ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

Credit Risk Assessment and Decision Support System with Comprehensive Visualization Using Python and Power BI

[1]M. Adarsh Kumar, [2]M. Niharika, [3]M. Bhavitha, [4]M. Venkata Kavya Sri, [5]M. Veda Sai Prakash Raja, [6]V. Pranav
[1,2,3,4,5] B. Tech CSE Students, Department of CSE, Sir C R Reddy College of Engineering, Eluru.
[6] Department of CSE, Sir C R Reddy College of Engineering, Eluru.

ABSTRACT

Credit risk assessment is really important in finance because it affects how banks decide on loans and manage risks to stay stable. With all the data piling up and people borrowing in more complicated ways, the old methods just do not cut it anymore for getting quick, accurate info. This project is about building a system that uses machine learning and some visualization tools to fix those issues. It pulls together different parts to predict if someone might default on a loan or not. I used Python for most of it, like Pandas to clean up the data and handle missing stuff, then Scikit-learn to build the models. Matplotlib helped with basic charts along the way. The dataset got processed to pull out features and standardize everything so the models could train properly. I tried out a few classifiers, Logistic Regression, Decision Trees, Random Forest, and Gradient Boosting. They all worked okay, but Random Forest seemed the best, with higher accuracy and it handles those tricky non-linear patterns in the data pretty well. It is robust too, I think. To make it easier for people to use, the results feed into Power BI for dashboards. Those show things like risk charts, how borrower's group up, trends in defaults, and some key metrics. Stakeholders can poke around without needing to know coding or anything technical. That part stands out because it makes complex predictions more approachable. The system cuts down on manual work and bias from humans, which is a big deal in evaluations. Loans get processed faster; decisions stay consistent. Since it is modular, it can scale up for bigger uses in real finance settings. It seems like there is room to improve, maybe add real-time data or put it on the cloud.

Keywords: Credit Scoring, Data Visualization, Machine Learning, Predictive Modelling, Risk Management.

I. INTRODUCTION

Credit risk is basically the chance that someone borrowing money won't pay it back, and that's a huge issue for banks and stuff. I mean, if they don't handle it right, they could lose a ton of money and even get in trouble with regulations. With all the data piling up these days and how fast things change in finance; the old ways of checking risks just don't seem to keep up anymore. Financial places need to figure out if borrowers are reliable to avoid big losses from defaults. If they mess up the assessment, it leads to instability, less money in reserves, and bad reps. A good system helps spot risky people early, makes smarter choices on loans, keeps the portfolio balanced, and follows the rules. This project is about building one that uses machine learning to predict defaults and shows visuals to help decide.

Traditional methods stick to fixed scores and people doing it by hand, which has problems. They can't handle big complicated data well, don't adjust when the economy shifts, take forever with all the manual work, and it's hard to understand the results. I think that's why we need something smarter that automates more. Now with digital finance, there's so much data available, and machine learning can dig into it. It processes huge amounts fast, finds patterns you wouldn't see otherwise, predicts better, and speeds up decisions without as much inconsistency. Things like logistic regression, decision trees, random forests, gradient boosting, they all look at borrower info and give solid risk estimates. And they update with new data, which fits the changing world.

Python is what I'm using to build this, because its got all these great tools for data stuff. Pandas handles cleaning and prepping the data, scikit-learn trains the models and checks how they do, and matplotlib makes charts. It all comes together for features, optimizing, evaluating, the whole thing. But just having models isn't enough, you need to see the insights clearly for decisions. That's where Power BI fits in, turning numbers into dashboards with graphs for trends,

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

KPIs to check quick, and filters to play around. It lets anyone, even non-tech folks, poke at the data and get what they need. The main goal here is to make a system that helps banks decide better on credit. Build models that predict risks accurately, beat the old methods at classifying, give visuals that make sense of it all, and make it work for real life, scalable and adaptable.

Overall, it should boost efficiency, cut down on biases from people, make things more transparent. Combining predictions with BI tools means faster, data-based calls. And it sets up for more, like real-time feeds, cloud stuff, other data types, keeping it useful as finance evolves. It feels like this could really change how they handle risks, though I'm not totally sure on every detail yet.

