Kylee Laczkovich and Barbara Uzun

Dr. Johnson

DS 3001

Machine Learning Final Paper

1. Abstract / Executive Summary

This project explores whether a machine learning model can accurately classify numeric datasets as either real or fabricated. The motivation stems from growing concerns around research fraud, data integrity, and synthetic data generation (Durtschi et al., 2004). Using a collection of datasets labeled as either real (sourced from reputable databases like Data.gov and CDC) or fake (sourced from mock data generators, retracted studies, and simulations), we extracted three key statistical features: Benford's Law conformity error (Benford, 1938), mean skewness, and mean kurtosis (Field, 2013).

We trained a random forest classifier (Hastie et al., 2009) on these features, using an 80/20 train-test split and conducted k-fold cross-validation to ensure robustness. The model achieved a mean classification accuracy of 99% across folds with a standard deviation of just 0.02, suggesting both high performance and consistency. The confusion matrix on the test set showed no false positives or false negatives.

Importantly, the model also performed well when tested on synthetic datasets not included in the training set, correctly identifying all as fake with high confidence. These results suggest that fabricated datasets, even when modeled after real-world distributions, often fail to replicate subtle statistical patterns found in authentic data (Durtschi et al., 2004).

This project highlights the potential for statistical forensics and machine learning to work together in detecting data fabrication. While the dataset size was small, the consistency and strength of the model's performance show that further exploration of this methodology could be a valuable tool in academic integrity verification, fraud detection, or simulation validation. Limitations include a narrow feature set, reliance on numeric-only data, and a relatively small number of datasets. Future work could expand into categorical features, temporal patterns, and more varied domains.

---

2. Introduction

In an era where data is central to science, business, and policy, ensuring its integrity is more important than ever. Yet, instances of fabricated or manipulated data (intentional or not) continue to surface, often with serious consequences (Durtschi et al., 2004). This project attempts to answer the question: **Can machine learning detect whether a dataset is real or fake based on its statistical properties?**

Our approach draws from techniques like Benford's Law analysis (Benford, 1938; Durtschi et al., 2004) and distribution shape analysis (Field, 2013), combining them with supervised machine learning to build a predictive model (Hastie et al., 2009). Real data often contains organic patterns and imperfections, whereas fabricated data may be "too perfect," overly symmetrical, or deviate from expected statistical distributions.

We compiled a dataset composed of twelve real and fake numeric datasets. Real datasets were sourced from established databases such as Data.gov (Open Data Catalog, n.d.), CDC, and

the U.S. Census, while fake datasets came from mock data generators like Mockaroo (Mockaroo, n.d.), simulated data, or sources flagged for research fraud (ResearchBox, n.d.). Each dataset was treated as an individual observation, rather than combining them into one large table—an unconventional setup that mirrors the real-world challenge of evaluating entire datasets for validity.

We extracted three main features from each dataset. The first was Benford's Law error, which is a measure of deviation from expected digit distributions(Benford, 1938). The second is the mean skewness, which captures asymmetry in distributions; extreme values may suggest artificiality (Field, 2013). The third is mean kurtosis, which indicates peakedness or flatness; unusually low or high values may signal tampering (Field, 2013).

We trained a random forest classifier to distinguish between real and fake datasets using these features. We used an 80/20 train-test split and supplemented our analysis with k-fold cross-validation (Hastie et al., 2009) to validate the model's generalizability.

This paper will proceed as follows: In section three, we describe the datasets and their sources. Section four outlines our methodology and feature engineering process. Section five presents the results, including accuracy scores and interpretations. Section six discusses the implications, limitations, and potential future directions for this work.

---

3. Data and Sources

To train and evaluate our machine learning model, we collected various fake and real data sets to train the machine on patterns. Some predetermined patterns were suspicious digit patterns (violating Benford's Law), and a lack of variation, as well as the difficulty of replicating results. We collected data from Data Colada (Fake Data section) and Kaggle, as well as Data.Gov, CDC, and Census.Gov data. We used these datasets to train and test our model.

