H.P.K.G. Pathirana

E222739

**Mr. W. A. D. B. C. Goonatillaka**

Advanced data modeling

Course Work 2

**Kingston University, BSc (Hons) (top-up)**

**CI6320 - Advanced Data Modeming**

**Weight of this assessment is 25% of the final grade.**

**Coursework Cover Sheet**

**Part 1 - To Remain with the Assignment after Marking**

|  |  |
| --- | --- |
| **Student ID: E222739** | **Student Name: H.P.K.G. Pathirana** |
| **Module Code: CI6320** | **Module Name: Advance Data Modeling** |
| **Assignment number:** | **ESoft Module Leader: Mr. W A D B C Goonatillaka** |
| **Date set:** | **Date due: 28th of April 2024** |

**Guidelines for the Submission of Coursework**

1. Print this cover sheet and securely attach both pages to your assignment. You can help us ensure work is marked more quickly by submitting at the specified location for your module. You are advised to keep a copy of every assignment.

2. Coursework deadlines are strictly enforced by the University.

3. You should not leave the handing in of work until the last minute. Once an assignment has been submitted it cannot be submitted again.

**Academic Misconduct**: **Plagiarism** and/or **collusion** constitute **academic misconduct** under the University's Academic Regulations. Examples of academic misconduct in coursework: making available your work to other students; presenting work produced in collaboration with other students as your own (unless an explicit assessment requirement); submitting work, taken from sources that are not properly referenced, as your own. By printing and submitting this coversheet with your coursework you are confirming that the work is your own.

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | ESoft Office Use Only:  Date stamp: work received | | |  | | --- | |  | |

**Kingston University, BSc (Hons) (top-up)**

**Coursework Cover Sheet**

**Part 2 – Student Feedback**

|  |  |
| --- | --- |
| **Student ID: E222739** | **Student Name: H.P.K.G Pathirana** |
| **Module Code: CI6320** | **Module Name: Advance Data Modeling** |
| **Assignment number:** | **ESoft Module Leader: Mr. W A D B C Goonatillaka** |
| **Date set:** | **Date due:** |

|  |
| --- |
| Strengths (areas with well-developed answers) |

|  |
| --- |
| Weaknesses (areas with room for improvement) |

|  |
| --- |
| Additional Comments |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ESoft Module Lecturer:** | **Provisional mark as %:** |  |
| **ESoft Module Marker:** | **Date marked:** |

**Section A: Revealing Dimensionality Reduction in Data Mining**

Dimensionality reduction in data mining is a crucial process that involves reducing the number of random variables under consideration and can be divided into feature selection and feature extraction. The key objectives of dimensionality reduction include simplification of models to make them easier to interpret, reduction of storage space, and speeding up computation. Additionally, it helps in mitigating the curse of dimensionality, which can cause models to overfit and perform poorly on unseen data.

### Benefits of Dimensionality Reduction,

* Simplification: High-dimensional data can be complex and challenging to work with. Reducing dimensions simplifies the data representation.
* Computational Efficiency: Fewer features mean faster computation, especially for machine learning algorithms.
* Visualization: Lower-dimensional data is easier to visualize, aiding exploratory data analysis.
* Noise Reduction: By focusing on essential features, we reduce the impact of noisy or irrelevant attributes.
* Generalization: Reduced dimensionality can improve model generalization and prevent overfitting.

### Challenges of High Dimensionality,

1. Curse of Dimensionality: As the number of features increases, the data becomes sparse, and the volume of the feature space grows exponentially. This makes it harder to find meaningful patterns.
2. Computational Complexity: High-dimensional data requires more computational resources for processing.
3. Overfitting: Models can overfit due to the abundance of features.
4. Data Sparsity: High-dimensional data points are often scattered and isolated.
5. Visualization Difficulty: Visualizing high-dimensional data is challenging.

### Real-World Applications,

1. Image Recognition:
   * PCA and t-SNE are used to reduce image feature dimensions for better visualization and clustering.
   * LDA helps improve classification accuracy by separating classes.
2. Natural Language Processing (NLP):
   * Word embeddings (e.g., Word2Vec, GloVe) reduce word vectors’ dimensions while preserving semantic relationships.
3. Bioinformatics:
   * Gene expression analysis: Dimensionality reduction helps identify relevant genes and pathways.
   * Protein structure prediction: Techniques like PCA and t-SNE aid in visualizing protein structures.
4. Recommendation Systems:
   * Collaborative filtering: Dimensionality reduction enhances recommendation accuracy.
5. Financial Data Analysis:
   * Portfolio optimization: Reduced dimensions improve risk assessment and asset allocation.

### Analysis of Dimensionality Reduction Methods,

* Principal Component Analysis (PCA): PCA identifies the axes (principal components) along which the variance in the data is maximized. It projects the data onto these new axes to reduce dimensions while retaining most of the variance.
* Incremental PCA: Incremental PCA is a variant of PCA that is suitable for large datasets. It processes the data in mini-batches and updates the PCA decomposition incrementally, which is more memory-efficient.
* Kernel PCA: Kernel PCA extends PCA to nonlinear dimensionality reduction. It uses kernel functions to project data into a higher-dimensional space where linear separation is possible, then applies PCA in this new space.
* Sparse PCA: Sparse PCA is a modified version of PCA that leads to principal components with sparse loadings, which are easier to interpret. It encourages sparsity by adding a regularization term to the PCA optimization problem.
* Truncated SVD (Singular Value Decomposition): Truncated SVD works similarly to PCA but is applicable to sparse datasets. It decomposes the data into singular values and vectors, keeping only the top components.
* Gaussian Random Projection: This technique reduces dimensionality by projecting the data onto a lower-dimensional subspace using a random Gaussian matrix. It relies on the Johnson-Lindenstrauss lemma to preserve distances within the data.
* Linear Discriminant Analysis (LDA): LDA aims to find a projection that maximizes the separation between multiple classes. It is particularly useful for supervised dimensionality reduction, where class labels are known.
* Neighborhood Components Analysis (NCA): NCA is a non-parametric method that selects features to maximize the accuracy of k-nearest neighbors classification. It learns a distance metric that improves the classification performance.
* Sparse Random Projection: Similar to Gaussian Random Projection but uses a sparse matrix instead. It’s more memory-efficient and faster for computation, especially for high-dimensional data.
* Isomap: Isomap is a manifold learning technique that preserves geodesic distances between all points. It’s particularly good at unfolding manifolds to reveal the intrinsic geometry of the data.
* Mini-Batch Dictionary Learning: This method learns a sparse representation of the data in a dictionary form. It processes the data in small batches, which makes it suitable for large datasets.
* FastICA (Independent Component Analysis): FastICA separates a multivariate signal into additive subcomponents that are maximally independent. It’s often used for blind source separation.
* Locally Linear Embedding (LLE): LLE computes low-dimensional, neighborhood-preserving embeddings of high-dimensional data. It assumes each data point can be linearly reconstructed from its neighbors.