II. LITERATURE REVIEW

Credit risk stuff has always been a big deal in finance. It affects how banks lend money, make profits, and stay stable overall. I mean, without good ways to check if someone will pay back a loan, things can go wrong fast. Back in the day, people used simple stats like linear regression or logistic regression for this. They were easy to understand and set up, which is why banks liked them. You just plug in things like income, past payments, and debts to get a score. But honestly, these models miss a lot. Real life borrower decisions involve all sorts of tangled factors that don't fit straight lines. It seems like they work okay for basic cases, but not when everything connects in complicated ways.

That is where tree methods come in, like Decision Trees. They handle curves in the data and mix numbers with categories better. Plus, you can see the decisions like a flowchart, which makes it clearer. Still, they overfit sometimes, especially with tons of data. That pushes things toward ensembles, where you team up a bunch of trees.

Random Forest is one that stands out. It builds lots of trees and averages them, so it is tougher and handles big data without falling apart. Then there is Gradient Boosting, like XGBoost, which keeps tweaking to fix mistakes from before. These have shown really good results in predicting credit risks. Banks use them a lot now for accuracy. But accuracy is not everything. Lately, there is more talk about explaining how models work, because rules say you have to justify loan choices. So XAI tools like SHAP or LIME help break down what features matter most and why the model says yes or no. It feels like that bridges the gap between black box predictions and real understanding. Visualization helps too. Tools like Power BI let you make dashboards with charts and stuff that anyone can poke around in. No need to be a tech whiz to see patterns or key stats. Some systems do predictions well, others visualize nicely, but not many tie them together smoothly. I think that is the problem we need to fix. This project tries to build something that uses Python for the machine learning part and Power BI for showing results. It should make predictions sharper and easier to grasp, helping decisions happen faster. Not sure if it covers every angle yet, but it seems promising.

Table 1. Summary of Literature Review.

Author(s)	Key Contribution	Techniques Used	Relevance to Project
Bussmann et al.	XAI in fintech risk	SHAP, LIME, XAI methods	Improves transparency & trust
Choi et al.	Interpretable sequential model	RETAIN	Useful for credit history analysis
Cohen et al.	Explainable clustering	Shapley values	Transparent risk segmentation
De Lange et al.	XAI in banking	SHAP, LIME	Enhances model interpretability
Giudici et al.	Fairness in modeling	Shapley, Lore	Ensures unbiased decisions

III. PROBLEM STATEMENT AND OBJECTIVES

3.1 Problem Statement:

Credit risk assessment is really important for banks and stuff like that, because it helps figure out if someone can get a loan and if they will pay it back. Traditional ways of doing this mostly involve people looking at things by hand or using these old scoring models that do not change much. They are starting to fall short with all the complicated data out there now, and the amount of it just keeps growing. These systems stick to fixed rules and only a few details about

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

borrowers, so they miss out on how things connect in more complicated ways, like how some behaviors might not be straightforward.

One big issue is that they cannot handle huge piles of data very well. With everything going digital in finance, there is so much information coming in, from transaction logs to credit histories and even how people act with money. Going through all that manually takes forever, and it is easy to make mistakes or let personal biases sneak in. That ends up making decisions less accurate, I think, and not as trustworthy as they should be.

Financial situations change a lot, with economy shifts or new rules or just people behaving differently over time. Static models do not keep up with that, so predicting risks gets harder. Plus, it is tough to understand why a decision was made, which frustrates everyone involved. And visualization is another weak spot, where even if there are models, results come out in boring reports that you cannot really poke around in. Decision makers cannot spot patterns easily or watch risks as they happen.

So, there is this clear push for something smarter that uses data better, predicts risks accurately, cuts down on manual work, and shows everything in a way that makes sense quickly. This project tries to tackle that by mixing machine learning with tools for visuals, aiming for a system that works well and you can rely on.

3.2 Objectives:

The main goal here is to build a system for assessing credit risk that makes evaluations more precise, faster, and open about how it works. It pulls in modern machine learning and visualization to help with decisions based on actual data in finance places. A key part is creating models that can sort loan applicants by how risky they are. That means trying out algorithms like logistic regression, decision trees, random forests, and gradient boosting, then seeing which one does best by comparing them. Looking at past data on finances, the idea is to find patterns that show why some borrower's default.

