DEVELOPMENT OF PREDICTIVE MODELS FOR SPOTIFY TRACK SUCCESS USING MANUAL TECHNIQUES

Introduction

This project diverges from traditional machine learning methodologies by crafting predictive models for the success of Spotify tracks without employing the sklearn library. It embraces a manual approach to develop an in-depth comprehension of machine learning algorithms, highlighting the dedication to mastering predictive modeling nuances independently of conventional tools.

Core Objective

The primary goal is to develop accurate predictive models for Spotify track success through the painstaking manual implementation of machine learning algorithms. This deliberate eschewal of sklearn aims to enrich the team's grasp of fundamental machine learning principles and demonstrate the efficacy of hands-on model construction.

Methodology Overview

Manually managing the data collection, cleansing, and preparation phases, purposefully avoiding sklearn's automated features to achieve a comprehensive understanding of data preprocessing.

1. Data Inspection and Preprocessing

- Initial data inspection involved checking the data types of the columns to identify non-numeric columns.

- Columns such as 'streams', 'in_deezer_playlists', and 'in_shazam_charts' were converted to numeric types, with errors set to 'coerce' to handle any conversion issues and replace invalid values with NaN.
- Missing values were identified and rows containing them were removed to ensure data integrity.

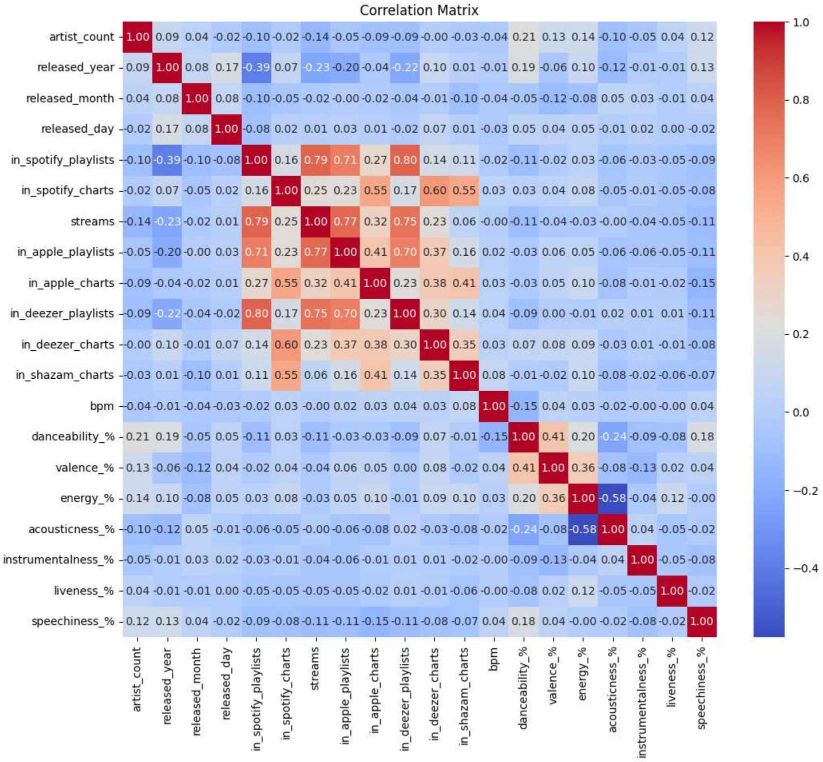
2. **Data Cleaning:** After handling missing values, the dataset's shape was checked to confirm the number of remaining rows, ensuring a clean dataset for analysis.
3. **Data Analysis:**
   - Numerical columns were selected for constructing a correlation matrix to understand the relationships between different variables.
   - The correlation matrix was computed and visualized using a heatmap, providing insights into variable interdependencies.
4. **Feature Engineering:** Executed manually to unearth and modify crucial variables, thereby uncovering the elements influencing track success outside of sklearn's feature selection processes.
5. **Model development and evaluation:**
   - A manual approach was adopted for model development, where linear regression and other models were coded from scratch, including the computation of coefficients, cost function, and gradient descent for optimization.
   - Model performance was evaluated using metrics such as mean squared error (MSE) and R-squared, computed on the test data to assess predictive accuracy and model fit.