These methods employ various underlying mechanisms to transform high-dimensional data into lower-dimensional spaces, such as linear projections, kernel methods, manifold learning, and sparse coding.

### Comparison of the Three Best Dimensionality Reduction Methods for the Provided Scenario,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Suitability** | **Information Preservation** | **Computational Complexity** | **Use Cases** |
| PCA | Linear | High (Captures Variance) | Low | Visualization, Feature Extraction |
| LDA | Linear | Moderate (Class Separation) | Low | Classification, Feature Extraction |
| NCA | Supervised | Moderate (Nearest Neighbor Accuracy) | Moderate | Classification, Clustering |

### Python Code Explanation,

The provided code follows these steps:

1. Import necessary libraries and datasets.
2. Load the Digits dataset from scikit-learn.
3. Split the data into training and test sets.
4. Define a dictionary of dimensionality reduction methods (e.g., PCA, LDA, Kernel PCA) and instantiate them with appropriate hyperparameters.
5. For each dimensionality reduction method:
   1. Fit the method on the training data.
   2. Transform the training and test data using the fitted method.
   3. Train a k-Nearest Neighbors (KNN) classifier on the transformed training data.
   4. Evaluate the KNN classifier's accuracy on the transformed test data.
   5. Visualize the transformed data in 2D, color-coded by the digit labels.

### Hyperparameters,

* n\_components: The number of components to keep after the reduction.
* kernel: The type of kernel used in Kernel PCA.
* n\_neighbors: The number of neighbors to consider in manifold learning methods.

These hyperparameters significantly impact the performance and outcome of the dimensionality reduction process. For instance, choosing a higher n\_components in PCA might preserve more information but at the cost of a less significant reduction in dimensionality.

### Modification Suggestions,

1. Implementing Additional Dimensionality Reduction Techniques:
   * t-SNE (t-Distributed Stochastic Neighbor Embedding): A powerful non-linear technique that is particularly good at preserving local structure and revealing clusters at many scales.
   * UMAP (Uniform Manifold Approximation and Projection): Similar to t-SNE but often faster and better at preserving global structure.
   * Factor Analysis: Useful when you suspect there are latent factors causing the observed variances.
   * Random Forests: Can be used for feature selection to identify important features and reduce dimensionality indirectly.
2. Experimenting with Different Hyperparameter Values:
   * n\_components: Vary this parameter to see how the number of dimensions affects the performance and the quality of the visualization.
   * kernel in KernelPCA: Try different kernels like poly, rbf, or sigmoid to see how they capture non-linear relationships.
   * n\_neighbors in manifold learning methods: Adjusting this can change the emphasis on local vs. global structure in the data.
3. Enhanced Analysis:
   * Cross-validation: Implement cross-validation to assess the stability of the dimensionality reduction across different subsets of data.
   * Pipeline Optimization: Use tools like GridSearchCV to find the best combination of preprocessing steps and hyperparameters.

### Impact of Modifications,

* t-SNE and UMAP: These methods could reveal more nuanced clustering patterns in the data, which might not be apparent with linear methods like PCA.
* Hyperparameter Tuning: This can lead to a more optimized model that captures the underlying structure of the data more effectively.
* Factor Analysis and Random Forests: These could provide insights into the nature of the dataset, such as identifying latent variables or the most informative pixels in the image data.

**Section B: Web Scraping and Movie Analytics**

IMDb (Internet Movie Database) is a highly suitable for web scraping movie data because,

It offers extensive information on movies, including titles, genres, directors, release years, ratings, and more.

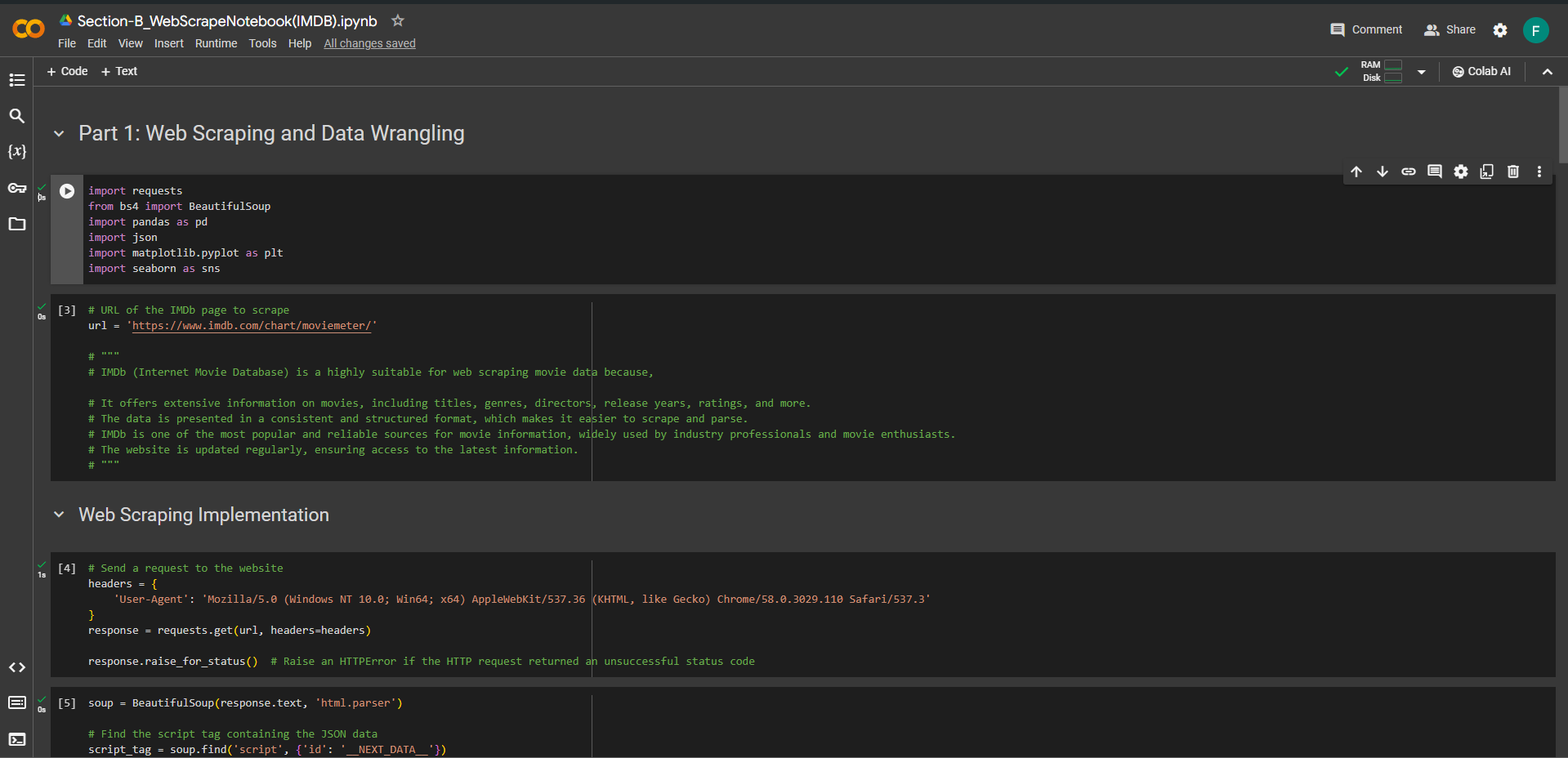
The data is presented in a consistent and structured format, which makes it easier to scrape and parse.

