Data Analysis and Forecasting for Online Retail

Description:

In today's dynamic e-commerce landscape, making informed decisions requires both a deep understanding of historical business performance and the ability to anticipate future trends. This project brings together two fundamental pillars of retail data analytics:

1. **Exploratory Data Analysis (EDA)** to uncover patterns, segments, and opportunities for improvement in historical and current data, and
2. **Sales Forecasting** using ARIMA models to project future demand and support strategic planning.

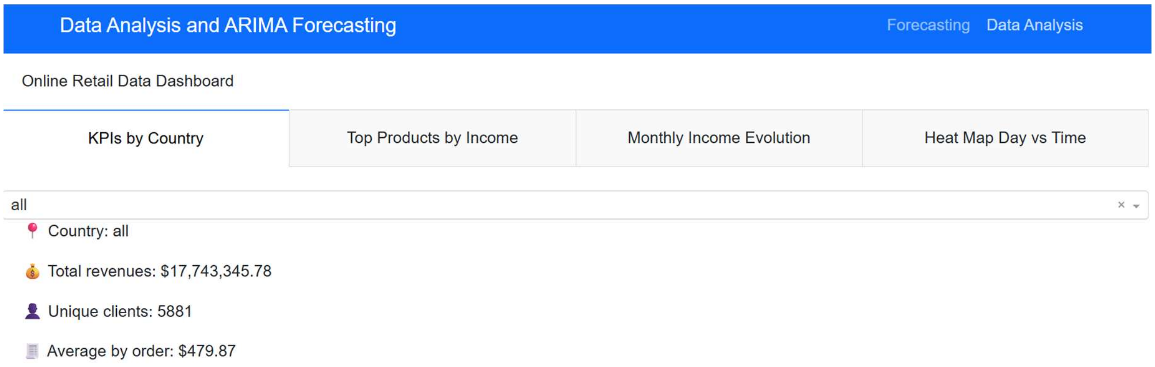
Objective:

The main goal was to build a comprehensive solution for visualizing, analyzing, and forecasting the sales behavior of an online store. To achieve this, we combined Big Data tools (Spark, Google BigQuery), advanced analytics (Python, statsmodels), and interactive visualization (Dash).

1. Exploratory Data Analysis (EDA)

The first step involved cleaning, transforming, and enriching historical data on sales, customers, and products, stored in Google BigQuery and processed with Spark. An interactive Dash dashboard was developed to enable:

- Visualization of total revenue, unique customers, and average ticket size by country and period.



