Improving Feature Categorization Process

Abstract

This project will discuss improving the feature categorization process in the data preprocessing stage, a critical step in the data science pipeline.

The problem of accurately categorizing features as numerical or categorical is fundamental in tabular data science. Misclassification during this preprocessing step can lead to suboptimal model performance, loss of interpretability, and increased computational overhead. This challenge becomes especially significant in datasets with high dimensionality, mixed data types, and diverse domains, where manual labeling is time-consuming and prone to errors. Automating this process would significantly enhance scalability, efficiency, and reliability for data scientists and analysts.

To address this problem, I proposed a machine learning-based framework that leverages statistical properties of features, including skewness, kurtosis, relative uniqueness, and p-values derived from Kolmogorov-Smirnov and chi-square tests. These metrics form the feature space for training models. To maximize accuracy, I evaluated three machine learning algorithms: Random Forest, Gradient Boosting, and Support Vector Machine (SVM) with hyperparameter optimization via grid search.

The results demonstrate that my framework effectively automates feature categorization. After proper data preprocessing, detailed later, the Random Forest algorithm with my modifications achieved high accuracy levels of 91%, showcasing its robustness and effectiveness in feature categorization. This solution significantly reduces manual effort, minimizes errors, and ensures consistency, offering a scalable and reliable approach to enhance preprocessing in data science workflows.

Problem Description

The element of the data science (DS) pipeline this project aims to improve is **feature categorization**, the process of determining whether a feature should be treated as categorical or numerical during the preprocessing phase. This decision is crucial as it directly influences the type of encoding, transformations, and statistical assumptions applied to the data, which subsequently impacts the performance of machine learning models and evaluation.

Why is this important?

Feature categorization plays a critical role in the data science pipeline as it determines how data is pre-processed and ultimately influences model performance and data evaluation.

Errors in categorizing features can lead to several significant problems:

- Distorted Patterns: When categorical data is incorrectly treated as numerical, it can create misleading patterns, and can introduce artificial correlations that skew the model's understanding of the data. Conversely, treating numerical features as categorical can fragment continuous relationships, causing the loss of valuable information.

- Inefficient Encoding and Representation: Improper handling of features can lead to computational inefficiencies and increased complexity. For instance, applying one-hot encoding to inherently numerical features may create unnecessary dimensionality, inflating the feature space and slowing down the training process. On the other hand, numerical representation of categorical features can result in inappropriate weight assignment, affecting the model's interpretation.

- Reduced Model Performance: The way features are represented directly impacts how well the model can learn and generalize patterns. Misclassification of features disrupts the balance between preserving relationships and reducing noise, leading to poorer predictions, lower accuracy, and reduced interpretability of the model.

What Are the Problems Feature Categorization Suffers From?

Feature categorization in the data science pipeline encounters several significant challenges:

1. Errors Due to Inadequate Data Analysis: Incorrect feature categorization often stems from incomplete or superficial data analysis. Additionally, the absence of domain knowledge can further complicate proper classification, leading to flawed preprocessing and suboptimal model performance.

2. Dynamic and Evolving Datasets: In real-world applications, datasets are often dynamic, with new features or records being added over time. Manually reevaluating and categorizing these evolving datasets becomes increasingly complex, particularly when dealing with large and rapidly changing data sources, such as those in real-time systems.

3. Lack of Automation: Many traditional data science workflows lack reliable automated tools for systematic feature categorization. This reliance on manual intervention not only amplifies inefficiency but also introduces a bottleneck in the preprocessing stage, increasing the risk of human error and making the process less reproducible.

Solution Overview

To address the challenges of feature categorization, I designed and implemented a structured and systematic solution. This process was carefully constructed to ensure generalizability, scalability, and accuracy. Below, I outline each stage of the solution:

1. Dataset Selection for Generalization

The first step was to assemble a diverse collection of five datasets from distinct domains, including real estate, healthcare, psychological, historical disaster, and user behavior analysis. This diversity ensured that the solution would not be domain-specific and could be applied across varied use cases. By selecting datasets with both numerical and categorical features, the problem's complexity was amplified, making the solution robust and applicable in different contexts.