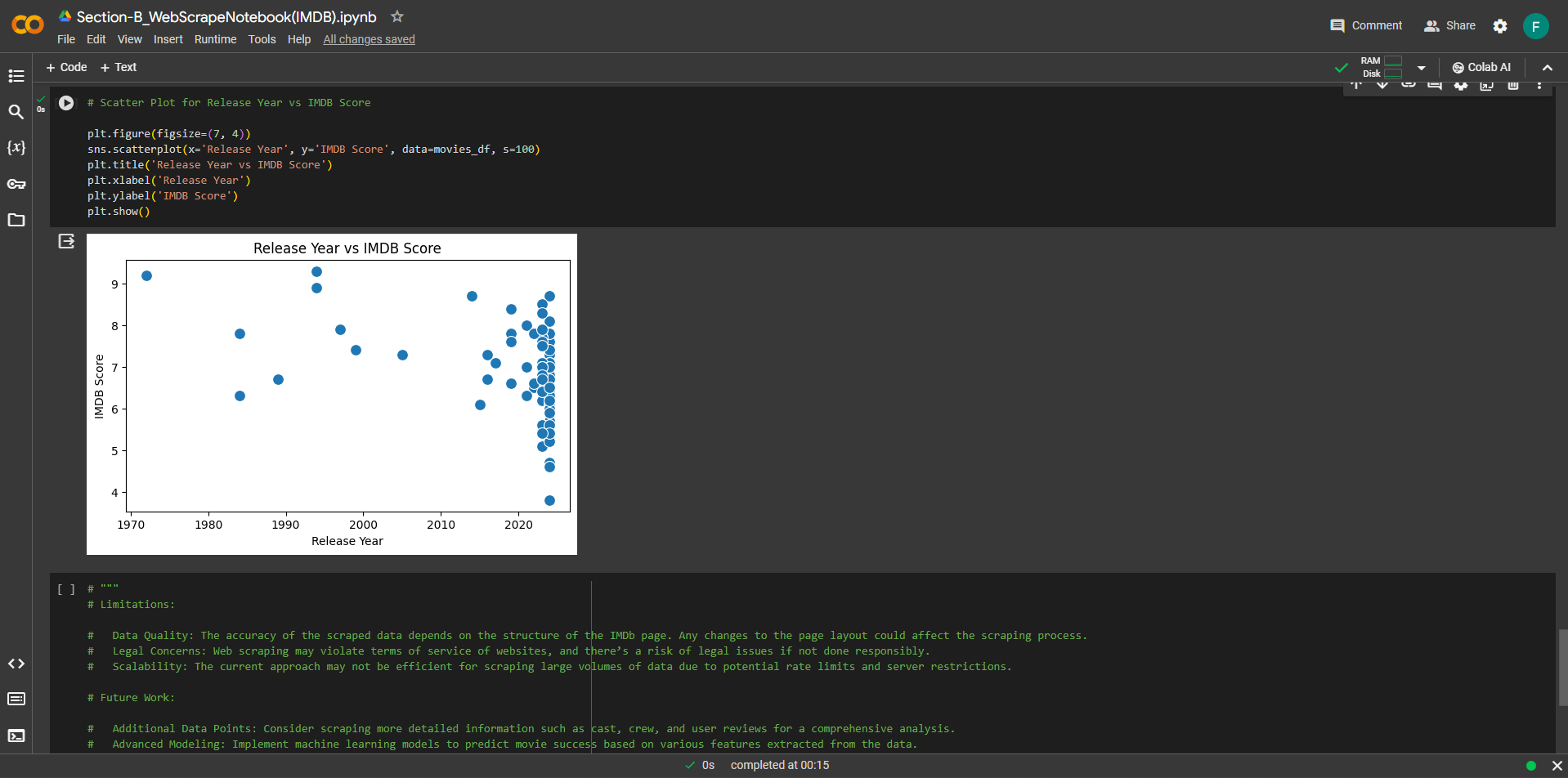
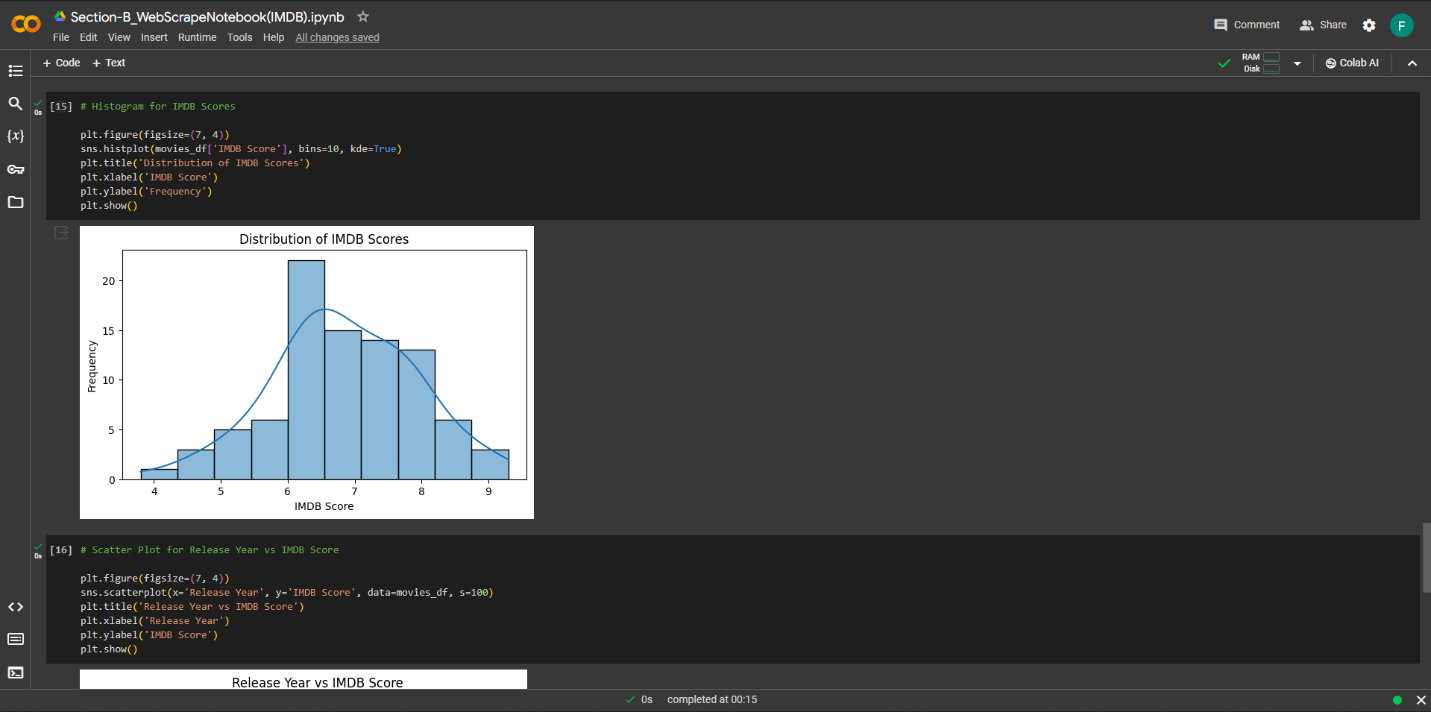
