Phase 6 - Submission

TEAM NAME: Data Digits

Sravani Ganipisetti - sgani2@unh.newhaven.edu

Divya Astalapuram – dasta1@unh.newhaven.edu

Bindu Priya Basa- bbasa4@unh.newhaven.edu

Please introduce your selected data set and research question.

The dataset we are working with is "H&M sales 2018 data". The dataset is taken from Kaggle website (https://www.kaggle.com/datasets/tulasiram574/hm-sales-data). This dataset contains 15 attributes and 100 rows. This dataset is about different products purchased from H&M company in United States and about the profits.

- This dataset has information on products sold by H&M in the year of 2018.
- Each product is represented by a unique ID(Product ID) and Order ID number and includes information such as the Customer Id, Order Date, Sales, Quantity, Discount and Profit.
- And this dataset includes information about the Category and Sub-Category of each product and ordered from which City, State and Region.

RESEARCH QUESTION:

To what extent does the geographic location, encompassing city, state and regional factors, influence the underlying dynamics of product co-purchasing behaviours, as revealed through market basket analysis? Furthermore, how can these discerned geographic variations inform the strategic customization of marketing and sales approaches for diverse locations, ensuring optimal alignment with local consumer preferences and demands?

HARDWARE USED:

- Processor Used: i5
- RAM: 8GB
- Operating System: Windows 11 64-bit
- Tools Used: Jupyter Notebook
- Language: Python

List Of Data Mining Techniques Used:

- Random Forest
- Regression
- Time Series

Visualization Techniques Used:

- Scatter Plots

  Visualizing actual vs predicted values for both Random Forest and Ridge Regression
- Parameter Heatmaps

  Displaying the impact of different hyperparameter values

Random Forest:

- Random Forest is a machine learning algorithm used for classification and regression tasks due to its high accuracy, robustness, feature importance, versatility and scalability. Random Forest reduces overfitting by averaging multiple decision trees and is less sensitive to noise and outliers in the data.

```
X = new_data.drop('Profit', axis=1)
y = new_data['Profit']

X = X.select_dtypes(exclude=['object'])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
grid_search_rf = GridSearchCV(RandomForestRegressor(random_state=42), param_grid_rf, cv=5, scoring='neg_mean_squared_error')
grid_search_rf.fit(X_train, y_train)

best_model_rf = grid_search_rf.best_estimator_
print("Best Parameters for Random Forest:", grid_search_rf.best_params_)

best_model_rf.fit(X_train, y_train)
y_pred_rf = best_model_rf.predict(X_test)

mse_rf = mean_squared_error(y_test, y_pred_rf)
mae_rf = mean_absolute_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

print("Random Forest Regression Results:")
print("Mean Squared Error:", mse_rf)
print("Mean Absolute Error:", mae_rf)
print("R-squared (R2) Score:", r2_rf)
```

The optimization techniques used in this code are StandardScaler() and GridSearchCV. Standardization ensures that features have mean of 0 and standard deviation of 1, which can be important for certain machine learning algorithms.

Regression:

- Regression is a machine learning technique used for investigating the relationship between independent variables or features and a dependent variable or outcome. It is used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcomes.

```
le = LabelEncoder()
new_data_cleaned['Category'] = le.fit_transform(new_data_cleaned['Category'])
X_ridge = new_data_cleaned[['Category', 'Quantity']]
y_ridge = new_data_cleaned['Profit']
X_train_ridge, X_test_ridge, y_train_ridge, y_test_ridge = train_test_split(X_ridge, y_ridge, test

warnings.filterwarnings("ignore", category=DeprecationWarning)
scaler_ridge = StandardScaler()
X_train_scaled_ridge = scaler_ridge.fit_transform(X_train_ridge)
X_test_scaled_ridge = scaler_ridge.transform(X_test_ridge)

param_grid_ridge = {
    'alpha': [0.001, 0.01, 0.1, 1, 10]
}

grid_search_ridge = GridSearchCV(Ridge(), param_grid_ridge, cv=5, scoring='neg_mean_squared_error'
grid_search_ridge.fit(X_train_scaled_ridge, y_train_ridge)

best_model_ridge = grid_search_ridge.best_estimator_
print("Best Parameters for Ridge Regression:", grid_search_ridge.best_params_)

ridge_predictions = best_model_ridge.predict(X_test_scaled_ridge)
ridge_mse = mean_squared_error(y_test_ridge, ridge_predictions)
ridge_r2 = r2_score(y_test_ridge, ridge_predictions)
```

