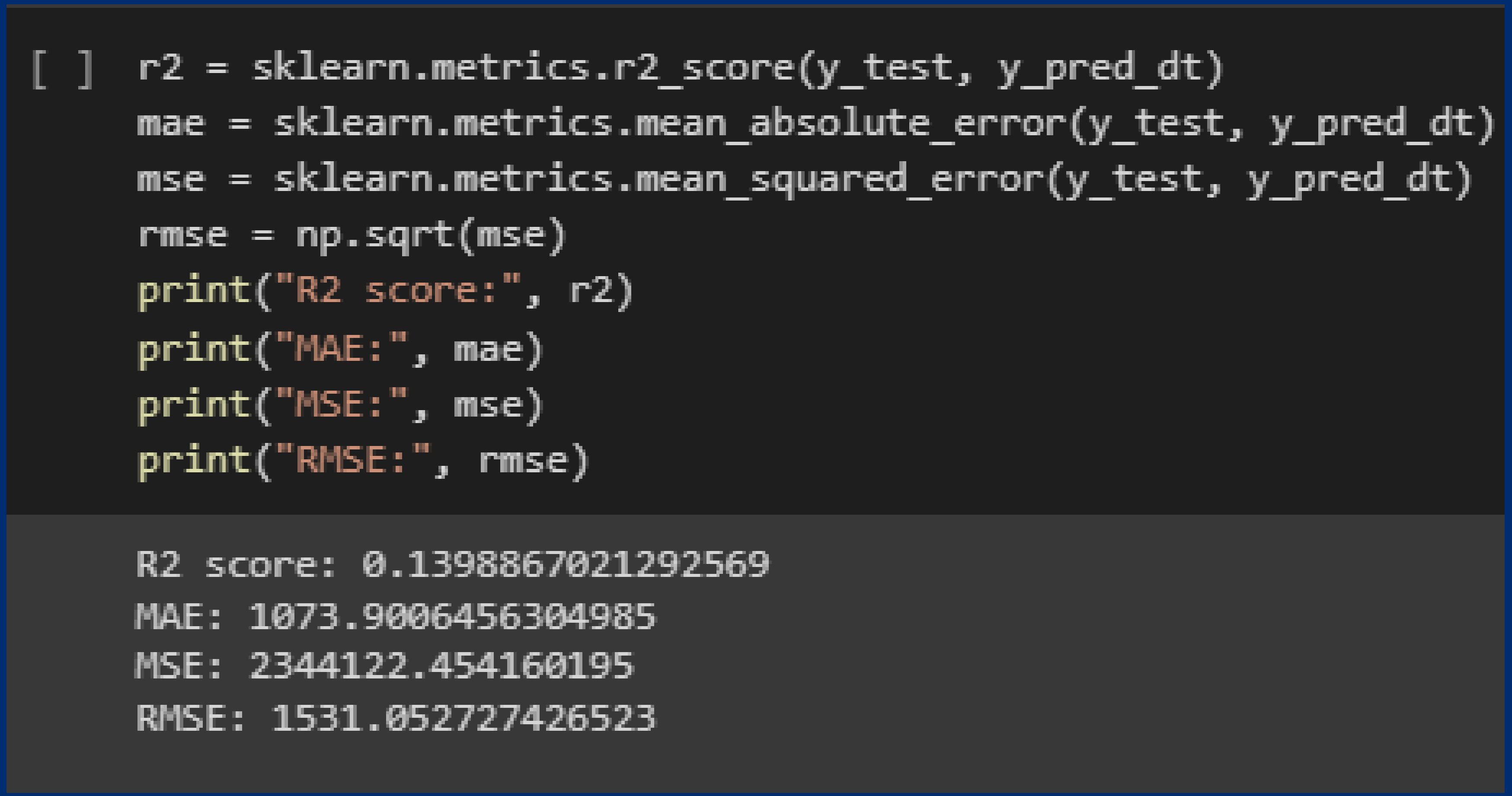
Applied the required code to implement the Decision Tree model on the dataset