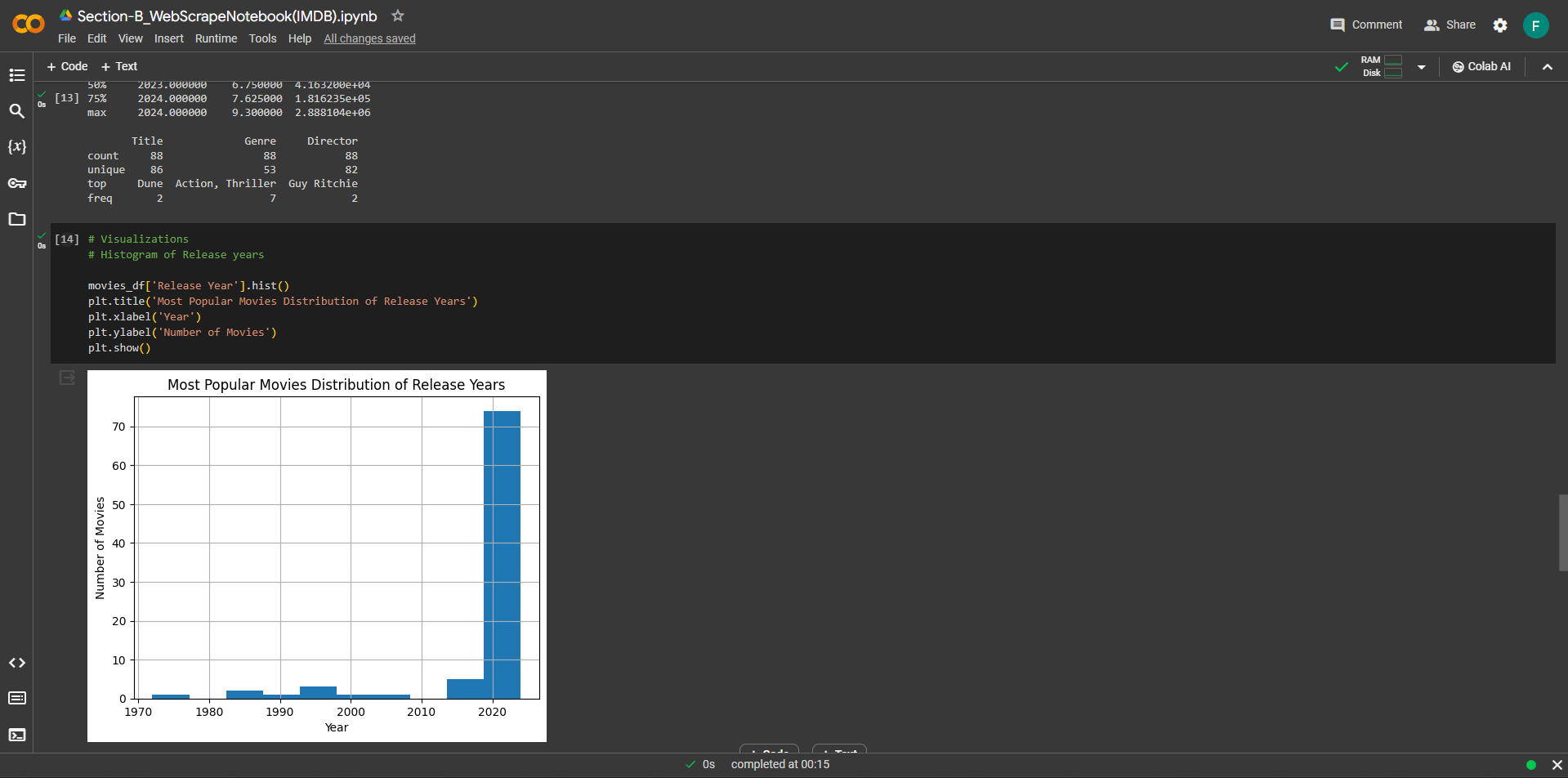
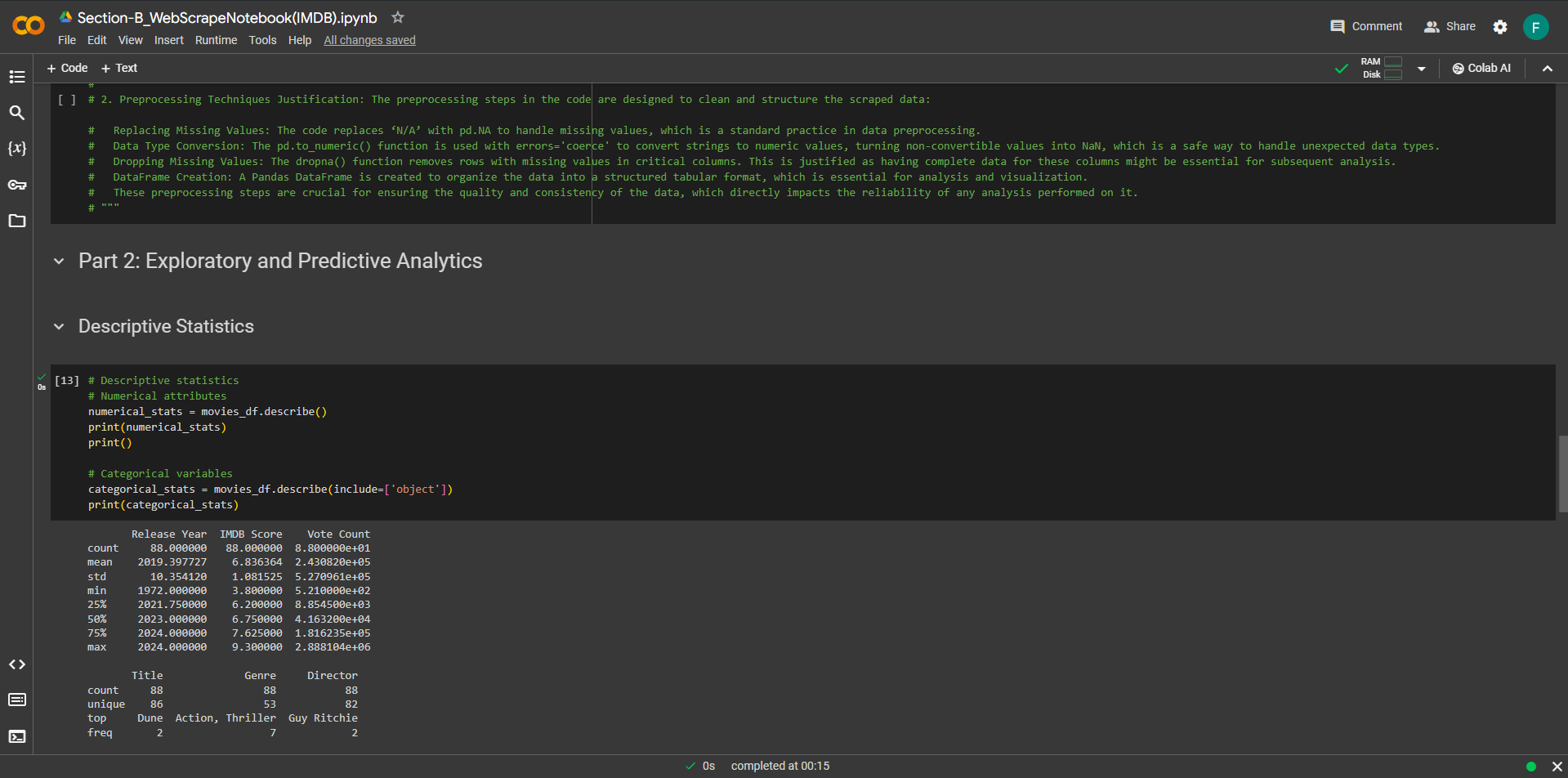
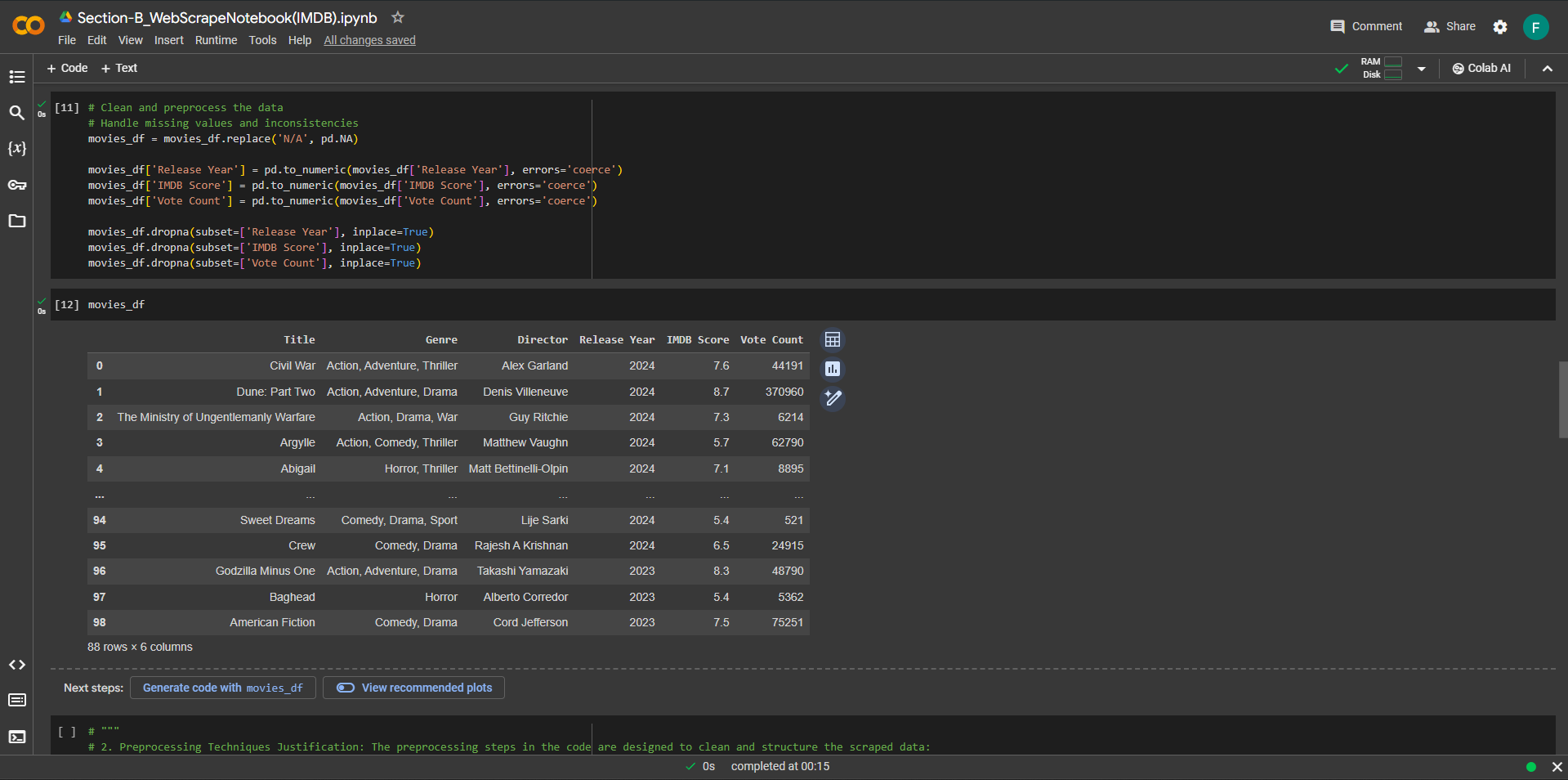