Improving how accurate and solid the predictions are matters a lot. Using steps to clean data, build better features, and check models with things like accuracy, precision, recall, and ROC-AUC helps make sure it performs well. Institutions can then decide on loans smarter and lower defaults, it seems.

The visualization side uses Power BI to make an interactive setup. Turning all that complex stuff into dashboards that show risk spreads, groups of borrowers, and how models do lets people dig in and understand patterns better. Stakeholders can explore on their own, which is helpful. Automating as much as possible reduces the need for people to step in all the time, cutting bias too. It keeps things consistent and handles tons of applications without hassle.

Overall, the system should scale up and adapt easily for real use. It leaves room for adding things later, like pulling in live data or using cloud stuff or even explainable AI, so it stays useful in finance for a while. That part gets a bit messy to think about fully, but I am not totally sure how all those extensions would play out yet.

IV. METHODOLOGY / SYSTEM DESIGN

The system for assessing credit risk and helping with decisions is set up like a step-by-step process to get things right in predicting if someone might default on a loan. It starts with gathering data and goes through cleaning it up, building models, checking how they work, and then showing the results in a way that makes sense. Each part matters because raw financial stuff needs to turn into something useful for banks or lenders. Collecting the data comes first, and it's all about getting info on borrowers like their age, how much they make, past credit, loan size, if they paid back before, and other money details. I think this foundation is really important since without good diverse data, the predictions just won't hold up well. The quality here can make or break everything later on.

Once you have the data, preprocessing takes over to fix problems like missing bits or weird outliers that could mess up the models. You might fill in missing values or just drop them if they are not too big a deal, and same for outliers to keep things from getting skewed. Then there's transforming the data, creating new features that capture better what is going on, like debt to income or how much credit someone uses. Categorical stuff gets turned into numbers, and normalization happens so everything is on the same scale. It seems like this makes the data ready for machine learning

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

without too many headaches. For the models, after preprocessing, you build a few different ones to sort borrowers into risky or not. Logistic regression is a simple starting point, decision trees handle those non-straight-line patterns, random forest builds on that for better accuracy, and gradient boosting pushes performance even higher. They all train on past data to guess defaults based on the features. Using more than one lets you compare and pick what works best, which feels smart.

Evaluating them uses metrics to see if they are reliable. Accuracy shows how often its right overall, precision and recall deal with the imbalance in data where defaults are rarer, F1 score balances those two, and ROC AUC checks how well it separates the groups. A confusion matrix breaks down the hits and misses, like true positives or false alarms. In this setup, random forest comes out on top because it manages complex patterns without falling apart, though I might be oversimplifying why exactly. The visualization part exports everything to Power BI for dashboards that are interactive. Bar charts for risks, lines for trends, heatmaps to spot issues, and key indicators all help show patterns clearly. Linking Python to Power BI keeps the flow smooth, so predictions feed right into visuals. Stakeholders can poke around, filter what they want, and decide faster that way.

The whole workflow goes data collection, then preprocessing, training models, evaluating, picking the best one, and visualizing in Power BI. It's structured to be efficient, but sometimes the connections between steps feel a bit loose in practice.



Figure 1. System Architecture of Credit Risk Assessment System.

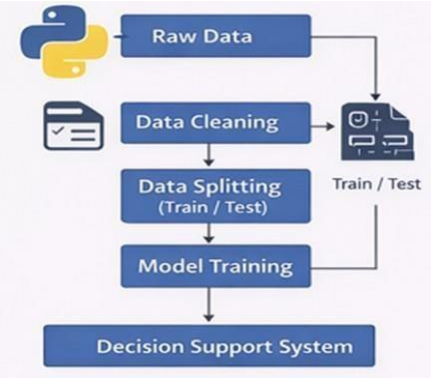


Figure 2. Machine Learning Workflow for Credit Risk Prediction.