Decision Tree Regression

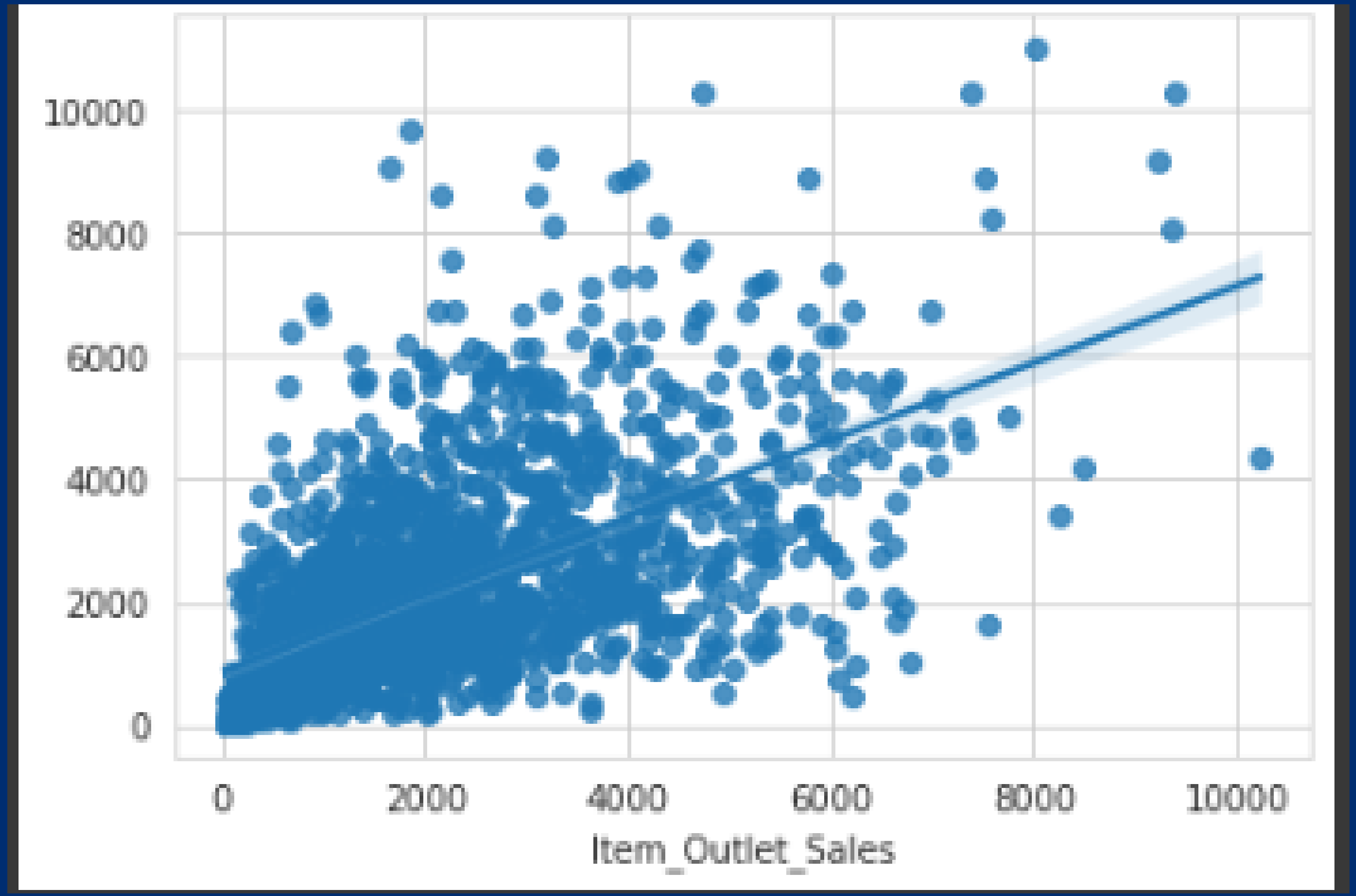
```
from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
```

Analysis and Result:



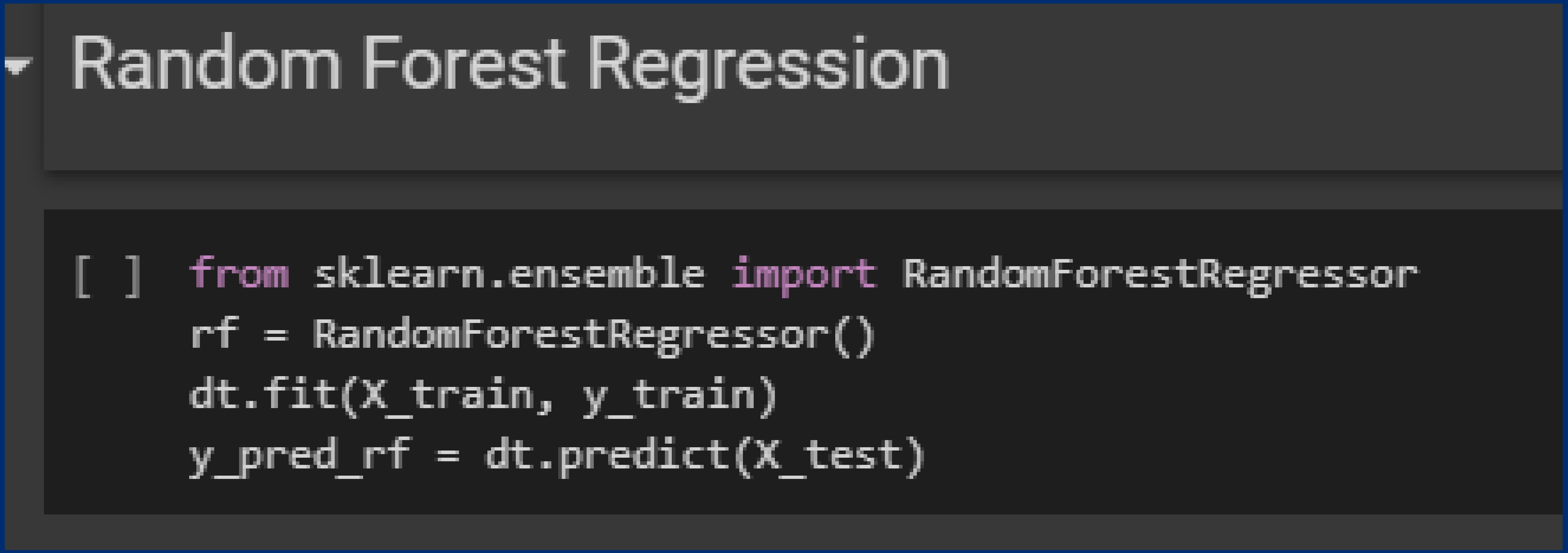
```
[ ]  r2 = sklearn.metrics.r2_score(y_test, y_pred_dt)
     mae = sklearn.metrics.mean_absolute_error(y_test, y_pred_dt)
     mse = sklearn.metrics.mean_squared_error(y_test, y_pred_dt)
     rmse = np.sqrt(mse)
     print("R2 score:", r2)
     print("MAE:", mae)
     print("MSE:", mse)
     print("RMSE:", rmse)
```

```
     R2 score: 0.1398867021292569
     MAE: 1073.9006456304985
     MSE: 2344122.454160195
     RMSE: 1531.052727426523
```



RANDOM FOREST REGRESSION:

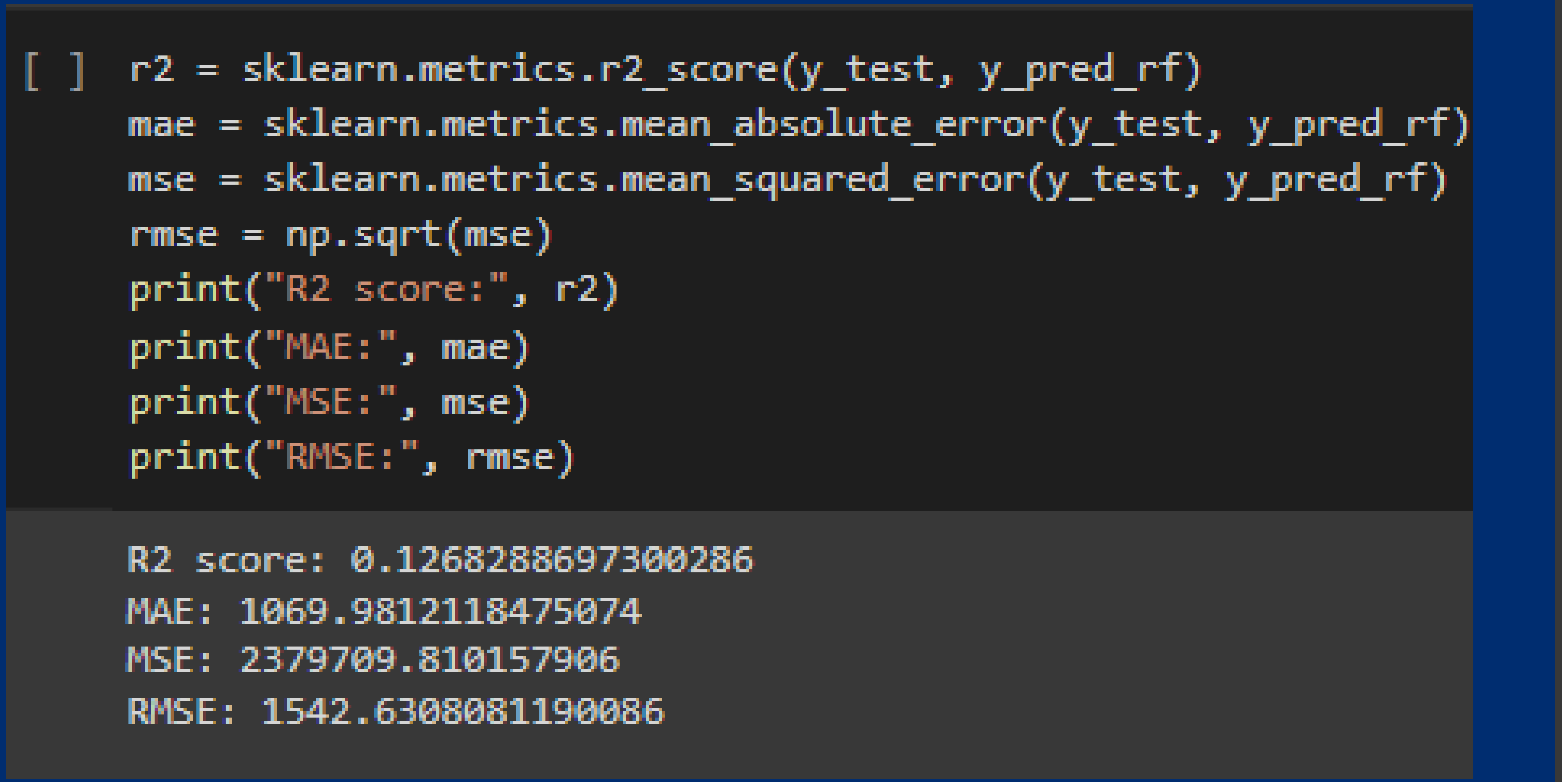
Code to represent and apply the Random Forest regression model



Random Forest Regression

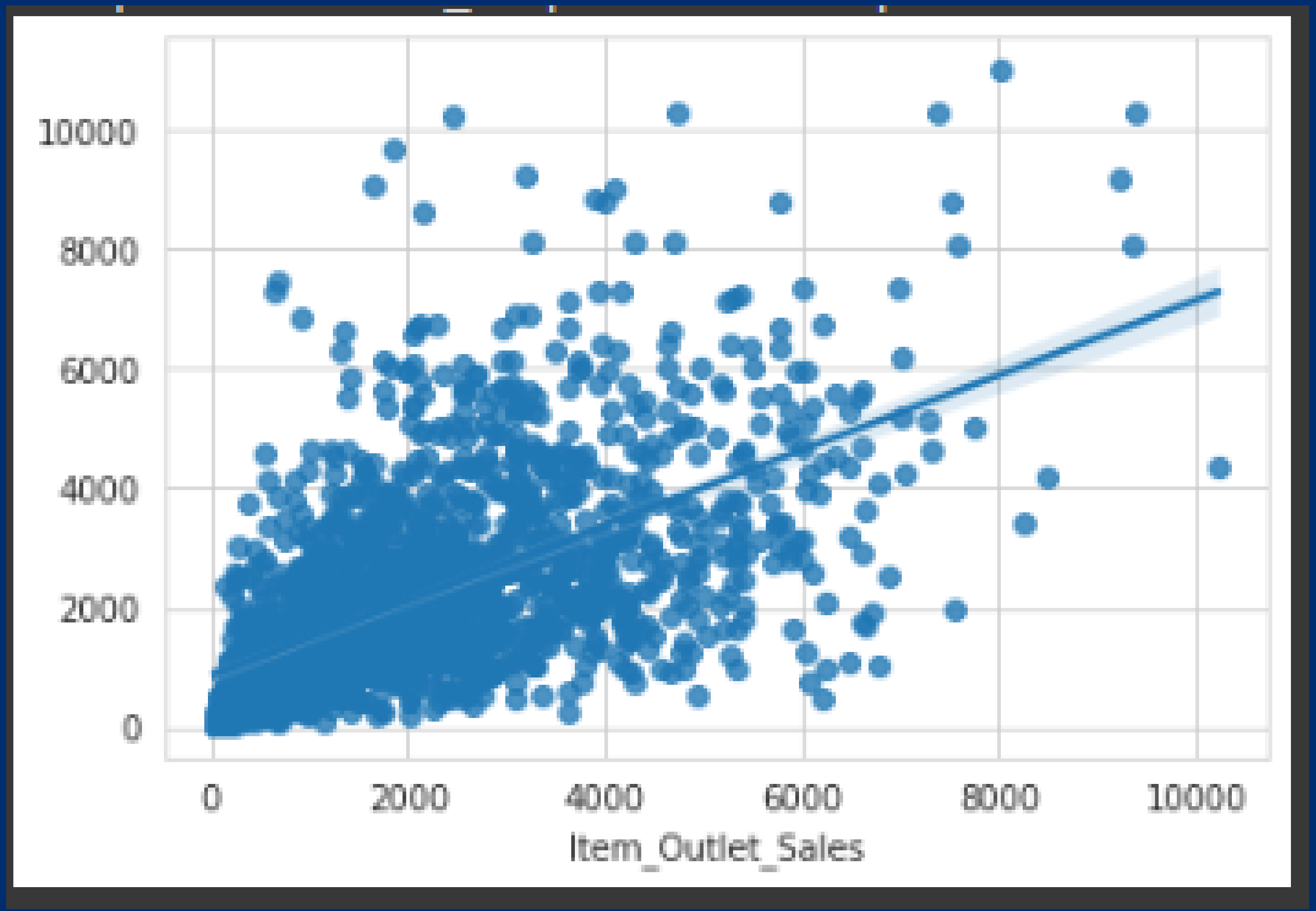
```
[ ]  from sklearn.ensemble import RandomForestRegressor
     rf = RandomForestRegressor()
     dt.fit(X_train, y_train)
     y_pred_rf = dt.predict(X_test)
```