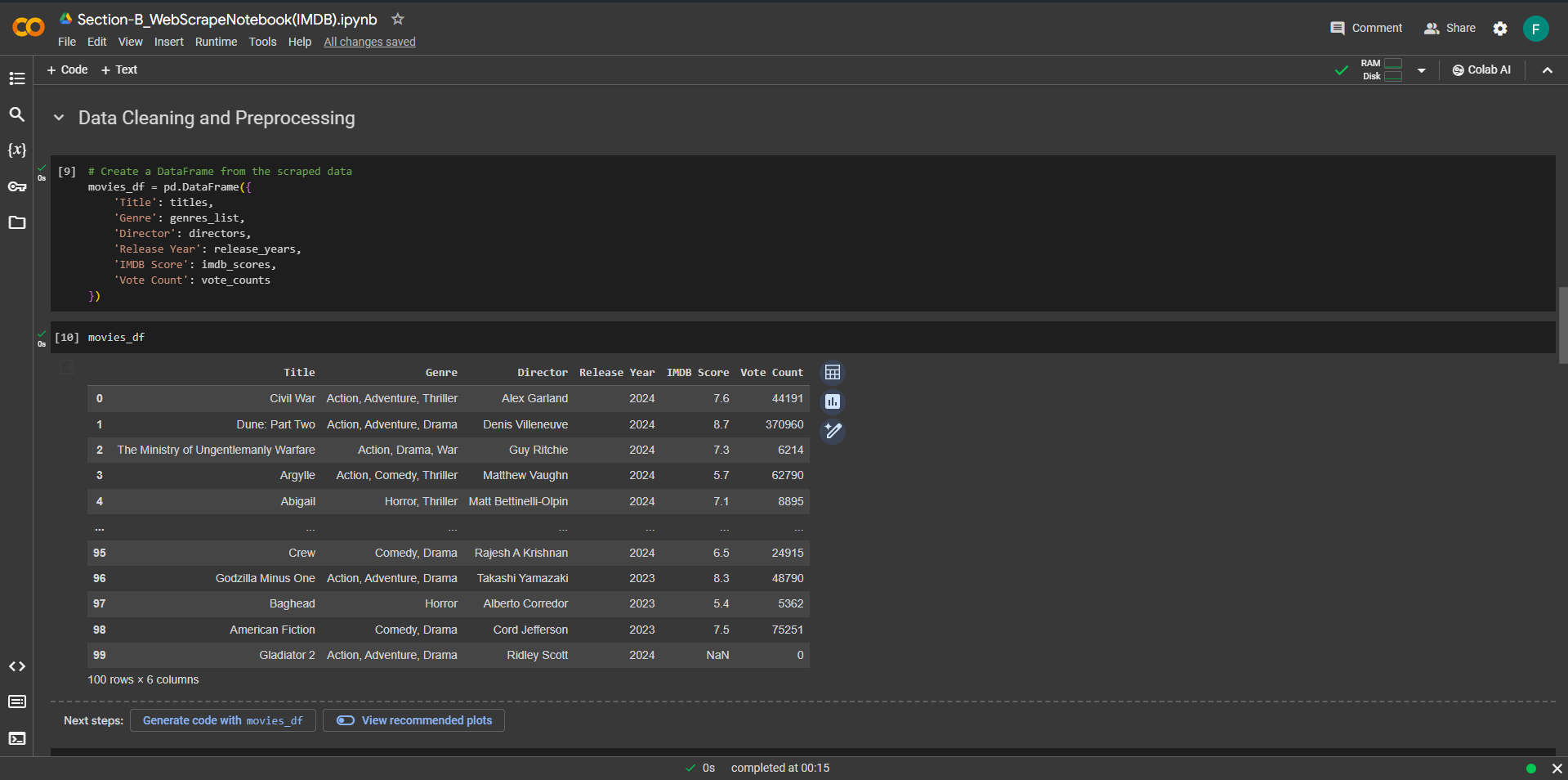
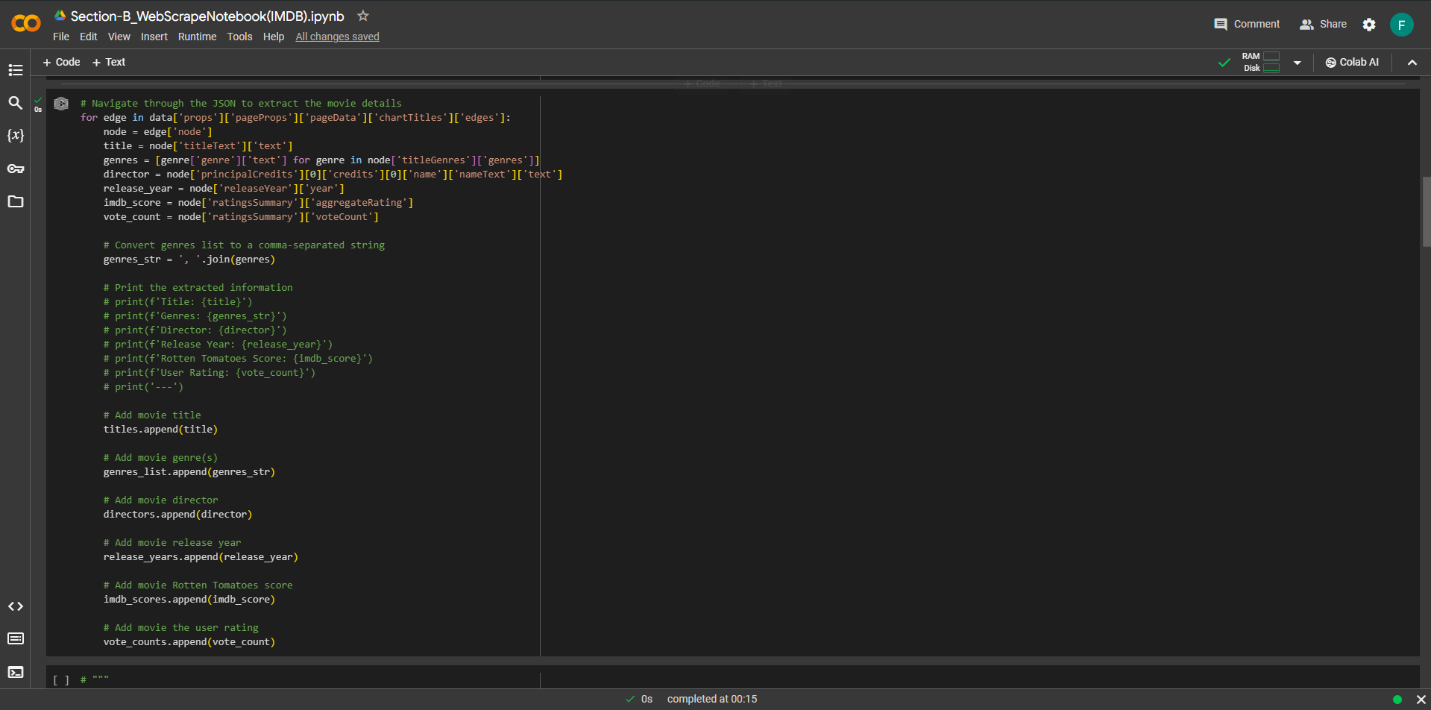