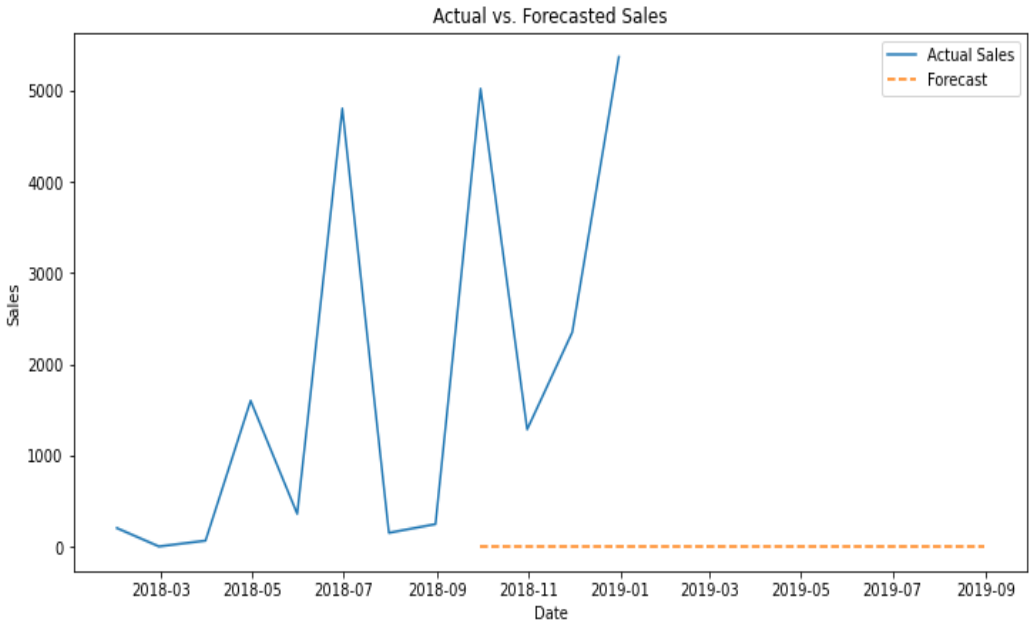
The optimization techniques used in this code are Label encoding (LabelEncoder()) which helps the given data for machine learning algorithms by converting into numerical values and ensuring that are in consistent scale. The parameters that is optimized in Ridge Regression is Alpha.

Time Series Analysis:

- Time series analysis is a statistical technique used to analyze and make predictions about data that is collected over a period of time.

```
time_series_data = new_data_cleaned[['Order Date', 'Sales', 'Profit']].copy()
time_series_data['Order Date'] = pd.to_datetime(time_series_data['Order Date'])
time_series_data.set_index('Order Date', inplace=True)

model = ExponentialSmoothing(time_series_data['Sales'], trend='add', seasonal='add', seasonal_peri
model_fit = model.fit()

forecast = model_fit.forecast(steps=12)
forecast = pd.Series(forecast, index=pd.date_range(start=time_series_data.index[-1], periods=12,

time_series_data = time_series_data.resample('M').sum()
forecast = forecast.resample('M').sum()

mse_time_series = mean_squared_error(time_series_data['Sales'], forecast)
r2_time_series = r2_score(time_series_data['Sales'], forecast)

print("Time Series Forecasting Results:")
print("Time Series MSE:", mse_time_series)
print("Time Series R2 Score:", r2_time_series)

plt.figure(figsize=(12, 6))
plt.plot(time_series_data.index, time_series_data['Sales'], label='Actual Sales')
plt.plot(forecast.index, forecast, label='Forecast', linestyle='--')
plt.title('Actual vs. Forecasted Sales')
```

To prepare the time series data, the relevant columns ('Order Date,' 'Sales,' and 'Profit') are retrieved. 'Order Date' is then converted to datetime format and set as the index of the DataFrame. This ensures that the format of the data is appropriate for time series analysis.
The objective of optimization is to generate precise forecasts, and the model's performance is evaluated using evaluation criteria.