The datasets selected from these platforms varied across multiple domains, including public health, demographics, and financial records, which brought variety and authenticity into the "real" classification group.

Fabricated datasets were drawn from mock data generators like Mockaroo, simulated or synthetic datasets from Kaggle, and collections of known or suspected fabricated research data from sites such as Data Colada and ResearchBox. These sources were selected because they either admitted to generating synthetic content or had previously been flagged for issues related to data manipulation. These datasets often showed signs of unrealistic distributions, overly consistent patterns, or violations of expected digit frequency laws, making them ideal "fake" datasets.

To maintain consistency and reduce confounding variables, we limited our data to numeric datasets. We attempted to find data across multiple domains, hoping to increase the accuracy of the model.

---

4. Methods

After collecting our data from credible sites for the real data and less-credible sources such as Kaggle for the fabricated data sets, we had to do some data cleaning. We used Google Colab to clean and analyze the data. After reading in each data set as a csv file, we dropped the categorical columns and verified that all numeric columns were integers or floats rather than strings. We also had to add a column, called type, that was filled with either ‘fake’ or ‘real’ to set as the y-variable later. After the initial cleaning was done, we used the module scipy.stats among others that had the skew, kurtosis, and normaltest capabilities for later testing.

Once the data was cleaned, we wrote the definitions of the statistical methods we wanted to test. This included a definition for calculating Benford’s Error, mean skewness and mean kurtosis. Then, we wrote a function to extract these three features from each dataset. Then this list was converted into a table that the model could read in and evaluate.

For the model, we drew from the sklearn modules to use functions like RandomForestClassifier, train_test_split, and confusion_matrix. We then trained the model, split the data 80-20, fit the random forest classifier, made the prediction and evaluated the accuracy. We decided to make the confusion matrix a key visual, so used the modules seaborn and matplotlib to elevate the visualization of the result.

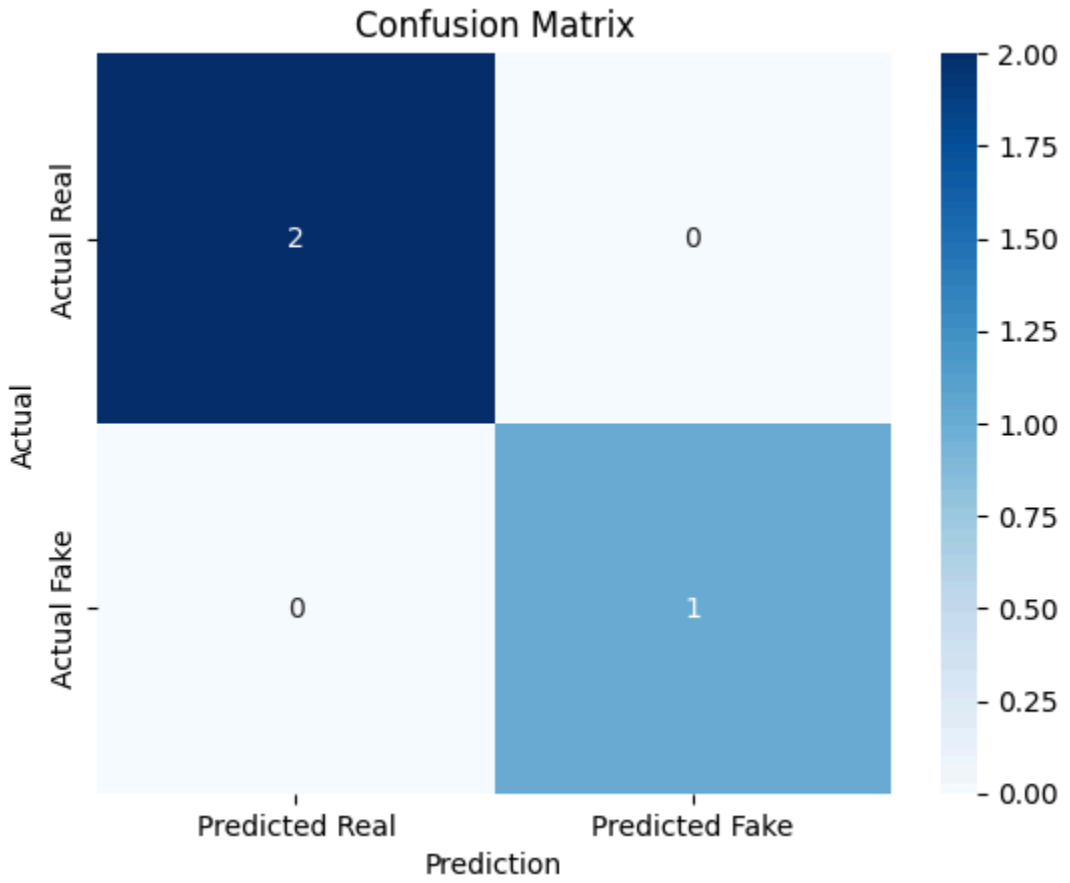
To further test the accuracy of the model, since the confusion matrix only evaluates one run through the model, we used a K-fold cross validation metric. This way, the model is tested on k-different combinations of these folds for a less-biased, more robust measure of performance accuracy.

