Technical Assessment Report:
DATA SCIENTIST

INTRODUCTION:

The Data Analysis and Modeling section of this report outlines a comprehensive process aimed at preparing and analyzing a raw dataset to extract meaningful insights. The first task focuses on data cleaning and transformation, where missing values are addressed, outliers managed, and data standardized to ensure consistency. Categorical variables are converted into numerical formats where necessary, and new features are created to enhance the model's predictive power. The second task involves exploratory data analysis (EDA) to uncover trends and relationships within the cleaned dataset, followed by statistical modeling to predict target variables. The section concludes with the delivery of a detailed report and a statistical model, along with visualizations to support the analysis.

ABOUT THE DATASET:

The dataset used for this analysis is titled "Technology Products," which contains detailed information on various tech items, including product features, pricing, and customer ratings. This dataset serves as the foundation for cleaning, transformation, and model development throughout the project.

LINK FOR DATA SET

Section 1: Data Analysis and Modeling
Task 1: Data Cleaning and Transformation

1. Introduction

This report outlines the steps taken to clean and transform a raw dataset related to technology products. The goal was to prepare the data for further analysis by addressing issues such as missing values, outliers, and ensuring that the dataset is in a format suitable for modeling.

2. Dataset Overview

The raw dataset consists of various features, including `Product_ID`, `Price`, `Launch_Year`, `Warranty_Period`, `Units_Sold`, `Customer_Rating`, etc. Below is an outline of the data cleaning and transformation process.

3. Data Cleaning Process

3.1 Handling Missing Values

- Missing values were identified using the `isnull().sum()` function.
- If any missing values existed, they were addressed by either removing rows/columns with excessive missing values or by imputing them with appropriate strategies (e.g., mean, median, or mode imputation).

3.2 Outlier Detection and Removal

- Outliers in the `Price` column were identified using the Interquartile Range (IQR) method.
- The lower bound was defined as `Q1 - 1.5 * IQR`, and the upper bound as `Q3 + 1.5 * IQR`.

- Outliers outside of these bounds were removed from the dataset, resulting in a cleaner and more reliable dataset for analysis.
- Number of outliers removed: {Number_of_outliers}

3.3 Data Type Conversions

- Columns such as `Product_ID`, `Launch_Year`, and `Warranty_Period` were converted to integer data types to ensure compatibility for further analysis.

4. Data Transformation Process

4.1 Feature Engineering

- A new feature, `Success_Rate`, was created by multiplying `Units_Sold` and `Customer_Rating`. This feature provides an indicator of how successful a product is based on sales and customer feedback.

4.2 Categorical to Numerical Transformation

- The `Success_Category` feature was introduced based on a threshold of `Success_Rate`. Products with a `Success_Rate` greater than or equal to the threshold were categorized as 'Successful', while those below were labeled as 'Not Successful'.

5. Standardization

- Numerical columns were standardized where necessary to ensure that features like `Price` were on a similar scale for future modeling purposes. This prevents any single feature from dominating the analysis due to scale differences.

6. Results

The final cleaned and transformed dataset, saved as 'Dashboard Insights.csv', contains the following key changes:

- Removal of {Number_of_outliers} outliers.

- Conversion of Product_ID, Launch_Year, and Warranty_Period to integer data types.
- Introduction of new features: Success_Rate and Success_Category.
- No missing values remain in the dataset.

7. Conclusion

This report highlights the data cleaning and transformation steps taken to prepare the dataset for further analysis. The cleaned dataset is ready for use in predictive modeling or dashboard creation.

LINK FOR THE CLEANED DATASET

Task 2: Statistical Modeling

1. Introduction

This report presents the Exploratory Data Analysis (EDA) and the development of a statistical model for predicting Units Sold. The analysis aims to uncover trends and relationships between key variables and to build a model that can accurately forecast the target variable.

2. Exploratory Data Analysis (EDA)

2.1 Data Distribution and Summary Statistics

- For continuous variables like Price, Units_Sold, Warranty_Period, and Success_Rate, histograms and boxplots were created to observe their distribution and detect any skewness or outliers.
- Bar charts were used to analyze categorical variables like Category, Brand, Processor_Type, and Operating_System.

2.2 Bivariate Analysis

- Scatter plots were created to explore the relationships between continuous variables like `Price`, `Units_Sold`, and `Success_Rate`. Key insights were drawn from the correlation between these variables.
- Boxplots were used to observe how categorical variables such as `Category` and `Brand` relate to the `Success_Rate`.
- Additionally, violin plots and pair plots were used to analyze how the `Success_Category` (successful vs. not successful products) relates to other variables.

2.3 Correlation Analysis

- A correlation matrix was computed for continuous variables (`Price`, `Units_Sold`, `Customer_Rating`, `Success_Rate`) to identify strong relationships between these variables.

