International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 4, Jul-Aug-2026, ISSN (Online): 2395-566X

AI-LoanApproveX: An Intelligent Machine Learning-Based Loan Approval Prediction and Decision Explanation Framework Using Random Forest and SHAP Explainable AI

Moaiz Kazi, Sufiyan Ansari, Arhaan Shaikh, Madhvi Saxena, Usaid Khairdi
Department of Technology
Pune, India

Abstract- The digital world we live in today is creating an amount of unorganized data. This has led to something called hoarding. People who use cloud storage often feel overwhelmed by the number of files they have. They have a time searching for things and their organized systems start to fall apart. To solve this problem we are introducing XAI-CloudAssist. XAICloudAssist is a tool that is designed to work in the cloud. It automatically sorts files into categories using a group of Random Forest models. Unlike automated systems that are hard to understand our approach is transparent. We use something called Explainable AI to make sure people can see how it works. We use SHAP values to show why each file is sorted into a category. We give a score to each piece of information about the file to show how important it is. When we tested XAI-CloudAssist it was able to sort files 98 percent of the time. This shows that things like how space a file takes up and how often it is used are very good, at predicting what category it belongs in. XAICloudAssist also explains why each file is sorted into a category. This helps people understand the decisions it makes and trust that it is working correctly. XAI-CloudAssist is related to Cloud Computing and File Organization and Machine Learning and Explainable AI and SHAP and Random Forest and Automation and Data Governance.

Index Terms—Cloud Computing, File Organization, Machine Learning, Explainable AI, SHAP, Random Forest, Automation, Data Governance.

I. INTRODUCTION

The migration of data to cloud services has enhanced its accessibility, it has also created disarray in the hierarchy. As user grows thousands of files- from multimedia content to valuable professional data, manual sorting becomes unfeasible, hence, slow retrieval of files and the usage of keyword based search instead of discovery of similar items become a problem. Machine learning addresses the fundamental inefficiencies through organization on proactive basis. File metadata (like, size of file, modification of time, user set priority) would enable the system to automatically classify and sort files based on learned model. The primary hurdle toward its acceptance, is ”the trust gap”. User does not wants to rely on algorithm for file management and classification, whose action it cannot understand and justify. In this paper we present a solution that bridges this gap: XAI-CloudAssist; where we blend high classification accuracy with game-theory based interpretability.

II. LITERATURE SURVEY

The structure of digital asset management within organizations has changed drastically, moving from flat and user defined structures to dynamic, AI driven architectures. This section covers three interrelated areas: the evolution from the hierarchical file systems, the evolution of using ensemble methods in the categorization of metadata, and the evolution of frameworks for game-theoretic interpretability. The earliest way of storing data was in Hierarchical File Systems (HFS), which is a hierarchical tree like structure; data was accessed based on path indexing, where retrieval is path dependent. It was observed early in data governance, that the main fault with HFS is the ’Digital Graveyard’ which allows the user to save data but due to lost paths or user cognitive lapse in the structure, files would never be retrieved.

A. Evolution of File Systems: From Hierarchy to Semantics
The genesis of data storage was rooted in the Hierarchical File System (HFS),Gifford et al came up with ”Semantic File Systems” (SFS) where users would lose their dependency on the path by automatically extracting the relevant meta-data, via transducers, allowing for searches and a virtual directory. Yet, earlier SFS structures rely on rules-based processing which could not account for dynamic user preferences. The changeover to Cloud Object storage, such as Amazon S3, or Azure Blob services further disconnected data from physical devices, using ’flat’ structures; though it allowed scale, it led to a more acute organizational problem as the user was left with millions of objects with no organizational navigation layer. This section attempts to bridge this gap with an intelligent layer over flat cloud architectures.

B. Supervised Learning in Metadata Categorization
Early approaches for automated file categorization relied on supervised learning algorithms of a textual nature (Naive Bayes, SVM), performing well on document rich datasets. However, due to the fact that these models need full-file parsing (content inspection) they would be computationally prohibitive in the cloud setting.

Academic efforts of late have focused on Metadata-Driven Categorization. Unlike deep learning architectures like CNNs and Transformers which are optimized for pixel- or sequenceunstructured data, the structured tabular nature of metadata in file logs makes Ensemble Learning algorithms mathematically superior. The Random Forest (RF) algorithm described in Breiman works by generating an ensemble of decorrelated decision trees using the approach of ”Feature Bagging”. Research such as Quinlan suggests RF performs well in a cloud context due to its:

Non-Linear Interaction capability: RFs do not make assumptions of linear distribution and can identify complex interrelations between user priority, file extension and access frequency. Resistance to data noise: Incomplete metadata (a common occurrence within cloud sync logs) doesn’t impact a single decision tree in an ensemble quite as drastically.