V. TOOLS AND TECHNOLOGIES USED

The whole credit risk system depends on mixing different programming stuff with machine learning tools and ways to show the data visually. It helps process info fast, make good predictions, and let people see results in an easy way.
Python is basically the main thing for building the prediction parts. I think it's why a lot of data science projects use it, since its simple and you can bend it to fit what you need, plus there are tons of libraries to grab. You can handle big

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

piles of data without too much hassle, code up tricky algorithms, and hook it right into tools for charts or whatever. With Python, you do all the prep work on data, like cleaning it up or tweaking features, then train models and check how they do. Its free and people are always helping out online, so it fits for making apps that handle money stuff reliably, even if they get bigger over time. Pandas comes in for messing with the data itself. It's this library in Python that makes sorting through tables of info way easier. You import files, fix the messy parts, and shape it so its ready for deeper looks. It deals with blanks or weird values, lets you filter out junk or group things, and even build new columns that might help the models predict better. Data Frames in Pandas are super handy for financial numbers that pile up quick. Then there's NumPy, which is more about the math side. It crunches numbers on arrays that are huge, like for prepping data or during model building. Stuff runs faster because its optimized, especially with data that's got a lot of dimensions. It teams up with Pandas and others to keep everything moving smooth in the system. For the actual machine learning, Scikit-learn is the go to. It has all sorts of algorithms ready to use for training and testing models. In this setup, they pull in things like logistic regression or decision trees, even random forests for classifying risks. You split the data so some trains the model and some tests it, do cross validation to stop overfitting, and measure with scores like accuracy or recall. It's pretty straightforward, which is why it works well for predictions without getting too complicated.

Matplotlib helps draw out the pictures of what's going on. During analysis, you plot stuff to see how the model performs or spot patterns in the data. That makes it clearer for tweaking before you put it out there. The algorithms they use vary to get the predictions right. Logistic regression is simple to start with for yes no stuff. Decision trees pick up on bends in the data that straight lines miss. Random forest bunches a lot together for better accuracy and less overfitting, and gradient boosting keeps fixing where it went wrong before. Out of those, random forest seems to stand out, it handles messy patterns without falling apart too much. I might be oversimplifying, but yeah.

Table 2. Tools & Their Purpose.

Tools	**Purpose**
Python	Core programming
Pandas	Data preprocessing
NumPy	Numerical operations
Scikit-learn	Model building
Matplotlib	Visualization
Power BI	Dashboard creation

Power BI is for the dashboard part, turning all that output into something interactive and not boring to look at. Stakeholders can poke around charts that update live, see KPIs, or filter with slicers. To tie Python with Power BI, they just export to CSV or shove into a database, then load it up in BI for the visuals. That way data flows without stops, you monitor risks as they happen, and decisions get easier from the dashboards.

Overall, putting Python libraries with machine learning and Power BI builds this setup that's solid for assessing credit risks. Each piece handles its job, keeping things accurate and usable as it scales. The way they connect boosts how well it predicts and helps decide. Some parts get a bit messy though, like integrating everything perfectly.

VI. RESULTS AND DISCUSSION

In this part, I looked at how well the credit risk system actually works by testing different machine learning models on the data we prepared. The main goal was to figure out which one does the best job at sorting borrowers into risk categories, like high risk or low risk. I tried out a few algorithms, Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting. They all got trained on the dataset, and then I checked them with metrics like accuracy, precision, recall, F1-score, and ROC-AUC to see what they could do.

Logistic Regression was kind of my starting point because its straightforward and easy to understand what it does. It did okay, but I noticed it struggles with the tricky non-linear stuff in the data, you know, the relationships that aren't just straight lines. Then Decision Trees came in, and they handled those patterns better, more flexible somehow. Still, they overfit a lot, especially with all the variety in this big dataset, which makes predictions shaky sometimes.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

Random Forest took that idea further by using a bunch of decision trees together, which cuts down on the variance and makes things more stable. It just performed way better overall, more accurate than the single trees. Gradient Boosting builds on errors step by step to improve, which sounds great, but it needs a lot of tweaking to not overfit too. After running everything, Random Forest stood out as the top one here. It hit the highest accuracy and balanced out the metrics nicely. I think its good at dealing with how variables interact in complicated ways without overfitting as much. For the metrics, accuracy tells you how often the model gets things right in general. Precision is about not approving risky people by mistake, so the true positives out of all positives it predicts. Recall makes sure it catches the real defaulters, which is huge for avoiding losses. F1-score mixes those two, helpful since the data is imbalanced with more non-defaulters probably. And ROC-AUC looks at how well it separates the risks at different cutoff points. Random Forest scored high on all of them, which makes me feel like its reliable enough for this.