2.4 Multivariate Analysis

- Pair plots and joint plots were used to visualize interactions between multiple continuous variables, such as `Price` vs. `Units_Sold`, colored by `Success_Category`.

3. Statistical Model Development

3.1 Target Variable Selection

- The target variable chosen for modeling is `Units_Sold`, which represents the sales volume of a product. Other potential target variables were considered, but `Units_Sold` was selected due to its importance in understanding product performance.

3.2 Model Choice: Linear Regression

- Linear regression was selected as the model to predict `Units_Sold` based on features like `Price`, `Customer_Rating`, `RAM`, `Storage`, and `Warranty_Period`.

- The decision to use linear regression was based on the assumption that there is a linear relationship between `Units_Sold` and these features. Additionally, linear regression is interpretable and efficient for relatively straightforward predictive tasks.

3.3 Model Training and Testing

- The dataset was split into 80% training and 20% testing subsets.
- The features used for prediction include `Price`, `Customer_Rating`, `RAM`, `Storage`, `Battery_Life`, and `Warranty_Period`.
- The linear regression model was trained on the training data and then tested on the test set.

3.4 Model Evaluation

- The model's performance was evaluated using the Root Mean Squared Error (RMSE) and R-squared (R^2) metrics.
- RMSE: {rmse_value}
- R-squared (R^2): {r2_value}
  These metrics help understand how well the model fits the data and how accurate the predictions are.

3.5 Actual vs Predicted Plot

- A scatter plot was generated to visualize the relationship between the actual and predicted values for `Units_Sold`. This helped assess how closely the model's predictions match the actual data.

4. Conclusion

This report highlights the key insights from the EDA and presents a linear regression model for predicting `Units_Sold`. The model demonstrates reasonable performance, and further tuning or more advanced models may be explored to improve accuracy.

LINK FOR THE NOTEBOOK WHERE ALL THE CODING PART IS DONE

Section 2: Machine Learning Development
Task 1: Model Development

1. Introduction

The goal of this task is to develop a machine learning model to classify products into `Success_Category` (e.g., 'Successful' or 'Not Successful') using various product attributes. Two models, K-Nearest Neighbors (KNN) and Naive Bayes, were selected and compared based on their performance on a cleaned dataset.

2. Model Selection

2.1 Problem Definition

The business problem is to predict the success of a product based on its attributes. This is a classification problem where the target variable is `Success_Category`.

2.2 Machine Learning Models

Two models were selected for this classification task:

- **K-Nearest Neighbors (KNN)**: A simple, non-parametric model that classifies based on the proximity of data points in feature space.
- **Naive Bayes**: A probabilistic classifier based on Bayes' Theorem, which assumes independence among predictors.

2.3 Feature Selection

The dataset features such as `Price`, `Units_Sold`, `Customer_Rating`, `RAM`, `Storage`, and `Battery_Life` were used as predictors. Categorical features like `Category` and `Brand` were excluded for simplicity.

2.4 Data Splitting

The data was split into training and testing sets with an 80-20 ratio using stratified sampling to maintain class balance in the target variable (`Success_Category`).

3. Model Training and Evaluation

3.1 K-Nearest Neighbors (KNN) Model

- **Training**: The KNN model was trained with `n_neighbors=5`.
- **Prediction**: Predictions were made on the test set.
- **Performance Metrics**:
  - Accuracy: {knn_accuracy:.2f}
  - Classification report metrics like precision, recall, and F1-score were used to evaluate the model.

3.2 Naive Bayes Model

- **Training**: The Gaussian Naive Bayes model was trained on the same training data.
- **Prediction**: Predictions were made on the test set.
- **Performance Metrics**:
  - Accuracy: {nb_accuracy:.2f}
  - The classification report provided additional metrics such as precision, recall, and F1-score.

4. Results

4.1 KNN Model Performance

- **Accuracy**: {knn_accuracy:.2f}
- **Classification Report**:
  - Precision, recall, and F1-score for both classes (`Successful`, `Not Successful`) were analyzed to understand the balance between false positives and false negatives.

4.2 Naive Bayes Model Performance

- **Accuracy**: {nb_accuracy:.2f}

- **Classification Report**:
    - Naive Bayes performance was evaluated similarly to KNN, focusing on how well it handled class imbalances.

5. Conclusion

The comparison between K-Nearest Neighbors and Naive Bayes showed that:

- KNN achieved an accuracy of {knn_accuracy:.2f}, while Naive Bayes had an accuracy of {nb_accuracy:.2f}.
- The detailed classification reports for each model show the strengths and weaknesses in handling the two classes (`Successful`, `Not Successful`).
- Based on the business context, one might prefer KNN for its slightly higher accuracy or Naive Bayes for faster computation in larger datasets.

LINK FOR THE NOTEBOOK WHERE ALL THE CODING IS DONE

Task 2: Model Optimization

1. Introduction

The goal of this task is to improve the performance of the initial models (K-Nearest Neighbors and Naive Bayes) by applying optimization techniques such as hyperparameter tuning, cross-validation, and feature scaling.