Analysis and Result:



```
[ ]  r2 = sklearn.metrics.r2_score(y_test, y_pred_rf)
     mae = sklearn.metrics.mean_absolute_error(y_test, y_pred_rf)
     mse = sklearn.metrics.mean_squared_error(y_test, y_pred_rf)
     rmse = np.sqrt(mse)
     print("R2 score:", r2)
     print("MAE:", mae)
     print("MSE:", mse)
     print("RMSE:", rmse)
```

```
     R2 score: 0.1268288697300286
     MAE: 1069.9812118475074
     MSE: 2379709.810157906
     RMSE: 1542.6308081190086
```



Final Interpretation And Analysis:

The RMSE(lower the value, better is the model) and R-squared(higher the value, better is the model) value is comparatively better for Linear regression and Knn regression rather than Decision Tree regression and Random Forest Regression, so we will consider these models in this case according to business requirements.

METHODOLOGY 2:

Post applying the previously mentioned models for presiction, we have conducted a hypothesis testing on the data variables/attributes using ANNOVA test.

The variables using which we have conducted ANNOVA test are:

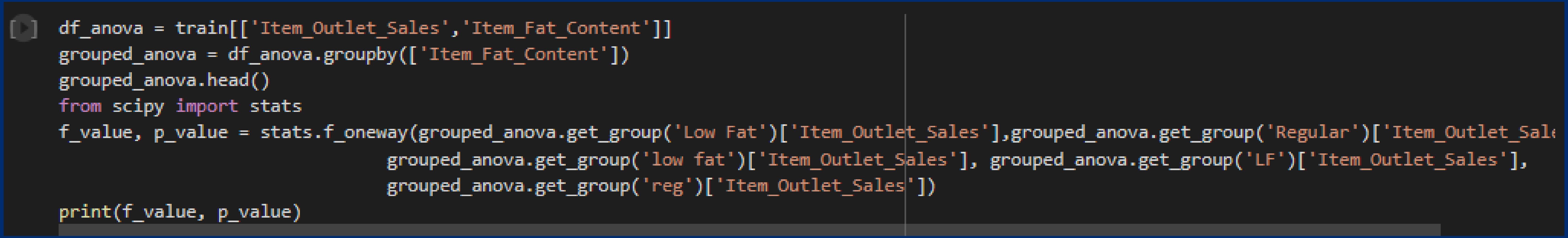
- Item_Outlet_Sales vs Item_Fat_Content
- Item_Outlet_Sales vs Item_Type