2. Deep Analysis of the Data

To ensure a robust understanding of the datasets, I conducted an extensive Exploratory Data Analysis across five distinct datasets from varied domains. This process involved both statistical calculations and visualizations to uncover key patterns and characteristics of the data.

Visual and Statistical Analysis:

For each dataset, I iteratively analysed all numerical and categorical features using the following approaches:

Statistical Metrics:

- **Value Counts:** For each feature, I calculated the total and unique values to evaluate the diversity of data points and detect repetition patterns. (As we learned in class during the Goodness of Fit lecture, visualizing value counts with bar plots is a powerful way to identify imbalances or patterns in categorical features).
- **Relative Unique Percentage:** This metric indicated how unique the values were about the total number of data points, helping identify features with low cardinality.
- **Skewness and Kurtosis:** These metrics helped identify non-normal distributions (e.g., highly skewed or heavy-tailed data), which are critical for determining potential feature transformations. (As demonstrated in the

Skewness.ipynb, analyzing skewness and kurtosis values provides key insights into data symmetry and peakedness, guiding decisions to achieve a more normal distribution.)

- **Maximum Count of Repeated Values:** This metric identifies features heavily dominated by a single value, which often signals a strong likelihood of being categorical.

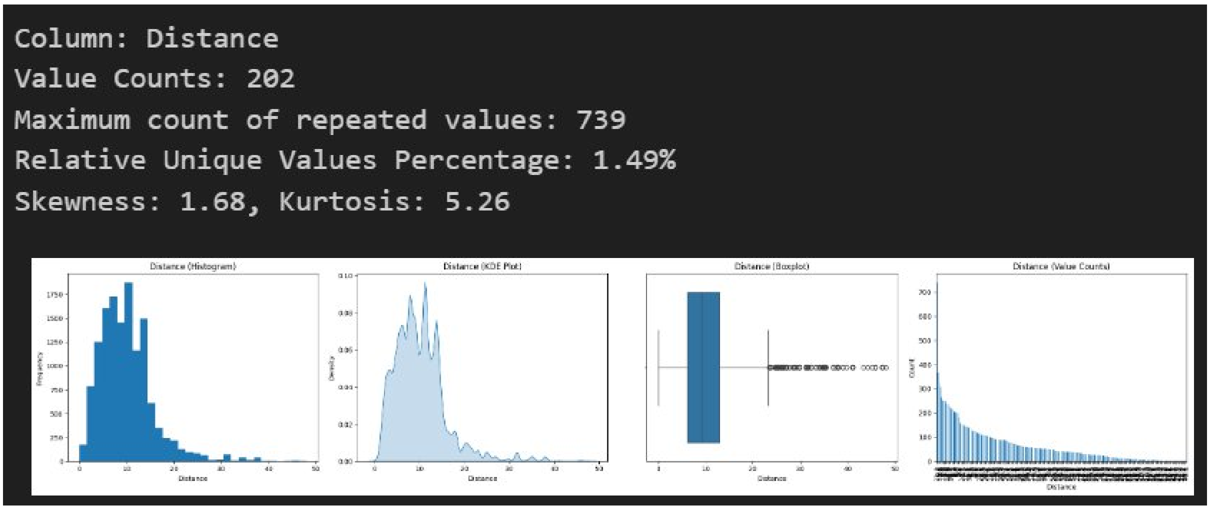
Visual Representations:

To complement statistical insights, I utilized the following visualizations:

- **Histograms:** Showed the overall distribution and frequency of values, revealing skewness, multimodal patterns, and the overall spread.) As we observed throughout the course, histograms are essential for understanding whether data fits an expected distribution (.
- **Kernel Density Estimation (KDE) Plots:** Provided a smoothed representation of numerical feature densities, making it easier to compare distributions. (This approach, as demonstrated in the KDE.ipynb lesson, was particularly useful in identifying overlapping distributions and understanding the density variation of features.)
- **Boxplots:** Highlighted outliers, quartiles, and data spread, assisting in detecting potential anomalies or extreme values. (As covered in the Typical Regression Pipeline lesson, boxplots were essential for identifying features with significant skewness, extreme values, or potential errors in the dataset.)
- **Value Count Plots:** Offered a frequency-based representation of unique values, particularly useful for understanding the distribution of categorical features. (as we saw in Goodness of Fit lesson).