I also used a confusion matrix to dig deeper into the classifications. It breaks down true positives, like spotting actual defaulters right, true negatives for the safe ones, false positives where it flags safe borrowers as risky, and false negatives, the ones it misses who are actually bad. The model we picked keeps those errors low, especially false negatives, because letting a high-risk borrower slip through could cost a lot in finances. That part seems critical, though I am not totally sure if every scenario covers it perfectly. To make sense of all this, we hooked up the results to Power BI for dashboards that you can interact with. It shows things like charts of risk distribution, how many high versus low risk borrowers there are. Then segmentation by demographics or finances, and trends in defaults over time. KPIs too, like approval rates or default percentages summed up. It's nice because you can filter and poke around, spotting patterns that might hide in plain numbers. That dynamic stuff helps stakeholders see the big picture without getting lost.

Overall, these machine learning methods beat the old traditional ways for credit checks, I think, since they handle big data and find those hidden connections for better predictions. Adding Power BI makes it even clearer with visuals that are easy to explore. Together, it speeds up decisions, cuts out manual errors and biases, and keeps things consistent for loan apps. The system does what it set out to, proving Random Forest works well and that mixing ML with tools like this is key, though some parts might still need refining depending on new data.

Table 3. Model Comparison Table.

Model	Accuracy	Precisio	Recall	F1Score	Model
Logistic egression	78%	75%	72%	73%	Logistic egression
Decision Tree	80%	78%	76%	77%	Decision Tree
Random Forest	**88%**	**86%**	**85%**	**85%**	Random Forest
Gradient Boosting	85%	83%	82%	82%	Gradient Boosting

VII. CONCLUSION & FUTURE SCOPE

CONCLUSION

This project shows how mixing machine learning with business tools can really help out in checking credit risks. It kind of fixes the problems with old ways, like how they are not very flexible or take too much time with people doing everything by hand. The system we built uses data to make things automatic and faster, which seems to make decisions more accurate too. Starting off, the usual methods have issues, such as not handling data well or changing when needed. So, to deal with that, the approach here involves getting the data ready first, picking out important features, building the models, and then checking how they do. Python makes it easier with its libraries to crunch numbers and spot patterns in financial stuff that are hard to see otherwise. We tried out a few algorithms for the machine learning part, things like logistic regression, decision trees, random forest, and gradient boosting. Comparing them, random forest came out on top, I think, because it is pretty accurate and handles messy relationships in the data without falling apart. To make sure, we looked at scores like accuracy, precision, recall, F1, and that ROC-AUC thing, and it held up okay in different tests. One part that stands out is adding Power BI to show the results visually. With dashboards that you can interact with, people can see what the models say about risks without getting lost in numbers. Charts and key numbers, plus filters to play around, make it simpler for everyone, even if they are not tech savvy. That connects the heavy analysis to actual

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

use in decisions, which feels important. It also cuts down on manual work for evaluating loans, so errors from people or biases drop, and applications get processed quicker. Financial places can handle tons of data this way and stay consistent. The design is modular, meaning you can add more stuff later, like new data or better techniques, so it scales up for real life. Overall, this setup with predictions and visuals helps in assessing risks better, improving how clear and easy it is to use. In a world full of data now, tools like this seem valuable for banks, though some parts might need tweaking for different situations.