Item_Outlet_Sales vs Item_Fat_Content

Test Hypothesis:

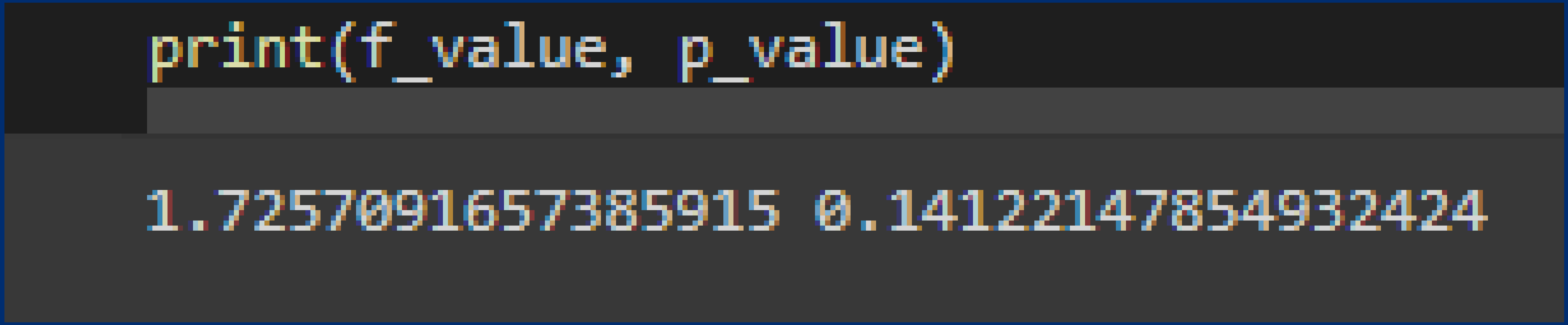
Null Hypothesis - With considerable change in item fat content, the item outlet sales will not differ.

Alernate Hypothesis - With change in item fat content, item outlet sales will change/vary.



```
df_anova = train[['Item_Outlet_Sales','Item_Fat_Content']]
grouped_anova = df_anova.groupby(['Item_Fat_Content'])
grouped_anova.head()
from scipy import stats
f_value, p_value = stats.f_oneway(grouped_anova.get_group('Low Fat')['Item_Outlet_Sales'],grouped_anova.get_group('Regular')['Item_Outlet_Sale
                                  grouped_anova.get_group('low fat')['Item_Outlet_Sales'], grouped_anova.get_group('LF')['Item_Outlet_Sales'],
                                  grouped_anova.get_group('reg')['Item_Outlet_Sales'])
print(f_value, p_value)
```

Test Result and Observation:



```
print(f_value, p_value)
```

```
1.7257091657385915 0.14122147854932424
```

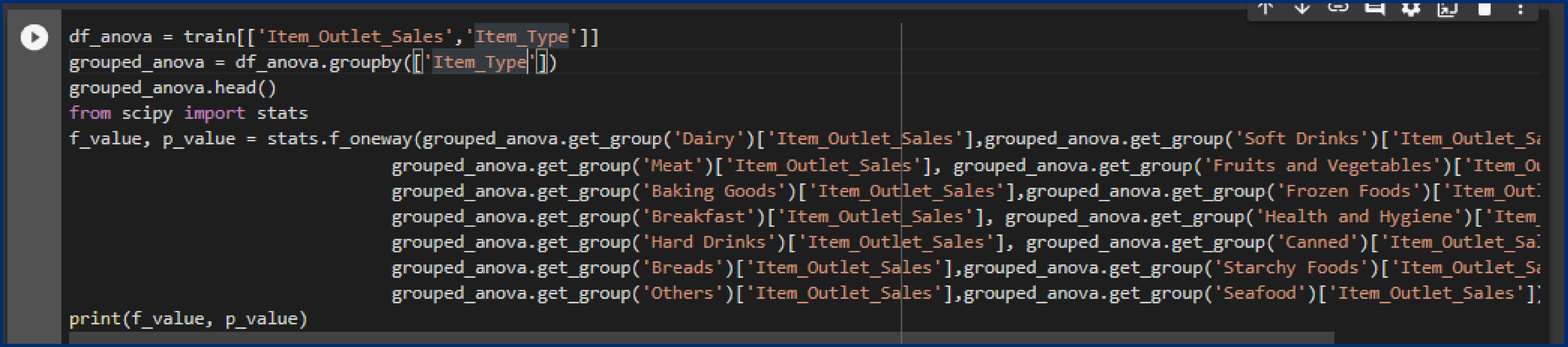
There is significance difference with the Item Fat Content. But there may be significant difference if we treat 'low fat' , 'LF' as 'Low Fat' and 'reg' as 'Regular'. Here we reject the null hypothesis thereby accepting the alternate one.