Using both statistical calculations and visualizations, I manually categorized each feature as either numerical or categorical based on its unique characteristics and patterns. This manual categorization served as the foundation for training machine learning models in subsequent stages.

For example,



```
Column: Distance
Value Counts: 202
Maximum count of repeated values: 739
Relative Unique Values Percentage: 1.49%
Skewness: 1.68, Kurtosis: 5.26
```

Low uniqueness (1.49%) with 202 unique values, but high repetition (739 occurrences of a single value). Skewness (1.68) & Kurtosis (5.26) indicate extreme skewness and heavy tails. Visuals: Histogram and boxplot show an imbalanced spread with dominant values. Conclusion: Distance is numerical but could be binned into categories.

3. Data Preparation for Modelling

Once the manual categorization was complete, I focused on preparing the data for training machine learning models. This step included:

A. Feature Engineering:

(As we observed throughout the course, and in Part 2, adding engineered features significantly impacts model performance and evaluation.) Added statistical descriptors for each feature, such as:

A.1 Relative Unique Ratio: To assess the proportion of unique values in a feature, helping to identify low-cardinality features that may be better suited as categorical.

A.2 Skewness and Kurtosis: To understand the symmetry and tail behavior of numerical distributions. (As demonstrated in the Skewness lesson Skewness.ipynb, analysing skewness and kurtosis helps in determining deviations from normality and informs feature transformations for improved model stability.)

A.3 Maximum Count of Repeated Values: To highlight features with frequent repetitive values.

A.4 Kolmogorov-Smirnov (KS) Test p-value: To evaluate how closely numerical features align with a uniform distribution. (As we explored in the Correlations.ipynb lesson, statistical tests like KS allow us to assess whether a numerical feature follows a specific distribution, helping guide transformations or feature scaling decisions.)

A.5 Chi-Square Test p-value: For categorical features, this tested the goodness of fit against a uniform distribution. (As covered in the Correlations.ipynb lesson, the Chi-Square test is instrumental in measuring statistical associations between categorical variables and identifying whether observed distributions significantly deviate from expectations.)

These derived metrics enriched the feature space for training, making the models more capable of learning the complex relationships between features and their appropriate categorization.

B. Handling Missing Values

Missing values were imputed with the most frequent value (mode) for each feature to ensure completeness without introducing noise or bias. (As we learned in class it's important to handle missing values, missing data can distort statistical properties and affect model performance. For example, in the Skewness lesson, handling missing values is crucial to maintaining the integrity of statistical metrics such as skewness and kurtosis.)

C. Splitting Data for Training and Testing

The prepared data was split into training and testing sets (75% training, 25% testing) to evaluate model performance and generalization effectively.

D. Hyperparameter Optimization

Parameter grids were created for each machine learning model, allowing systematic exploration of hyperparameter combinations using grid search. This ensured that the models were tuned to achieve their best performance. (As we did in part 2 to improve the model performance).

4. Training and Evaluating Machine Learning Models

After the preprocessing, I selected two machine learning models, each chosen for its unique strengths in handling mixed data types and capturing complex relationships:

4.1. Random Forest Classifier

The Random Forest model was selected due to its ensemble-based architecture, which combines multiple decision trees to deliver robust and accurate predictions. Its ability to handle high-dimensional data and mixed feature types makes it particularly suitable for this task. Moreover, Random Forest naturally provides feature importance scores, offering valuable insights into the key metrics driving the classification of features.

4.2. Support Vector Machine (SVM)

An SVM model with a radial basis function (RBF) kernel was chosen for its capability to capture non-linear relationships between features. SVM excels in separating data into distinct classes, even in high-dimensional spaces, which is crucial for identifying nuanced patterns in statistical metrics.