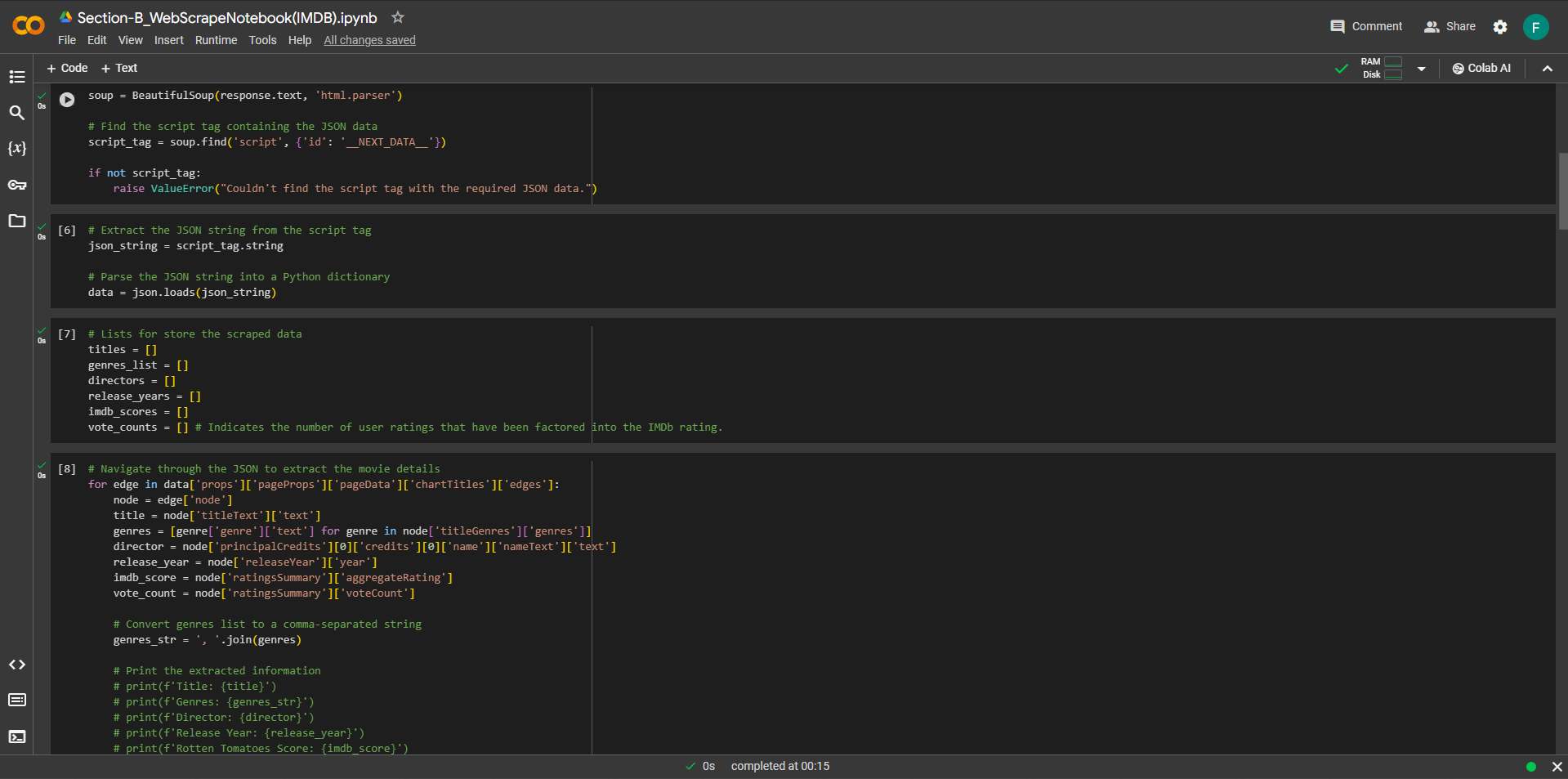
IMDb is one of the most popular and reliable sources for movie information, widely used by industry professionals and movie enthusiasts.