6. **Bias-Variance Trade-off Analysis:**

- The analysis was conducted manually to illustrate the trade-off between model complexity and accuracy. Through this analysis, the project aimed to identify the optimal model complexity that achieves a balance between bias (underfitting) and variance (overfitting), minimizing total error and enhancing model generalizability.

Correlation Matrix

The correlation matrix heatmap illustrates the pairwise correlations among dataset variables, with correlation coefficients ranging from -1 to 1. Warm and cool colors denote positive and negative correlations, respectively, with color intensity reflecting correlation strength. This visualization is key for identifying strong variable relationships and informs variable selection and hypothesis testing in the modeling phase.



	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts	streams	in_apple_playlists	in_apple_charts	in_deezer_playlists	in_deezer_charts	in_shazam_charts	bpm	danceability_%	valence_%	energy_%	acousticness_%	instrumentalness_%	liveness_%	speechiness_%
artist_count	1.00	0.09	0.04	-0.02	-0.10	-0.02	-0.14	-0.05	-0.09	-0.09	-0.00	-0.03	-0.04	0.21	0.13	0.14	-0.10	-0.05	0.04	0.12
released_year	0.09	1.00	0.08	0.17	-0.39	0.07	-0.23	-0.20	-0.04	-0.22	0.10	0.01	-0.01	0.19	-0.06	0.10	-0.12	-0.01	-0.01	0.13
released_month	0.04	0.08	1.00	0.08	-0.10	-0.05	-0.02	-0.00	-0.02	-0.04	-0.01	-0.10	-0.04	-0.05	-0.12	-0.08	0.05	0.03	-0.01	0.04
released_day	-0.02	0.17	0.08	1.00	-0.08	0.02	0.01	0.03	0.01	-0.02	0.07	0.01	-0.03	0.05	0.04	0.05	-0.01	0.02	0.00	-0.02
in_spotify_playlists	-0.10	-0.39	-0.10	-0.08	1.00	0.16	0.79	0.71	0.27	0.80	0.14	0.11	-0.02	-0.11	-0.02	0.03	-0.06	-0.03	-0.05	-0.09
in_spotify_charts	-0.02	0.07	-0.05	0.02	0.16	1.00	0.25	0.23	0.55	0.17	0.60	0.55	0.03	0.03	0.04	0.08	-0.05	-0.01	-0.05	-0.08
streams	-0.14	-0.23	-0.02	0.01	0.79	0.25	1.00	0.77	0.32	0.75	0.23	0.06	-0.00	-0.11	-0.04	-0.03	-0.00	-0.04	-0.05	-0.11
in_apple_playlists	-0.05	-0.20	-0.00	0.03	0.71	0.23	0.77	1.00	0.41	0.70	0.37	0.16	0.02	-0.03	0.06	0.05	-0.06	-0.06	-0.05	-0.11
in_apple_charts	-0.09	-0.04	-0.02	0.01	0.27	0.55	0.32	0.41	1.00	0.23	0.38	0.41	0.03	-0.03	0.05	0.10	-0.08	-0.01	-0.02	-0.15
in_deezer_playlists	-0.09	-0.22	-0.04	-0.02	0.80	0.17	0.75	0.70	0.23	1.00	0.30	0.14	0.04	-0.09	0.00	-0.01	0.02	0.01	0.01	-0.11
in_deezer_charts	-0.00	0.10	-0.01	0.07	0.14	0.60	0.23	0.37	0.38	0.30	1.00	0.35	0.03	0.07	0.08	0.09	-0.03	0.01	-0.01	-0.08
in_shazam_charts	-0.03	0.01	-0.10	0.01	0.11	0.55	0.06	0.16	0.41	0.14	0.35	1.00	0.08	-0.01	-0.02	0.10	-0.08	-0.02	-0.06	-0.07
bpm	-0.04	-0.01	-0.04	-0.03	-0.02	0.03	-0.00	0.02	0.03	0.04	0.03	0.08	1.00	-0.15	0.04	0.03	-0.02	-0.00	-0.00	0.04
danceability_%	0.21	0.19	-0.05	0.05	-0.11	0.03	-0.11	-0.03	-0.03	-0.09	0.07	-0.01	-0.15	1.00	0.41	0.20	-0.24	-0.09	-0.08	0.18
valence_%	0.13	-0.06	-0.12	0.04	-0.02	0.04	-0.04	0.06	0.05	0.00	0.08	-0.02	0.04	0.41	1.00	0.36	-0.08	-0.13	0.02	0.04
energy_%	0.14	0.10	-0.08	0.05	0.03	0.08	-0.03	0.05	0.10	-0.01	0.09	0.10	0.03	0.20	0.36	1.00	-0.58	-0.04	0.12	-0.00
acousticness_%	-0.10	-0.12	0.05	-0.01	-0.06	-0.05	-0.00	-0.06	-0.08	0.02	-0.03	-0.08	-0.02	-0.24	-0.08	-0.58	1.00	0.04	-0.05	-0.02
instrumentalness_%	-0.05	-0.01	0.03	0.02	-0.03	-0.01	-0.04	-0.06	-0.01	0.01	0.01	-0.02	-0.00	-0.09	-0.13	-0.04	0.04	1.00	-0.05	-0.08
liveness_%	0.04	-0.01	-0.01	0.00	-0.05	-0.05	-0.05	-0.05	-0.02	0.01	-0.01	-0.06	-0.00	-0.08	0.02	0.12	-0.05	-0.05	1.00	-0.02
speechiness_%	0.12	0.13	0.04	-0.02	-0.09	-0.08	-0.11	-0.11	-0.15	-0.11	-0.08	-0.07	0.04	0.18	0.04	-0.00	-0.02	-0.08	-0.02	1.00