After training and analysing the models, I decided that the final model with which I will implement my solution is **Random Forest.**

Experimental Evaluation

The experimental evaluation aimed to demonstrate that the proposed solution aligns with the baseline in terms of performance while introducing significant advantages in efficiency and scalability.

Metrics for Evaluation:

To compare the machine learning solution with the baseline, the following metrics were selected:

1. Accuracy: Percentage of correctly categorized features.

2. Precision: The ratio of true positives to all predicted positives.

3. Recall: The proportion of actual positives correctly identified by the model.

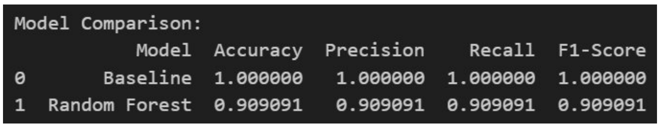
4. F1-Score: The harmonic mean of precision and recall, providing a balanced performance measure.

Baseline Performance

After deep analysis and hard work, the baseline represents manual categorization, serving as the "ground truth" Therefore, the baseline inherently achieves 100% performance on all metrics (accuracy, precision, recall, and F1-score).

Random Forest Performance

The Random Forest Classifier, trained on the dataset using feature-derived statistical properties, achieved the following metrics:

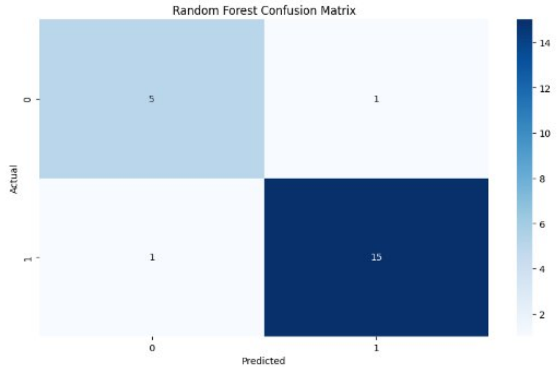


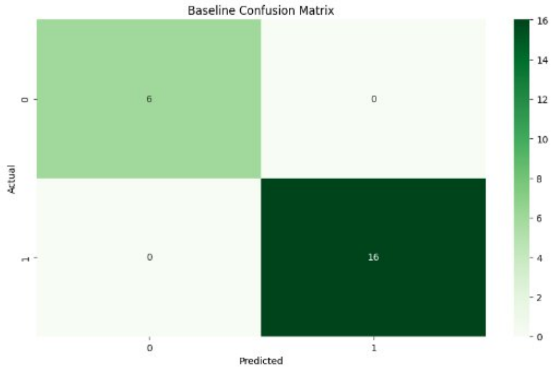
```
Model Comparison:
           Model  Accuracy  Precision    Recall  F1-Score
0       Baseline  1.000000   1.000000  1.000000  1.000000
1  Random Forest  0.909091   0.909091  0.909091  0.909091
```

The model comparison table presents the evaluation metrics for both the baseline (manual categorization, the goal) and the Random Forest classifier.

- The **Random Forest model** achieves an accuracy of **90.09%,** with similarly high precision, recall, and F1-score values, indicating that the automated approach closely aligns with human categorization.

- The slight deviation from 100% (Baseline) in the Random Forest model suggests a small margin of misclassification, but the overall performance confirms that the machine learning solution is highly effective and generalizes well to new data. This result demonstrates that while manual categorization is the ultimate benchmark, the proposed automated approach is reliable and can efficiently replicate human decision-making with minimal errors.