2. Optimization Techniques

2.1 Feature Scaling
Feature scaling was applied to standardize the dataset. Scaling helps models like KNN, which rely on distance measures, to perform better. The

`StandardScaler` was used to scale all the features to have a mean of 0 and a standard deviation of 1.

2.2 Hyperparameter Tuning for KNN

- **GridSearchCV** was employed to find the optimal value of the hyperparameter `n_neighbors` for the KNN model.
- The range of `n_neighbors` tested was from 1 to 10, and a 5-fold cross-validation was applied to assess the performance of each value.
- The best value for `n_neighbors` was selected based on the cross-validation accuracy score.

2.3 Cross-Validation for KNN

- Cross-validation was used to estimate the performance of the KNN model after hyperparameter tuning.
- The 5-fold cross-validation technique provided a more reliable estimate of model performance by reducing overfitting.

3. Performance Evaluation

3.1 Optimized KNN Model

- After hyperparameter tuning, the optimized KNN model was evaluated using the test dataset.
- The accuracy of the optimized KNN model was {knn_optimized_accuracy:.2f}, compared to the initial model's accuracy of {knn_accuracy:.2f}.
- The cross-validation accuracy score for the optimized KNN model was {knn_cv_scores.mean():.2f}, showing an improvement over the initial performance.

3.2 Naive Bayes Model

- Although Naive Bayes does not have significant hyperparameters to tune, feature scaling was applied to the data for consistency.

- The accuracy of the Naive Bayes model after scaling was {nb_optimized_accuracy:.2f}, compared to the initial accuracy of {nb_accuracy:.2f}.

4. Results

4.1 Optimized KNN Performance

- The optimized KNN model showed a noticeable improvement in accuracy from {knn_accuracy:.2f} to {knn_optimized_accuracy:.2f}.
- The cross-validation accuracy provided further confirmation that the model's performance was stable across different data folds.
- The classification report for the optimized KNN model showed improvements in precision, recall, and F1-score for both classes (`Successful` and `Not Successful`).

4.2 Naive Bayes Performance (After Scaling)

- The Naive Bayes model showed a slight improvement in accuracy from {nb_accuracy:.2f} to {nb_optimized_accuracy:.2f}, though the change was not as significant as for KNN.
- The classification report for the Naive Bayes model did not show substantial improvements, indicating that scaling did not have a large effect.

5. Conclusion

The optimization of the KNN model through hyperparameter tuning and scaling resulted in a significant performance improvement. The accuracy increased from {knn_accuracy:.2f} to {knn_optimized_accuracy:.2f}, and the cross-validation score indicated consistent performance. In contrast, the Naive Bayes model showed only minor improvement after feature scaling.

LINK FOR THE NOTEBOOK WHERE ALL THE CODING IS DONE

Section 3: Data Visualization and Communication

Task 1: Visualization Dashboard

INTRODUCTION

As part of my data analysis project, I developed an interactive dashboard using Power BI, which presents key insights derived from the dataset. The dashboard is designed to communicate these findings effectively to non-technical stakeholders through a set of visualizations that are easy to interpret and interact with.

Key Insights Presented in the Dashboard:

1. **Sales Performance by Product and Category**
   One of the primary insights is the distribution of sales performance across different product categories. The dashboard highlights which categories, such as electronics or apparel, are performing better in terms of units sold and revenue generated. This provides a clear picture of where the business is excelling and where there is room for growth.
2. **Customer Satisfaction Trends**
   The dashboard includes an analysis of customer ratings per product and brand, which helped identify trends in customer satisfaction. High-rated products are clustered, allowing stakeholders to focus on enhancing offerings that resonate well with customers. On the other hand, products with lower ratings are also highlighted, which could indicate areas requiring attention for quality improvements.
3. **Impact of Pricing on Sales**
   Another critical insight is the relationship between pricing and sales. The dashboard visually demonstrates how price adjustments correlate with changes in sales volume. This analysis will assist

decision-makers in determining optimal pricing strategies to maximize revenue without compromising on product appeal.

4. **Yearly Trends and New Launches**
   A time-series analysis shows how the number of new product launches and sales have evolved over the years. This section of the dashboard helps stakeholders understand market trends, enabling them to plan future launches or marketing strategies based on historical data.
5. **Operational Efficiency Metrics**
   The dashboard also presents operational insights such as warranty claims and battery life statistics. These metrics are key to understanding product durability and reliability, providing a foundation for improving future products.

Conclusion:
The interactive dashboard provides a comprehensive view of the business’s performance, with a focus on sales, customer satisfaction, pricing strategies, and product performance over time. It is tailored to enable non-technical stakeholders to extract actionable insights that can inform strategic decision-making.

LINK FOR THE DASHBOARD