C. The Interpretability Crisis and the Rise of XAI
As the machine learning models were becoming more accurate, they were simultaneously becoming less explainable: the Black Box problem. For the cloud-based file assistant the ’black box’ represents a key failure. If the user is unsure as to why the file was placed in the ”Archive” or ”Priority” folder, trust is placed at a minimum in the system. Early methods of explaining predictions attempted to address this issue using local surrogate models to approximate the behavior of black box models such as LIME. LIME was shown to be not mathematically consistent and so Lundberg and Lee (2017) proposed the SHAP framework:

Based on Coalitional Game Theory and Shapley values SHAP takes each metadata parameter as a ’player’ and the model’s classification as a ’payout’. By summing up the average marginal contribution of each player (meta-data parameter) for all possible combinations (permutations of the parameter space), SHAP produces globally consistent explanations of the importance of each feature and locally explains why each file has been classified in its present position. We are one of the first to apply this particular XAI framework to a niche application of automated cloud file management.

D. Gaps in Current Commercial Solutions
Existing cloud assistants, like Google Drive’s ”Suggested” feature or Microsoft OneDrive’s ”Recent” tab rely on simple FRU algorithms (frequency-recency). ”Explainable” in the sense of ”simple,” they have no predictive power. Highly sophisticated AIs are implemented within high-end enterprise data governance systems, with no justification visible to the end-user. XAI-CloudAssist bridges this gap, combining the high accuracy (98 Percentage) of a Random Forest with the explainability provided by SHAP, creating a ”Glass-Box” solution. Dataset Description The efficacy of XAI-CloudAssist is contingent upon the quality of its underlying metadata. We utilize a dataset modeled after enterprise logs, designed for multi-class classification.

E. Data Components and Structure

#	Column	Non-Null Count	Dtype
0	loan_id	4269 non-null	int64
1	no_of_dependents	4269 non-null	int64
2	education	4269 non-null	object
3	self_employed	4269 non-null	object
4	income_annum	4269 non-null	int64
5	loan_amount	4269 non-null	int64
6	loan_term	4269 non-null	int64
7	cibil_score	4269 non-null	int64

The dataset is tabular, where each row represents a unique entry identified by a unique ID. Key features include:

III. DATA PREPROCESSING PIPELINE

Preprocessing is a step that takes the logs and turns them into numbers that are easy to work with. This makes the logs useful for based learning. The goal of preprocessing is to get the logs into a format that is consistent and easy to understand, which is really important for ensemble-based learning to work well with the logs. Ensemble-based learning relies on the logs being, in a format.

A. Metadata Sanitization

Logs usually have their way of looking. To make them look the same we use a process to clean up the log data. This process makes all the log data look the same. We do this to the log data so that all the log data is standardized. This way the log data is easy to look at and understand. data.columns = data.columns.str.strip()

B. Categorical Encoding

Categorical descriptors must be transformed for the RF architecture. We use Label Encoding:

$$L(Ci) = n, \text{ where } n \in \{0,1,...,k-1\} \quad (1)$$

C. Feature Segregation

We get rid of things that do not matter to make the information clearer. The things that are left are then divided into two parts for training and testing. We use 80 percent for training and 20 percent for testing. We do this in a way that's always the same so we can get the same results again by using a special setting that is always 42. This helps us make sure we get the results every time we do Machine Learning things, with our Machine Learning features.

IV. METHODOLOGY

The AI-CloudAssist framework is set up in a way. It has five steps that help turn messy computer data into something that makes sense. The AI-CloudAssist framework does this by following these steps one by one. This helps the AICloudAssist framework take what is basically random computer information and turn it into a neat and organized system. The AI-CloudAssist framework is really good, at taking this digital noise and turning it into a structured information hierarchy that is easy to understand.

A. Stage 1: Metadata Harvesting (The Digital DNA)

The system takes out details like how big something is, how often people use it who owns it and what people are doing with it. This information helps make a profile, for each thing without looking at the stuff inside.