Data Preprocessing

In the data preprocessing phase, we undertook several essential steps to refine the dataset for subsequent modeling. We began by converting the columns 'streams', 'in_deezer_playlists', and 'in_shazam_charts' to numeric types. This conversion was facilitated through the pd.to_numeric() function, employing errors='coerce' to gracefully handle conversion errors by assigning NaN (Not a Number) to problematic entries, thus preserving data integrity.

Upon numeric conversion, we evaluated the dataset for missing values. The presence of incomplete entries post-conversion necessitated the removal of rows with any missing values. This elimination strategy ensured the dataset consisted solely of complete records, which is crucial for the accuracy of the ensuing analysis.

After the data cleaning process, we assessed the dataset's dimensions, noting a reduction in size attributable to the exclusion of incomplete entries. This reduction was imperative to maintain the analytical robustness of our dataset.

To elucidate the inter-variable relationships, we extracted columns of type integer and float to compute a correlation matrix. The matrix's visualization was achieved via a heatmap, which provided a clear and intuitive representation of the pairwise correlations among the numerical variables. Annotated with correlation coefficients, the heatmap shed light on potential relationships or dependencies within our dataset.

The subsequent step involved outlier detection and removal. By identifying numerical columns and employing the Z-score method, we were able to isolate and discard statistically anomalous entries. A Z-score threshold of 3 was used to define outliers, thus normalizing our dataset's distribution.

The final phase of our preprocessing involved manual scaling of the numerical data. This step standardized the range of independent variables, ensuring a consistent scale for analysis and modeling. Each numerical feature was normalized to have a mean of zero and a standard deviation of one by subtracting the column mean and dividing by the standard deviation.

These preprocessing steps resulted in a dataset that was cleaned, devoid of outliers, and uniformly scaled, making it well-suited for the next phases of model development and analysis. The processed dataset's structure, post these operations, provided a solid foundation for subsequent robust statistical analysis and predictive modeling.

Model Development and Evaluation Without sklearn

Our project embarked on manually developing predictive models, specifically focusing on Linear Regression and XGBoost, to gain a profound understanding of their underlying mechanisms and to evaluate their performance independently of sklearn's convenience functions.