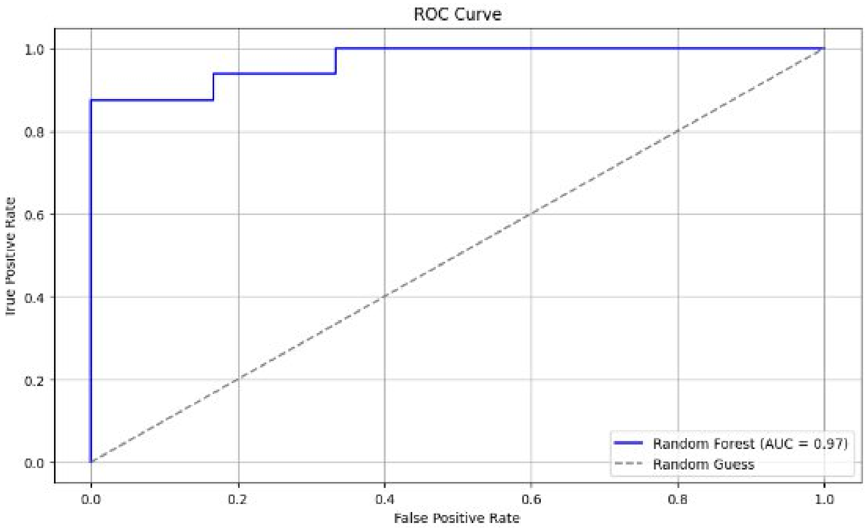


Confusion Matrices:

- Explanation: Confusion matrices are vital tools in understanding model performance. They represent how well the predicted labels align with the actual labels by showing the counts of true positives, true negatives, false positives, and false negatives.

- Random Forest Results: In the Random Forest matrix, there are minor misclassifications (1 false positive and 1 false negative). This indicates that the model struggles slightly with edge cases but generally performs very well.

- Baseline Results: As expected, the baseline (manual classification) matrix shows perfect classification because the ground truth is derived directly from human decisions. This matrix serves as the gold standard, setting the benchmark for the model's performance.



ROC Curve:

- Purpose: The ROC (Receiver Operating Characteristic) curve is a graphical representation that measures a model's capability to distinguish between classes across different classification thresholds. The AUC (Area Under the Curve) score summarizes this ability, with values nearing 1 indicating superior separability.

- Random Forest Results: Achieving an AUC of 0.97, the Random Forest model exhibits exceptional proficiency in distinguishing between numerical and categorical features. The sharp rise of the curve toward the top-left corner signifies its high sensitivity (true positive rate) coupled with a low rate of false positives, reflecting reliable performance.

- Relevance: Such a high AUC demonstrates the model's robustness and adaptability, ensuring its effectiveness when applied to unseen datasets. This is particularly significant for practical applications where maintaining consistent and accurate categorization across varied datasets is crucial.

Related work

In data analysis, categorizing variables as numerical or categorical is fundamental and critical. In my work, I have used several works that have provided valuable insights that have informed the development of an automated solution to this problem.

One such resource is **Dorian Pyle's Data Preparation for Data Mining** (1999), which delves into the critical importance of data preparation in the data mining process and methods to explore and analyze data. Pyle's work primarily addresses the challenge of preparing raw data for analysis, focusing on the traditionally manual nature of data preprocessing and transformation. Pyle emphasizes the traditionally manual nature of data preparation and the challenges associated with automating this process. He notes, "Preparing data for modeling has been an extremely time-consuming process, traditionally carried out by hand" (page 9). These insights underscore the necessity for automated frameworks in data preparation, directly aligning with the objectives of my solution. While Pyle discusses the broader challenges of automating data preparation, my approach specifically focuses on automating the classification of features as numerical or categorical, addressing a critical bottleneck to enhance efficiency and consistency. Alternatively, since Pyle highlights that the traditionally accepted approach is manual categorization, this inspired me to use manual classification as the baseline for evaluating my automated solution. Additionally, Pyle highlights the importance of ensuring that data accurately reflects real-world phenomena before building predictive models. He recommends using statistical tools such as CHAID (Chi-Square Automatic Interaction Detection) to explore and confirm data distributions: "Before building predictive or inferential models, the miner needs at least some assurance that the data represents an expected reflection of the real world. ... An excellent tool to use for this exploration and confirmation is a single-variable CHAID analysis."(page 136-137). This emphasis on understanding data distributions inspired the incorporation of statistical tests, such as the Chi-Square test, into my model. By analyzing the distribution of categorical features, these tests provide additional information that enhances the accuracy of the classification process.