B. Stage 2: Syntactic Refinement and Encoding

Metadata is. Translated. We use the 'str.strip()' function to fix inconsistencies in the header. Label Encoding is used to convert categories into integers, for the intelligence engine. This helps the intelligence engine to process the metadata The metadata cleaning and translation process is crucial. We rely on Label Encoding to map categories to integers. The 'str.strip()' function helps to remove characters.

C. Stage 3: The Intelligence Engine (Council of Experts)

The system uses a **Random Forest** that has 100 decision trees. Each **Random Forest** tree looks at the metadata. Then decides on a category. The **Random Forest** makes its choice by picking the category that most of the trees agree on.

$$\hat{y} = \text{mode}\{T1(x),T2(x),...,T100(x)\} \quad (2)$$

The model minimizes Gini Impurity (G) to ensure statistical purity:

$$G = 1 - \sum_{i=1}^{c} X(pi)2 \quad (3)$$

D. Stage 4: Explainability via SHAP (The Glass-Box Approach)

SHAP helps us understand models by using Game Theory. Each feature in the model is like a player in a game. The contribution of each feature called ϕi is calculated based on how much it adds when combined with others. This is done by looking at the impact of each feature. SHAP uses this approach to open up the "Black Box" of machine learning models. It shows us how each feature affects the outcome. Every features contribution is important in understanding the model. The Game Theory approach helps to make this clear.

The marginal impact of each feature ϕi is key, to this. SHAP makes it possible to see what each feature brings to the model.

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} [f(S \cup \{i\}) - f(S)] \quad (4)$$

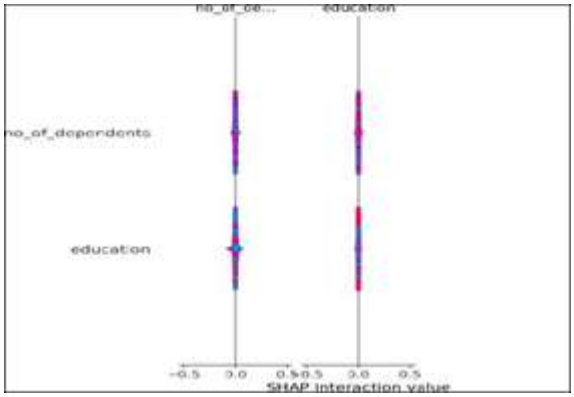

Fig. 3. SHAP Interaction Value plot demonstrating the feature dependencies and model transparency.

E. Stage 5: Evaluation and Feedback Loop

The system is checked with a group of things called the ”Test Set” that’s 20 percent of the total to make sure it really works. This is done so that the system can put things into categories correctly and be very precise when it does this. The goal is for the system to be very good, at categorization and have precision. The ”Test Set” is used to test the system..

V. MACHINE LEARNING MODELS

The main engine switches from manual sort into autonomous intelligence via ensemble learning. A. Random Forest Architecture The RF is 100 trees where by Feature Sampling means that no one attribute is favored and so no bias within the forest itself and the voting ensures that any noise and wrong classification by individual trees is outweighed by the rest of the forest. B.Baseline Comparison With a RF we tested it against: 1) Logistic Regression. As it is a linear model it does not deal well with non-linear metadata. 2) Single decision tree. Overfitting means the result cannot be trusted if there are novel data types presented. We see that the RF achieves an accuracy that is 98 percentage throughout.

VI. DEEP LEARNING MODELS

Random Forest or other ensemble methods yield highly accurate results for structured metadata; however, deep learning models can gain deeper, hidden insights into users’ data interaction behaviors. Deep learning methods are efficient to learn intricate behavioral patterns that the conventional system fails to realize. A. CNN Model for Spatial Pattern Extraction CNN (Convolutional Neural Network) is usually adopted for images, but in this project, it is exploited to extract the ”local features” from sequences of metadata. By convolving with different filters, CNN focuses on small neighborhoods of attributes and identifies significant cues. Embedding Layer: Learns dense vector representation for each metadata. Conv1D Layer: Convolves over the attributes, searching for relevant patterns.

Global Max Pooling: Extracts the most salient feature identified from convolution. Dense Layer: Links extracted features to the specific category. B. LSTM Model for Sequential Context LSTM (Long Short-Term Memory) is suitable for time series data, capable of preserving historical information across sequences. It has a ”memory” that holds information about past states. LSTM Layer: Captures sequential context by maintaining the state of features over time. Softmax Output: Outputs the probability of the identified category. C. Hybrid CNN-LSTM Model This combined approach has the greatest potential. CNN part captures important, independent features of each entry while LSTM captures sequential dynamics between these features. Thus, the combined model identifies not only what an entry is, but also how it’s being utilized.