all

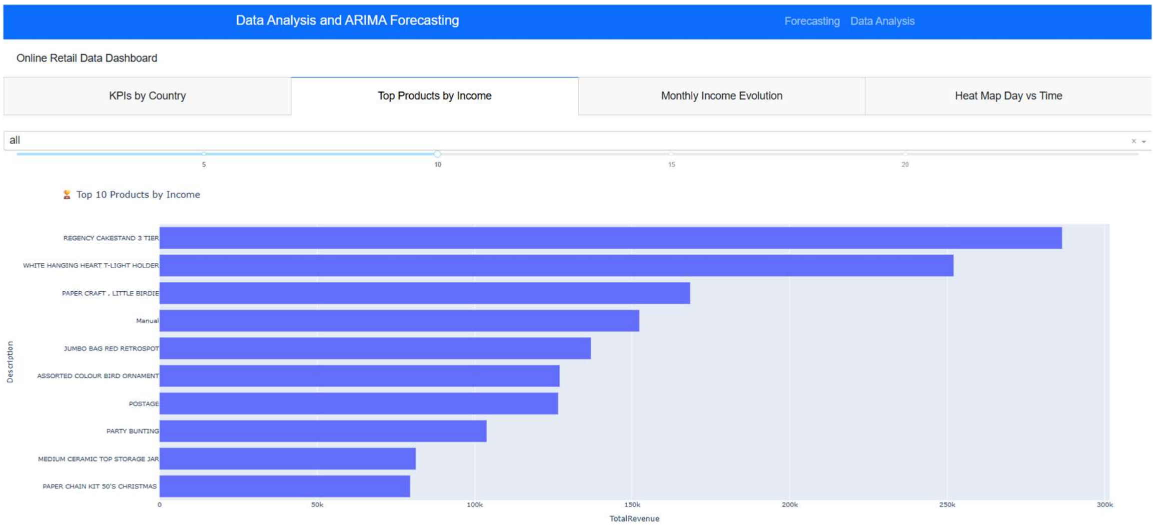
📍 Country: all

💰 Total revenues: $17,743,345.78

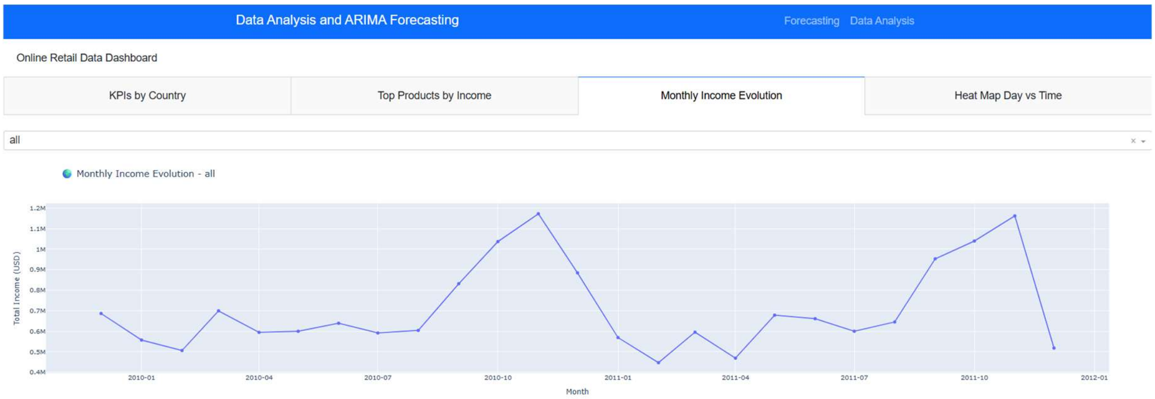
👤 Unique clients: 5881

🧾 Average by order: $479.87

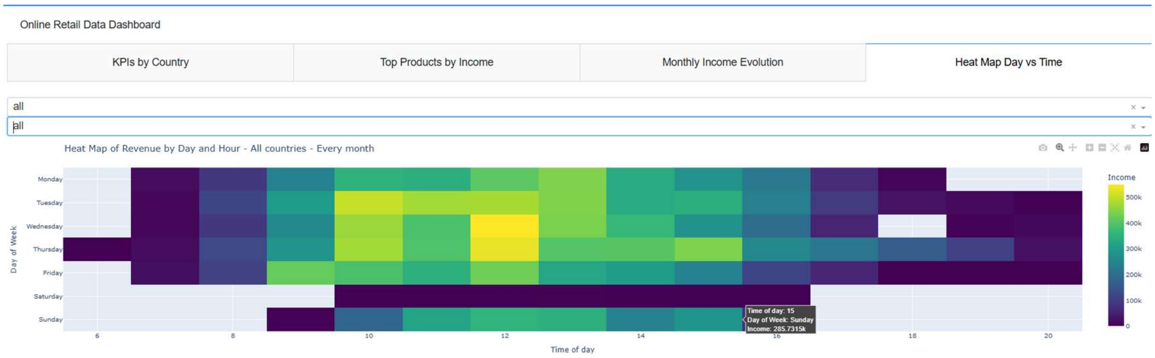
- Identification of best-selling products and top revenue generators, filtering and segmentation of information by country.


Data Analysis and ARIMA Forecasting
Forecasting
Data Analysis
Online Retail Data Dashboard
KPIs by Country
Top Products by Income
Monthly Income Evolution
Heat Map Day vs Time
all
5
10
15
20
Top 10 Products by Income
REGENCY CAKESTAND 3 TIER
WHITE HANGING HEART T-LIGHT HOLDER
PAPER CRAFT , LITTLE BIRDIE
Manual
JUMBO BAG RED RETROSPOT
ASSORTED COLOUR BIRD ORNAMENT
POSTAGE
PARTY BUNTING
MEDIUM CERAMIC TOP STORAGE JAR
PAPER CHAIN KIT 50'S CHRISTMAS
Description
0
50k
100k
150k
200k
250k
300k
TotalRevenue

- Analysis of monthly sales evolution to detect trends and seasonality, filtering and segmentation of information by country.