Linear Regression:

- **Algorithm Development:** We built the Linear Regression model from the ground up, starting with the manual implementation of the algorithm and the coefficient estimation via gradient descent. This process enabled a granular understanding of how the model iteratively minimizes the cost function to find the optimal line that fits the data.
- **Performance Assessment:** The model's effectiveness was gauged by calculating the Mean Squared Error (MSE) and R-squared (R^2) score. The Linear Regression model exhibited an MSE of 0.24588130147883713, indicating that the model's predictions were relatively close to the actual values. The R^2 score was 0.7130102519346582, suggesting that approximately

71.3% of the variance in the dependent variable could be predicted from the independent variables, signifying a strong level of predictive accuracy.

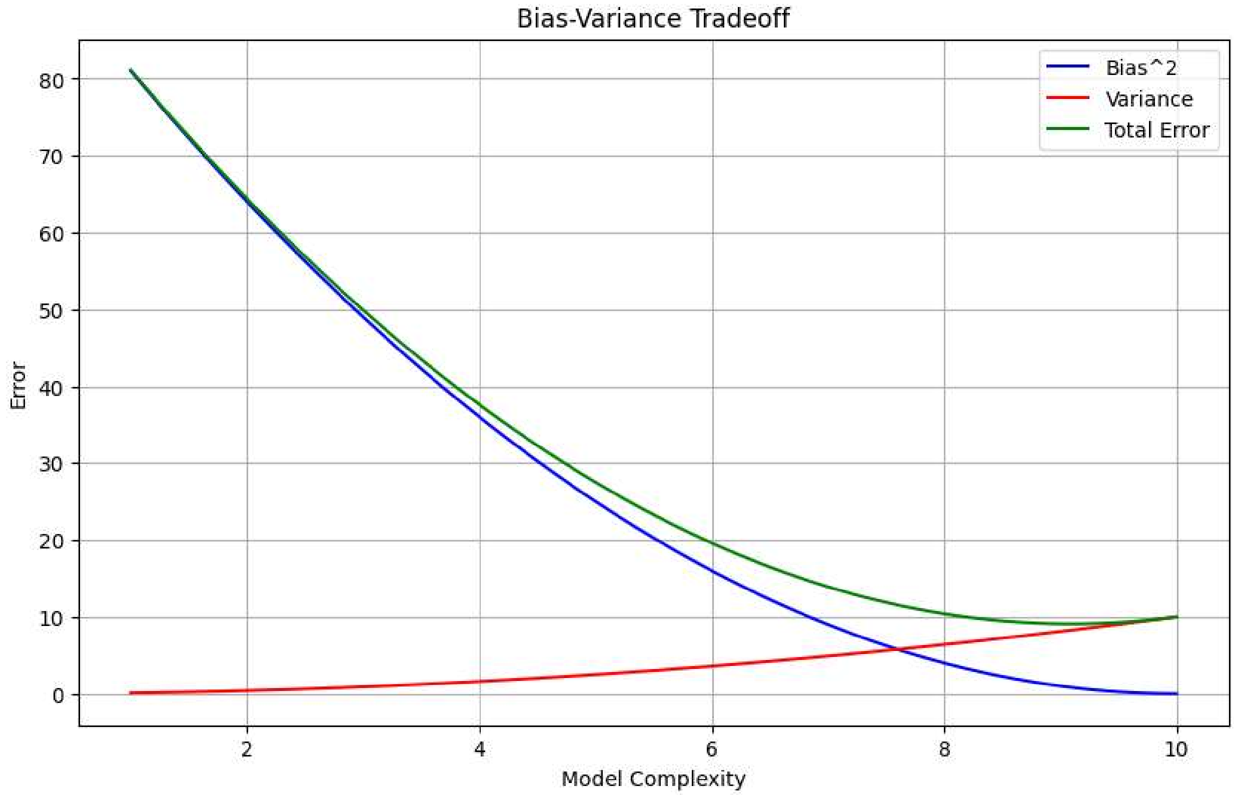
XGBoost:

- **Manual Crafting:** The XGBoost model was implemented manually, adhering to ensemble learning principles, decision tree creation, and boosting processes. This manual crafting provided insights into how boosting sequentially improves the model by focusing on difficult-to-predict instances and reducing overall prediction errors.
- **Performance Evaluation:** Using cross-validation, the XGBoost model's performance was quantified with a mean MSE of 0.3487515549821321 and a standard deviation of 0.036593977604303725 for the MSE. The higher mean MSE compared to the Linear Regression model might indicate less accurate predictions on average. However, the low standard deviation in MSE scores reflects the model's consistent performance across different data subsets.