The website is updated regularly, ensuring access to the latest information.

Code File: <https://colab.research.google.com/drive/141pG7uCJYajb871ab5ZKo2Xo_SgSkyVH?usp=sharing>

**Notebook Screenshots:**

****

****

**1. Code Structure and Techniques:**

Python libraries like requests to send HTTP requests and BeautifulSoup to parse the HTML content has used. Here’s how the code works:

**HTTP Request:** The requests.get() function fetches the webpage content from IMDb’s chart page.

**User-Agent Header:** The headers dictionary includes a User-Agent string to mimic a browser request, which is often necessary to avoid being blocked by the server.

**HTML Parsing:** BeautifulSoup parses the HTML response and locates the <script> tag containing the JSON data (\_\_NEXT\_DATA\_\_).

**JSON Extraction:** The JSON string is extracted from the script tag and loaded into a Python dictionary using json.loads().

**Data Extraction:** The code iterates over the ‘chartTitles’ key in the JSON data to extract movie details into lists.

**2. Preprocessing Techniques Justification:**

The preprocessing steps in the code are designed to clean and structure the scraped data:

Replacing Missing Values: The code replaces ‘N/A’ with pd.NA to handle missing values, which is a standard practice in data preprocessing.

**Data Type Conversion:** The pd.to\_numeric() function is used with errors='coerce' to convert strings to numeric values, turning non-convertible values into NaN, which is a safe way to handle unexpected data types.

**Dropping Missing Values:** The dropna() function removes rows with missing values in critical columns. This is justified as having complete data for these columns might be essential for subsequent analysis.

**DataFrame Creation:** Pandas DataFrame is created to organize the data into a structured tabular format, which is essential for analysis and visualization.

These preprocessing steps are crucial for ensuring the quality and consistency of the data, which directly impacts the reliability of any analysis performed on it.

* **Limitations:**

**Data Quality**: The accuracy of the scraped data depends on the structure of the IMDb page. Any changes to the page layout could affect the scraping process.

**Legal Concerns:** Web scraping may violate terms of service of websites, and there’s a risk of legal issues if not done responsibly.

**Scalability:** The current approach may not be efficient for scraping large volumes of data due to potential rate limits and server restrictions.

* **Future Work:**

**Additional Data Points:** Consider scraping more detailed information such as cast, crew, and user reviews for a comprehensive analysis.

**Advanced Modeling:** Implement machine learning models to predict movie success based on various features extracted from the data.

**Real-time Analysis:** Develop a system for real-time data scraping and analysis to capture the latest trends and ratings.

**Conclusion**

**Dimensionality Reduction:**

Various methods such as PCA, LDA, and NCA offer tailored solutions for reducing the dimensionality of data, each with distinct advantages suited to different use cases. PCA stands out for its ability to capture variance while maintaining simplicity, making it ideal for visualization and feature extraction. On the other hand, LDA prioritizes class separation, making it valuable for classification tasks, while NCA offers supervised dimensionality reduction with a balance between accuracy and computational complexity for classification and clustering.

**Web Scraping and Movie Analytics:**

IMDb emerges as a rich and structured source for movie data, facilitating efficient scraping and analysis. Leveraging Python libraries like requests, BeautifulSoup, and pandas streamlines the scraping and preprocessing processes, enabling stakeholders to extract valuable insights. However, challenges such as data quality concerns, legal considerations, and scalability issues need to be addressed to ensure robust and responsible scraping practices.

**Recommendations:**

For dimensionality reduction, stakeholders are encouraged to explore additional techniques such as t-SNE, UMAP, Factor Analysis, or Random Forests to uncover non-linear relationships and latent factors in the data. Experimentation with different hyperparameter values and the enhancement of analysis techniques through cross-validation and pipeline optimization are also recommended to ensure the stability and reliability of dimensionality reduction methods.

In the realm of web scraping and movie analytics, stakeholders are advised to scrape additional data points such as cast, crew, and user reviews to enrich the analysis and provide comprehensive insights into movie attributes. Implementation of advanced modeling techniques to predict movie success based on extracted features, alongside the development of real-time scraping and analysis systems, can further enhance decision-making in the entertainment industry.

By embracing these recommendations, stakeholders can harness the power of dimensionality reduction techniques and web scraping practices to unlock valuable insights and drive informed decision-making, ultimately fostering innovation and growth in their respective domains.

**References**

* GeeksforGeeks. (2023). Dimensionality Reduction. Retrieved from <https://www.geeksforgeeks.org/dimensionality-reduction/>
* IBM. (2024). Dimensionality Reduction. Retrieved from <https://www.ibm.com/topics/dimensionality-reduction/>
* Analytics Vidhya. (2022). A Guide to Web Scraping Rotten Tomatoes. Retrieved from <https://www.analyticsvidhya.com/blog/2022/05/a-guide-to-web-scraping-rotten-tomatoes/>
* Medium. (2022). Web Scraping with Python and BeautifulSoup. Retrieved from <https://pub.towardsai.net/web-scraping-with-python-and-beautifulsoup-87c68341d100>