- Exploration of purchase patterns by weekday and hour using heat maps.



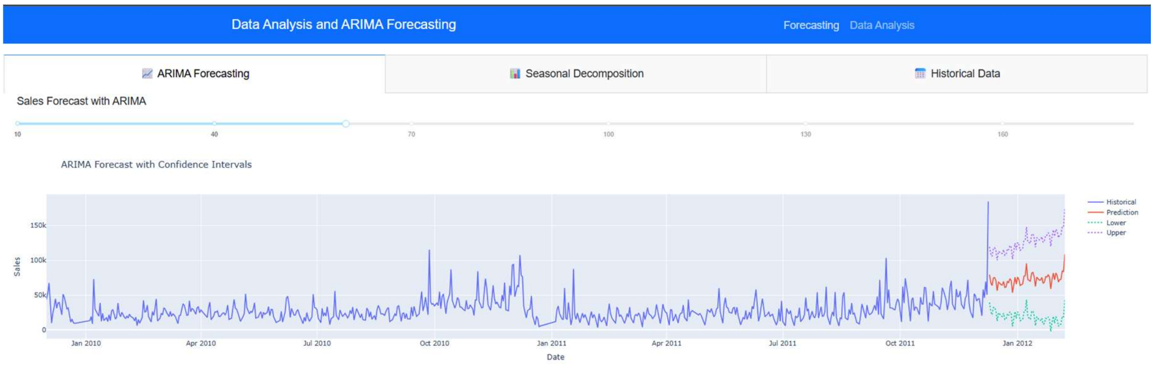
Filtering and segmentation of information by country and month, facilitating both operational and strategic decision-making.

This analysis helped identify growth opportunities, optimize promotions, and gain a deeper understanding of customer behavior, laying the groundwork for more accurate forecasting.