Lastly, to address potential criticisms of the model, we tested three synthetic datasets. We again had to perform the same cleaning methods to prepare for the model. We then made a

for-loop for the synthetic dataset list, which again extracted the three features and made a prediction. For this code block, we also added a confidence level test. Finally, this outputted the datasets in order, their prediction label of real or fake, and a probability of the truth.

---

5. Results



One of our main experiment results was assessing the accuracy of the model through a confusion matrix. Our dataset is a bit smaller than normal because each required some cleaning beforehand to make sure it ran through the model. We also only focused on numeric data, as this is easier to mathematically detect fabricated data by running these well-known tests. We

performed the 80-20 split of training and testing data, so this resulted in testing three datasets for one confusion matrix output. The matrix indicated that the model was highly accurate with no false positives or false negatives.

Index	Benford Error	Mean Skewness	Mean Kurtosis
0	0.3105	-0.0029	-1.2087
1	0.1874	24.7502	746.3266
2	0.3744	-0.0125	-1.1929
3	0.3230	0.0852	-1.0900
4	0.1406	1.6099	7.8481
5	0.0631	2.4162	7.0540
6	0.9254	-0.0362	-0.2406
7	0.4622	11.5551	223.6456
8	0.2459	2.1435	21.8211
9	0.4458	0.6098	1.8016
10	0.4800	0.4505	-0.3877
11	0.4800	0.4505	-0.3877

The next piece of results were the numeric test outputs for each of the datasets. This is what the algorithm used to make the classification prediction of a real or fake dataset. We did not

specify an exact cutoff for a given column to indicate fake or real. The model determined this on its own through supervised learning. This is because there is no absolute rule for a cutoff and varies depending on the data being used, whereas these datasets come from different sources and topics. However, there are some general guidelines for these tests such as generally if Benford's error is greater than 0.40-0.5, this indicates that the data is fabricated. Moreover, for kurtosis, if the mean is less than zero, this suggests artificial symmetry or if there are extreme kurtosis values this suggests manipulation as well. Lastly, for mean skewness, real datasets mainly have moderate skewness between (-1,1), so extreme values outside of this are suspicious.

Accuracies:	[1.0, 1.0, 0.95, 1.0, 1.0]
Mean Accuracy:	0.99
Standard Deviation	0.020000000000000018

Another numeric summary we have is a k-fold cross validation. This method assesses the model's accuracy across multiple runs of the data. This results in reduced bias as the resampling method allows for different training and testing splits and robust performance checks. Overall, the output is highly accurate with a mean accuracy of 0.99 and 0.02 standard deviation. This is a great performance outcome and means that it could be potentially a valuable model for testing datasets in the future.