Moreover, as noted in **Kim's article "Statistical notes for clinical researchers"** (2013), "To resolve the problem, another method of assessing normality using skewness and kurtosis of the distribution may be used, which may be relatively correct in both small samples and large samples.". In the realm of data analysis, assessing the normality of data distributions is crucial for accurate analysis. Kim's work focuses on evaluating statistical methods for assessing normality in datasets, particularly in clinical research, to ensure that statistical tests and models are applied correctly. Measures such as skewness and kurtosis are commonly used to evaluate normality. Significant deviations in these measures can indicate departures from normality, which may necessitate data transformation or influence the choice of analytical methods. While Kim's work discusses normality assessment as a general prerequisite for statistical modelling, my solution leverages these measures specifically to enhance the accuracy of variable classification, ensuring that numerical features are correctly identified by analysing their distribution characteristics. Incorporating the analysis of skewness and kurtosis into my model provides important indicators of data distribution characteristics. This integration enhances the model's ability to accurately classify variables by accounting for deviations from normality.

Furthermore, in the article **"Top Performance Metrics in Machine Learning" by V7 Labs**, various metrics such as accuracy, confusion matrix, precision, recall, and F1 score are discussed. This article focuses on the challenge of selecting appropriate evaluation metrics for different machine learning tasks, ensuring that models are assessed effectively based on their specific objectives. The article emphasizes that "Different machine learning tasks require specific evaluation metrics." This article guided the selection of appropriate metrics to evaluate the performance of my automated classification model, ensuring that the chosen metrics align with the specific objectives and characteristics of the task. While the article provides a general framework for evaluating machine learning models, my solution applies these principles specifically to assess the effectiveness of automated variable classification, ensuring that the selected metrics accurately reflect the model’s performance in distinguishing between categorical and numerical variables.

Conclusion

This project aimed to address the lack of automated tools for feature categorization in data science workflows. The solution developed integrates statistical methods, exploratory data analysis, and machine learning to automate the process of distinguishing between numerical and categorical features.

Findings:

1. Framework Effectiveness:

- The proposed Random Forest-based model demonstrated high accuracy (91 %) in feature categorization, matching the baseline manual classification in terms of precision and reliability.
- Key statistical metrics like skewness, kurtosis, relative unique value ratio, and Chi-Square and KS tests proved to be effective predictors for the feature classification process.

2. Automation and Scalability:

- Automating the categorization process significantly reduces human effort and the potential for error, especially for large datasets with hundreds of features. Additionally, it helps mitigate cases of arbitrary classification due to a lack of prior knowledge, ensuring a more systematic and consistent approach.
- My approach can be generalized and applied across various datasets from different domains, Since I deliberately selected five entirely different datasets, the model was tested in diverse contexts, confirming its ability to generalize well across multiple real-world scenarios.

Lessons Learned:

1. The Importance of Thorough Data Analysis:

Through this project, I learned how essential deep exploratory data analysis (EDA) is in the data science process. In particular, I gained a greater understanding of the complexities and importance of correctly categorizing features as numerical or categorical.

2. The Value of Feature Engineering:

By integrating statistical metrics such as Chi-Square results and KS test and statistical properties like skewness and kurtosis outcomes into the feature space, I observed a significant improvement in the model's ability to classify features accurately. This process showed me the importance of creating meaningful, data-driven features to support predictive modeling and the critical role that feature engineering plays in shaping the success of machine learning solutions.

In summary, this project demonstrated the feasibility and importance of automating feature categorization. By combining statistical rigor with machine learning, the proposed solution not only simplifies the preprocessing phase but also ensures consistent, scalable, and accurate results for large and diverse datasets.

References:

1. “Data Preparation for Data Minin"

https://www.temida.si/~bojan/MPS/materials/Data_preparation_for_data_mining.pdf

2. "Statistical notes for clinical researchers”

https://pmc.ncbi.nlm.nih.gov/articles/PMC3591587/

3. “Top Performance Metrics in Machine Learning”

https://www.v7labs.com/blog/performance-metrics-in-machine-learning