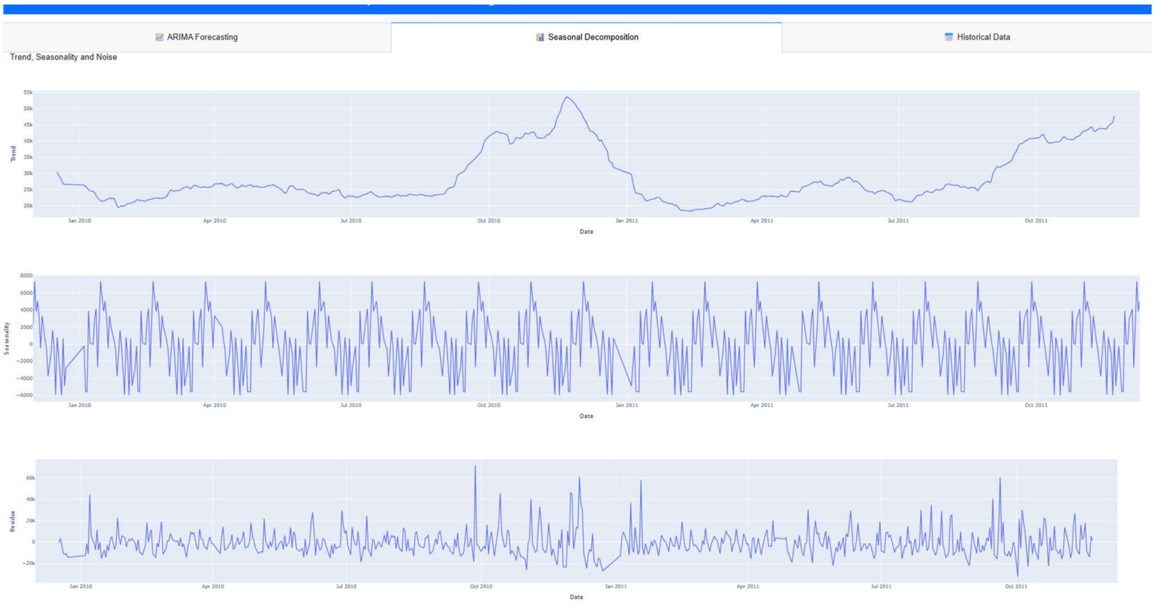
2. Sales Forecasting with ARIMA

Building on the preprocessed data, an ARIMA model was implemented to forecast daily sales. The model was tuned and evaluated, taking into account the seasonality and trends identified during the EDA. The dashboard allows users to:

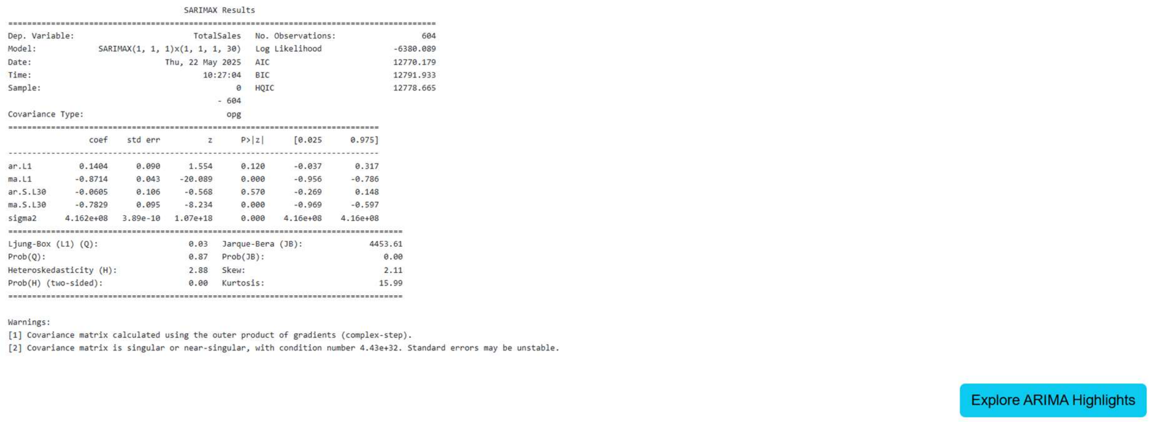
- Visualize the historical sales series alongside forecasts for upcoming months, adjusting the prediction horizon interactively.



- Review model metrics and diagnostics (parameter significance, residual tests, etc.).