Association Rules:

Association Rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures.

```
df_association = new_data_cleaned[['Order ID', 'Sub-Category']]

basket = df_association.groupby('Order ID')['Sub-Category'].apply(list).reset_index()

basket = basket.set_index('Order ID')['Sub-Category'].str.join('|').str.get_dummies().reset_index()

min_support = 0.005
frequent_itemsets = apriori(basket.drop('Order ID', axis=1), min_support=min_support, use_colnames=True)

rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)

print("Association Rule Mining Results:")
print(rules)

plt.scatter(rules['support'], rules['confidence'], alpha=0.5)
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.title('Association Rules - Support vs. Confidence')
plt.show()
```

Association Rule Mining Results:

	antecedents	consequents
0	(Flip flops)	(Bags)
1	(Bags)	(Flip flops)
2	(Heels & Flats)	(Bags)
3	(Bags)	(Heels & Flats)
4	(Jackets)	(Bags)
...	...	...
1495	(Belts)	(Tops, Dresses, Jackets, Sport shoes, Formals)
1496	(Dresses)	(Tops, Belts, Jackets, Sport shoes, Formals)
1497	(Jackets)	(Belts, Dresses, Tops, Sport shoes, Formals)
1498	(Sport shoes)	(Tops, Belts, Dresses, Jackets, Formals)
1499	(Formals)	(Tops, Belts, Dresses, Jackets, Sport shoes)

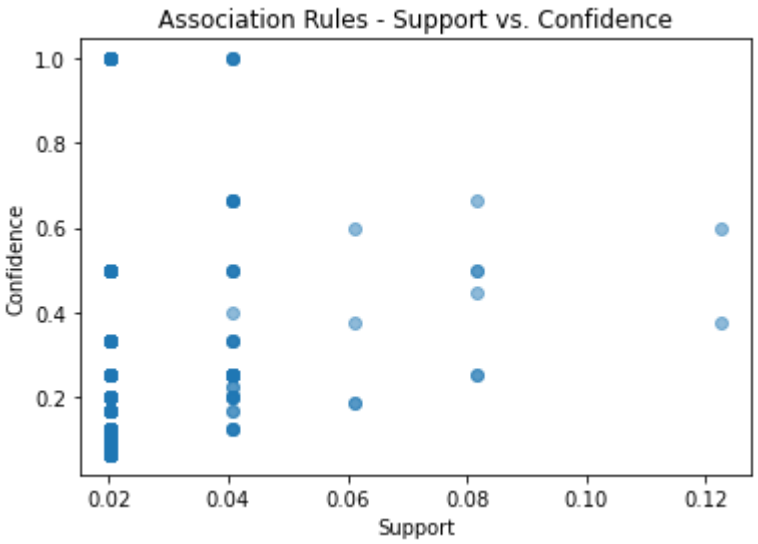
	antecedent support	consequent support	support	confidence	lift
0	0.061224	0.183673	0.020408	0.333333	1.814815
1	0.183673	0.061224	0.020408	0.111111	1.814815
2	0.061224	0.183673	0.020408	0.333333	1.814815
3	0.183673	0.061224	0.020408	0.111111	1.814815
4	0.163265	0.183673	0.081633	0.500000	2.722222
...	...	...	...	...	...
1495	0.122449	0.020408	0.020408	0.166667	8.166667
1496	0.326531	0.020408	0.020408	0.062500	3.062500
1497	0.163265	0.020408	0.020408	0.125000	6.125000
1498	0.163265	0.020408	0.020408	0.125000	6.125000

...

1498 0.017076 1.119534 1.000000

1499 0.017076 1.119534 1.000000

[1500 rows x 10 columns]



Techniques Used in Optimization:

Technique Used	Parameters	Hyperparameters	Impact
Random Forest	'n_estimators', 'max_depth', 'min_samples_split', 'min_samples_leaf', 'max_features'	param_grid_rf	These parameters and Hyperparameters affects the performance, complexity of Random Forest Model.
Ridge Regression	'alpha': [0.001, 0.01, 0.1, 1, 10]	param_grid_ridge	By choosing of the alpha the trade-off between accurately fitting the training data and keeping small model coefficients to prevent overfitting is strongly impacted parameter.
Time Series	trend' , 'seasonal', ' seasonal_periods'		The selection of seasonal components and additive trend suggests that both linear trend and seasonality should be included in the time series data.
Association Rules	min_support'		The choice of the min_support parameter influences the number and significance of the discovered association rules

Conclusion:

In machine learning, optimization refers to determining the ideal combination of parameters, or hyperparameters, to enhance model performance. Different techniques require different optimization approaches. Optimization techniques improved the accuracy and accessibility of the models. The effectiveness of each optimization technique was assessed using specific evaluation metrics relevant to the task e.g., MSE, R-squared for regression; support, confidence for association rule mining.

In the end the models we used provides valuable information into factors influencing profit. These insights can be used to customize sales and marketing strategies according to geographic variations and local customer preferences. We split the dataset into 80% training set and 20% testing set to assess the performance.

Github Link:

https://github.com/Sravani1698/Data-Digits-Phase_6.git