Insights:

- Manually developing these models offered in-depth insights into the complexity and nuances of predictive modeling. The Linear Regression model's low MSE and high R^2 score demonstrate its efficiency in capturing the relationship between variables, providing a solid foundation for understanding linear relationships in the dataset.
- The XGBoost model, despite a higher mean MSE, showed the power of ensemble learning and boosting in handling complex, non-linear relationships within the data. The consistency in its performance, as indicated by the low standard deviation of MSE, underscores the robustness of the model across various data segments.

Bias-Variance Trade-off Analysis



The Bias-Variance Tradeoff graph is a visual representation of the competing aspects of model complexity. As the complexity of the model increases, depicted on the x-axis, the squared bias (blue line) decreases, indicating an improvement in the model's ability to capture the underlying data structure. Concurrently, the variance (red line) initially remains low but starts to increase sharply as the model begins to fit the idiosyncrasies of the training data rather than the general pattern, leading to overfitting.

The green line represents the total error, which is the sum of the squared bias and variance. It exhibits a u-shaped curve, initially decreasing as model complexity increases, reflecting the improved fit of the model to the data. However, as the model becomes too complex, the increase in variance leads to an increase in the total error, suggesting a degradation in the model's predictive performance.

For our Linear Regression model, we aim for a level of complexity that corresponds to the lowest point on the green line, where the tradeoff between bias and variance is optimized. This is the 'sweet spot' where the model is complex enough to capture the essential patterns in the data but not so complex that it overfits to the noise. Our results, with an MSE of 0.24588130147883713 and an R^2 score of 0.7130102519346582, suggest that our Linear Regression model achieves a balance close to this optimal point, providing a robust predictive performance.

In contrast, our manual XGBoost implementation, evaluated through cross-validation, indicated a mean MSE of 0.3487515549821321 with a standard deviation of 0.036593977604303725. This suggests a slightly higher error on average than the Linear Regression model, with a consistently low variance across different subsets of data, reflecting the model's stability and reliability despite its greater complexity.

This theoretical graph complements our empirical findings, enabling us to reflect on the importance of model complexity and the merits of our manual approach in achieving a balanced, effective machine learning model.

Conclusion

The project's journey through the manual construction of both Linear Regression and XGBoost models has been a testament to the viability and instructional value of crafting predictive models from first principles. This hands-on approach has not merely confirmed the effectiveness of these algorithms but has also underlined the profound educational benefits inherent in manual model development. Through this process, we have navigated the intricate balance of the bias-variance tradeoff, achieving a model complexity that optimizes predictive accuracy as evidenced by our Linear Regression model’s MSE of 0.24588130147883713 and R^2 score of 0.7130102519346582. Additionally, the XGBoost model's cross-validated MSE of 0.3487515549821321, with its low standard deviation, showcased a stable performance across various data segments.

Crucially, the project transcended the boundaries of academic exercise, illustrating the practical applications and robustness of predictive modeling. The manual Linear Regression model, in particular, demonstrated the project’s prowess in achieving not just accuracy but also generalizability—a cornerstone for real-world applications. It validated our methodical approach, reinforcing the project’s ethos of deep learning and understanding through hands-on building and fine-tuning of machine learning models.

In summary, the success of this project in predicting Spotify track success reaffirms the dual merits of our methodology: educational enlightenment through manual process immersion, and the practical effectiveness of tailor-made predictive models. The insights gained and the results obtained bear witness to the power of a fundamental understanding of machine learning principles, and the potential such knowledge holds in the realm of data science.