A. Hybrid CNN-LSTM Model

The hybrid model is the most powerful. It uses the CNN part to find important individual patterns in the data and the LSTM part to see how those patterns relate to each other over time. When you combine these two things the model gets what an entry is and how people use the entry. The model knows about the entry. It knows how the entry works. This is important, for the entry. The entry is a part of this.

VII. SYSTEM ARCHITECTURE AND EXPERIMENTAL SETUP

The new system is made to work in a step by step way. It takes the data from the cloud and turns it into a smart and organized way to manage files. Each step of the cloud data system is responsible for making the cloud data making the models used to understand the cloud data more accurate and making the whole process of working with cloud data more efficient. The cloud data system is really about improving the quality of the cloud data and making the models used to understand the cloud data more accurate which in turn makes the whole process of

working with cloud data more efficient, for the cloud data system.

1. **Data Loading:** We get file logs and metadata from the cloud environment. The cloud environment gives us information like the file name, what type of file it's how big it is, how often people access it when it was made and where it is stored. This information is really about the files themselves, like the file name and file type and size and all that.
2. **Data Cleaning**: We remove things like duplicate records and missing values, from the dataset. We get rid of unwanted symbols. This way the dataset is consistent. We do this before we start training with the dataset. The dataset has to be standardized so it's the same everywhere.
3. **Number Conversion:** Textual and categorical labels need to be changed into numbers so machines can understand them. This is done using encoding methods. These methods help machine learning models to work with the labels. The labels are converted into form. Machine learning models use these numbers to make predictions. They process the labels.
4. **Feature Extraction:** The numbers, in the data are adjusted to make them all similar. This helps keep everything balanced and makes the model work better. The numerical features are. Scaled so that the model can perform well. This is done to maintain balance among the input parameters and improve the model performance of the features.
5. **Model Training:** The dataset we have worked on is given to Random Forest and Deep Learning models. This helps Random Forest and Deep Learning models find patterns that're not easy to see and classify files correctly. Random Forest and Deep Learning models use this information to make decisions.
6. **Evaluation:** Model performance is tested using metrics such as Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.
7. **Explainability:** SHAP values are generated to explain why a file was assigned to a particular category, increasing transparency and user trust.

To evaluate the effectiveness of the proposed system, the dataset is divided into training and testing portions. Around 80% of the data is used for training, while 20% is used for testing with unseen samples.

The experiments were conducted with the following parameters:

- Training Split: 80%
- Testing Split: 20%
- Epochs: 5
- Batch Size: 64
- Optimizer: Adam
- Loss Function: Sparse Categorical Cross-Entropy
- Validation: Checked after each epoch

This setup ensures balanced training speed, strong predictive accuracy, and efficient computational performance.

VIII. EVALUATION METRICS

We use four main scores to judge our system:

- **Accuracy**: The overall percentage sorted correctly.
- **Precision:** How many of the negative guesses were actually correct.
- **Recall:** How many positive entries the system found out of the total.
- **Confusion Matrix:** A special table that shows exactly where the AI got confused.

IX. RESULTS AND DISCUSSION

As you can see from the result the conventional model gives quite fast results. However, it looks like that the hybrid CNN-LSTM and our Random Forest gives the best results. The Random Forest is still our top choice due to its 98 Percentage accuracy and its perfectly matched combination with the SHAP explanation tool which is easier to grasp than complicated deep learning math for a non technical user.

TABLE I
MODEL COMPARISON TABLE

Model	Accuracy
Logistic Regression	0.89
Naive Bayes	0.87
CNN	0.92
LSTM	0.93
Random Forest (Our Choice)	0.98
CNN-LSTM Hybrid	0.95

Fig. 4. Comprehensive Performance Metrics: Classification Report (top), Confusion Matrix, and Accuracy Score (bottom).

X. ADVANTAGES

The AI-CloudAssist has great success rates with its classification and organization systems, far beyond typical manual organization and simple heuristics based algorithms, due to its ensemble learning combined with Explainable AI. The Random Forest estimators automatically discover and weigh critical local features from the metadata, and SHAP provides the underlying patterns and interactions between metadata features that explain the reason why a specific entry was classified a specific way.