Synthetic Dataset Classifications

df_13	Fake	Confidence: 0.77

df_14	Fake	Confidence: 0.75
df_15	Fake	Confidence: 0.95

One potential criticism for the current approach would be the classification of synthetic datasets. To combat this, we tested our model on three synthetic datasets to see if there were any patterns. For all three synthetic datasets, the model classified them as fake with relatively high confidence levels. This is an important distinction. This means the simulation of the real data does not fit the normal patterns closely enough to be considered real.

---

6. Conclusion

This project attempts to show whether machine learning can detect fabricated numeric datasets by analyzing statistical properties commonly associated with natural data(Durtschi et al., 2004; Benford, 1938). Our model achieved a mean classification accuracy of 99% with no false positives or false negatives on the test set. Also, its consistent performance across folds and successful classification of out-of-sample synthetic datasets suggest the model is robust and generalizes well within numeric datasets (Hastie et al., 2009).

The project has several important findings. First, traditional metrics such as Benford's Law conformity, skewness, and kurtosis remain powerful indicators of data authenticity (Field, 2013). Second, even simulated or "synthetic" datasets generated with real-world distributions often fail to replicate the subtle irregularities and noise present in authentic data, which the model can learn to detect (Durtschi et al., 2004). Third, though we did not enforce strict

thresholds, the model learned classification boundaries based on patterns that aligned with statistical intuition.

Limitations

There are several limitations. The most notable is the small dataset size: we only use twelve datasets for training and validation. While the model's performance was strong, this limits the generalizability of the findings. Additionally, we only used numeric data; fabricated datasets with categorical features may behave differently and be more complex. The current model also treats datasets as single observations, which simplifies the problem but may not scale well.

Another potential issue is human error. Real datasets might contain mistakes, missing values, or inconsistencies that mimic the patterns of fabricated data. Our model could mistakenly classify such datasets as fake if the mistakes are large enough (Field, 2013).

Future Directions

There are many directions this could go in the future. First, it would be necessary to expand the dataset. Larger amounts of real and fake datasets would improve model training and evaluation. Then, we could expand to categorical datasets. Most datasets are not solely numeric. Another possibility would be to employ granular classification. Instead of treating each dataset as a single observation, we could develop models that classify individual rows or features as suspicious. Additionally, unsupervised anomaly detection could produce some interesting results in future research. Clustering or density-based methods could reveal hidden patterns or detect fabrication without labeled data (Hastie et al., 2009). Lastly, researching explainability and

interpretability would be beneficial. Tools such as SHAP and LIME could offer deeper insights into why the model flags certain datasets as fake (Lundberg & Lee, 2017).

In summary, this work provides a promising outlook for automated dataset authentication using simple statistical features and machine learning. By identifying patterns that are difficult for fabricators to mimic and easy for models to learn, there may be opportunities for systematic checks for data integrity across domains.

Works Cited

Benford, F. (1938). *The law of anomalous numbers*. *Proceedings of the American Philosophical Society, 78*(4), 551–572.

Breiman, L. (2001). *Random forests*. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Durtschi, C., Hillison, W., & Pacini, C. (2004). *The effective use of Benford's Law to assist in detecting fraud in accounting data*. *Journal of Forensic Accounting, 5*(1), 17–34.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 4765–4774.

Nigrini, M. J. (2012). *Benford's Law: Applications for forensic accounting, auditing, and fraud detection*. John Wiley & Sons.

NIST/SEMATECH. (2012). *e-Handbook of Statistical Methods*. National Institute of Standards and Technology. https://www.itl.nist.gov/div898/handbook/

Scikit-learn developers. (2023). *Scikit-learn: Machine learning in Python*. https://scikit-learn.org/stable/

Mockaroo. (n.d.). *Mock data generator*. https://mockaroo.com/

Open Data Catalog. (n.d.). *Data.gov*. https://www.data.gov/

ResearchBox. (n.d.). *Fake data archive*. https://researchbox.org/

Kaggle. (n.d.). *Datasets*. https://www.kaggle.com/datasets