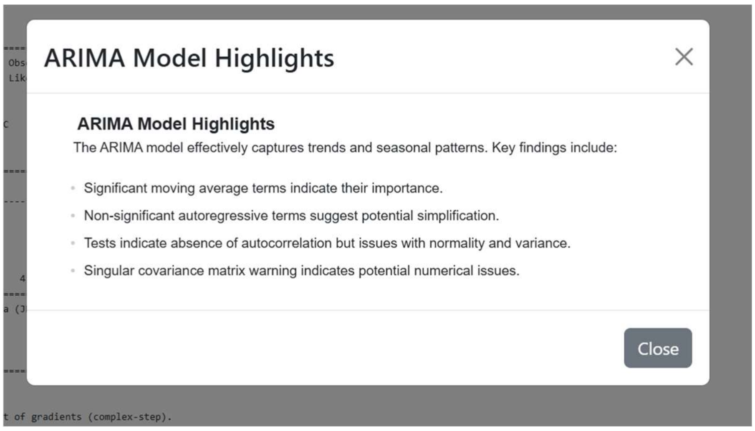


- Interpret results and limitations, highlighting areas for further improvement.



```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                       TotalSales   No. Observations:                  604
Model:             SARIMAX(1, 1, 1)x(1, 1, 1, 30)   Log Likelihood               -6380.089
Date:                          Thu, 22 May 2025   AIC                          12770.179
Time:                                  10:27:04   BIC                          12791.933
Sample:                                       0   HQIC                         12778.665
                                          - 604
Covariance Type:                            opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1404      0.090      1.554      0.120      -0.037       0.317
ma.L1         -0.8714      0.043    -20.089      0.000      -0.956      -0.786
ar.S.L30      -0.0605      0.106     -0.568      0.570      -0.269       0.148
ma.S.L30      -0.7829      0.095     -8.234      0.000      -0.969      -0.597
sigma2      4.162e+08   3.89e-10   1.07e+18      0.000    4.16e+08    4.16e+08
===================================================================================
Ljung-Box (L1) (Q):                   0.03   Jarque-Bera (JB):              4453.61
Prob(Q):                              0.87   Prob(JB):                         0.00
Heteroskedasticity (H):               2.88   Skew:                             2.11
Prob(H) (two-sided):                  0.00   Kurtosis:                        15.99
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 4.43e+32. Standard errors may be unstable.
```



ARIMA Model Highlights

ARIMA Model Highlights

The ARIMA model effectively captures trends and seasonal patterns. Key findings include:

- Significant moving average terms indicate their importance.
- Non-significant autoregressive terms suggest potential simplification.
- Tests indicate absence of autocorrelation but issues with normality and variance.
- Singular covariance matrix warning indicates potential numerical issues.

The forecasting tool provides key insights for inventory planning, marketing campaigns, and supply chain management.

Key Technical Details:

This project integrated several key tools and libraries:

- pandas: Fundamental library for data analysis in Python. Allows you to manipulate structures such as DataFrames and Series, facilitating the handling of tabular data.
- numpy: Provides efficient structures and functions for numerical calculations, including multidimensional arrays and vector operations.
- dash: Python framework for building interactive web applications focused on data visualization. It is ideal for building dashboards with dynamic components.
- plotly: Interactive graphics library that integrates with Dash to represent data in formats such as lines, bars, maps, and more, in a visually appealing way.

- pyspark: Apache Spark API for Python. Enables distributed processing of large volumes of data and integrates with BigQuery through specific connectors.
- google-cloud-bigquery: Official Google Cloud client for connecting to BigQuery, Google's cloud-based data analytics solution. Facilitates SQL queries on large datasets.
- google-cloud-storage: Official client for interacting with Google Cloud Storage, used to store files in the cloud, such as CSV datasets or backups.
- google-auth: Library that manages authentication and authorization with Google services, including credential management using key.json.
- google-auth-oauthlib: Extends google-auth with OAuth 2.0 capabilities, useful for browser-based authentication in web applications.
- google-cloud-core: Contains common and essential components used by Google Cloud clients, such as transport configuration, credential management, and version control.
- google-auth-httplib2: Google-auth extension that allows authenticating HTTP requests using httplib2. Facilitates authenticated API calls.
- dash-bootstrap-components: Dash plugin that allows you to use Bootstrap framework styles and components (buttons, tabs, cards) to enhance your interface design.
- scikit-learn: Key library for machine learning. Provides regression, classification, and clustering algorithms, as well as validation and preprocessing tools.

How to Run the Project:

- **Prerequisites**
  Python 3.x, Anaconda/Miniconda recommended
- **How to clone the repository.**
  - **Create and activate the virtual environment.**

    e.g. *conda create -n my_arima_environment python=3.9*

    conda activate my_arima_environment.
  - **Upload the database to Google BigQuery (GCP).**

    (If you don't have an account, you can use the free tier)

    Steps:

Create a project in Google Cloud.

Activate BigQuery and create a dataset (e.g., ecommerce_dw).

Upload the online_retail_II.csv table using the dashboard or via Python (google-cloud-bigquery).

- **Create the Google Cloud credentials file.**

  In your Google Cloud account, create an authentication file (.json) for the project you created where downloaded the dataset in the previous step.

  The user must name it/place it in a specific path (in your project's root directory), or the run_dashboard.bat script will prompt you for it).

  Remember NOT to upload your own credentials if you fork.

- **Install the dependencies.**

  run run_dashboard.bat which already does this, or pip install -r requirements.txt manually.

- **run the application.**

  using run_dashboard.bat or python main.py (--key_path PATH_TO_YOUR_KEY --project_id YOUR_PROJECT_ID).