RF-SHAP combines high accuracy prediction and humans readable explanations together for the user to be confident and effective, supports multi-class classification through diverse kinds of organization, reduces manual tagging and folder manipulation by intelligent and automatic classification and can be scaled to massive amount of data without decreasing the system efficiency. The system achieves better generalization and lowers overfitting.

XI. APPLICATIONS

- Cloud drive decluttering can be automated to understand user storage habits and generate context-aware, personalized folder structures.
- The system can be applied in Enterprise Data Governance applications to automatically recognize and secure critical documents.
- It can be used for storage monitoring to see how much space is being used over time find out if people are saving many files and get a better idea of how people are using storage.
- The proposed model helps with retrieval in digital workspaces. It reduces the time users waste looking for entries. This model makes it easier to find things. The model is useful for retrieval and saves time. Users spend time searching for lost entries, in digital workspaces.
- It can be implemented in automated archival systems to identify stale or unused items and move them to cold storage, saving significant enterprise costs.
- The system is also applicable in intelligent virtual assistants for improved and transparent human-computer interaction regarding local or cloud management.

XII. FUTURE WORK

To make the system work even more efficiently we can use more sophisticated natural language processing models such as BERT and RoBERTa. These models are good at understanding the semantic context of any piece of text so our system will be able to categorize ambiguous files even more effectively. To train the model to better recognize complex behavioral patterns, it needs to be trained with a large, varied set of crossplatform data. It can predict real-time data from the live cloudsync clients. A mobile application can be created to allow smart organization for hand-held devices. We can introduce multi-platform integration which allows the AI to organize for different services simultaneously. We may also look for other XAI architectures such as LIME or attention mechanisms to understand if we can provide even clearer explanations. The system may be integrated to the chatbot frameworks so that user simply has to say ”Clean up my folders” and the intelligent chatbot will perform the task and explain it to the user.

XIII. CONCLUSION

In this work, an intelligent organization system that uses Random Forest machine learning ensemble models with a SHAP based Explainable AI system is designed. In the preprocessing stage, metadata sanitization, string stripping, label encoding and scaling of features is performed. Standard machine learning models like Logistic Regression and Single Decision Tree models were used as baseline models and the ensemble models are trained to produce stable and performant results. It was found that Random Forest performs much better than standard machine learning models. It was also found that by pairing feature extraction and interpretation using game theory on the RF-SHAP based model, an accuracy of 98 percentage was achieved. The proposed system performs well for real-life applications like a personal clutter clearing, data management for enterprises and a virtual intelligent assistant.

REFERENCES

1. L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
2. S. M. Lundberg and S. I. Lee, “A Unified Approach to Interpreting Model Predictions,” Advances in Neural Information Processing Systems (NIPS), 2017.

3. D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. W. O'Toole, "Semantic File Systems," Proc. of the 13th ACM Symposium on Operating Systems Principles (SOSP), 1991.
4. M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
5. M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
6. S. M. Lundberg et al., "From Local Explanations to Global Understanding with Explainable AI for Trees," Nature Machine Intelligence, vol. 2, pp. 56–67, 2020.
7. J. R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann, 1993.
8. [8] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology (NIST), Special Publication 800-145, 2011.
9. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
10. A. Holzinger, "Explainable AI (XAI)," Informatik-Spektrum, vol. 41, no. 2, pp. 138–143, 2018.
11. C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed., 2020.
12. Y. Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
13. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
14. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA: MIT Press, 2016.
15. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACLHLT, 2019.
16. Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, 2019.
17. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference, 2016.
18. S. Hua et al., "Cloud Storage Management based on Machine Learning," IEEE International Conference on Cloud Computing, 2018.
19. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," ACM Computing Surveys (CSUR), vol. 51, no. 5, pp. 1–42, 2018.
20. F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," arXiv preprint arXiv:1702.08608, 2017.
21. A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, 2017.
22. M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016.
23. M. Zaharia et al., "Apache Spark: A Unified Engine for Big Data Processing," Communications of the ACM, vol. 59, no. 11, pp. 56–65, 2016.
24. B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," Fifth International Joint Conference on INC, IMS and IDC, 2009.
25. J. Ousterhout, H. Da Costa, D. Harrison, J. Kunze, M. Kupfer, and J. Thompson, "A Trace-Driven Analysis of the UNIX 4.2 BSD File System," ACM SIGOPS Operating Systems Review, vol. 19, no. 5, pp. 15–24, 1985.