Item_Outlet_Sales vs Item_Type

Test Hypothesis:

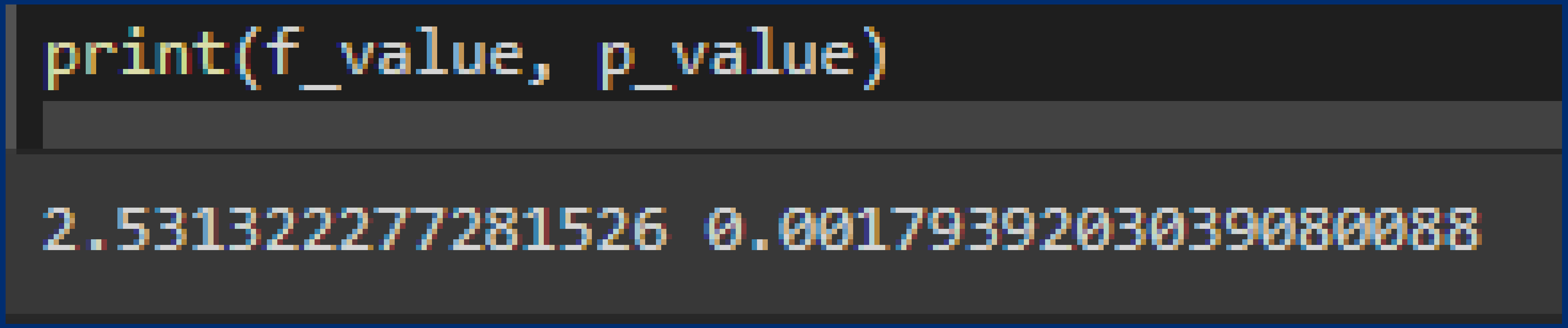
Null Hypothesis - With considerable change in item type, the item outlet sales will not differ.

Alernate Hypothesis - With change in item type, item outlet sales will change/vary.



```
df_anova = train[['Item_Outlet_Sales','Item_Type']]
grouped_anova = df_anova.groupby(['Item_Type'])
grouped_anova.head()
from scipy import stats
f_value, p_value = stats.f_oneway(grouped_anova.get_group('Dairy')['Item_Outlet_Sales'],grouped_anova.get_group('Soft Drinks')['Item_Outlet_Sa
                               grouped_anova.get_group('Meat')['Item_Outlet_Sales'], grouped_anova.get_group('Fruits and Vegetables')['Item_Ou
                               grouped_anova.get_group('Baking Goods')['Item_Outlet_Sales'],grouped_anova.get_group('Frozen Foods')['Item_Out
                               grouped_anova.get_group('Breakfast')['Item_Outlet_Sales'], grouped_anova.get_group('Health and Hygiene')['Item_
                               grouped_anova.get_group('Hard Drinks')['Item_Outlet_Sales'], grouped_anova.get_group('Canned')['Item_Outlet_Sa
                               grouped_anova.get_group('Breads')['Item_Outlet_Sales'],grouped_anova.get_group('Starchy Foods')['Item_Outlet_Sa
                               grouped_anova.get_group('Others')['Item_Outlet_Sales'],grouped_anova.get_group('Seafood')['Item_Outlet_Sales']
print(f_value, p_value)
```

Test Result and Observation:



```
print(f_value, p_value)
```

```
2.531322277281526 0.0017939203039080088
```

There is a significance difference between Item sales of different item types. Dairy products have the higher Item Outlet sales than other product categories. Thus we reject the null hypothesis thereby accepting the alternate one.

Overall Conclusions:

- As per suited business requirements, we can consider choosing Linear and KNN regression models in order to predict the item sales as per their training efficiency.

- Having applied the ANNOVA hypothesis testing, we have inferred that the chosen variables(Item_Outlet_Sales, Item_Fat_Content and Item_Type) have significant difference between each other as the null hypothesis have been rejected for both the test outcomes, implying that each category of the variable differs in accordance to the item sales value.