FUTURE SCOPE

The credit risk system they proposed looks pretty solid in how it works and could actually be used in practice. But there is room to make it even better, especially with all the new tech coming out and more data everywhere. It might get more accurate and handle bigger stuff, which would help in real situations. I think one thing that stands out is adding explainable AI, like those SHAP or LIME methods. Right now, the models just spit out predictions, but explaining why they decide something could build more trust. That seems important for banks and regulations, sort of to avoid ethical issues or whatever. Stakeholders need to know what factors are really pushing the credit decisions. On top of that, deploying it on cloud platforms might improve everything. Accessibility would go up since people could log in from anywhere, and it would scale better for lots of users. Plus, handling huge data loads and linking to other systems sounds practical. I am not totally sure how seamless that integration would be, but it seems worth trying. Extending to alternative data, like social media or mobile payments, could give deeper looks at people without much credit history. Transaction patterns might reveal behaviors that regular data misses. This part gets a bit messy, because privacy concerns could come up, but it probably boosts accuracy overall. Deep learning models are another option to consider. They might catch complex patterns in big datasets better than the standard machine learning ones used now. Although the traditional approaches do okay, neural networks could push performance higher in some cases. Maybe not necessary for everything, but it is something to explore.

Finally, adding automated features for loan approvals or interest rate ideas based on risk would make the whole thing more like a full decision tool. That transforms it from just assessment to supporting the entire process.

REFERENCES

[1] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2009.
[2] L. Breiman, Random Forests, Machine Learning Journal, vol. 45, no. 1, pp. 5-32, 2001.
[3] J. Brownlee, Machine Learning Mastery with Python, Machine Learning Mastery, 2016.
[4] F. Chollet, Deep Learning with Python, Manning Publications, 2017.
[5] Scikit-learn Documentation, Machine Learning in Python, Available: https://scikit-learn.org
[6] Microsoft Power Bi Documentation Available at: https://learn.microsoft.com/en-us/power-bi/
[7] Biecek, Przemysław, et al. (2021). Enabling Machine Learning Algorithms for Credit Scoring – Explainable Artificial Intelligence (XAI) Methods for Clear Understanding Complex Predictive Models. arXiv:2104.06735.
[8] Bussmann, Niklas, et al. (2020). Explainable AI in Fintech Risk Management. Frontiers in Artificial Intelligence, 3: 26.
[9] Choi, Edward, et al. (2017). RETAIN: An Interpretable Predictive Model for Healthcare Using Reverse Time Attention Mechanism. NIPS'16 Proceedings.
[10] Cohen, Joseph, et al. (2023). Shapley-Based Explainable AI for Clustering Applications in Fault Diagnosis and Prognosis. arXiv:2303.14581.
[11] de Lange, Petter Eilif, et al. (2022). Explainable AI for Credit Assessment in Banks. Journal of Risk and Financial Management, 15: 556.
[12] Demajo, Lara Marie, et al. (2020). Explainable AI for Interpretable Credit Scoring. Computer Science and Information Technology.
[13] Doersch, Carl (2021). Tutorial on Variational Autoencoders. arXiv:1606.05908v3.
[14] European Union (2016). Regulation (EU) 2016/867 of the European Central Bank on the Collection of Granular Credit and Credit Risk Data (ECB/2016/13). Official Journal of the European Union, CELEX:32016R0867.
[15] Fahner, Gerald (2018). Developing Transparent Credit Risk Scorecards More Effectively: An Explainable Artificial Intelligence Approach. Data Analytics 2018.
[16] Galindo, Jorge, & Tamayo, Pablo (2000). Credit Risk Assessment Using Statistical and Machine Learning: Basic Methodology and Risk Modeling Applications. Computational Economics, 15: 107–143.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

[17] Giudici, Paolo, & Raffinetti, Emanuela (2020). Shapley-Lorenz Decompositions in Explainable Artificial Intelligence. SSRN Electronic Journal, 1: 1–15.
[18] Gramegna, Alex, & Giudici, Paolo (2021). SHAP and LIME: An Evaluation of Discriminative Power in Credit Risk. Frontiers in Artificial Intelligence, 4: 752558.
[19] Guan, Charles, et al. (2023). Responsible Credit Risk Assessment with Machine Learning and Knowledge Acquisition. Human-Centric Intelligent Systems, 3: 232–243.
[20] Hand, David J. (2009). Measuring Classifier Performance: A Coherent Alternative to the Area Under the ROC Curve. Machine Learning, 77: 103–123.
[21] Heng, Yi, & Subramanian, Preethi (2022). A Systematic Review of Machine Learning and Explainable Artificial Intelligence (XAI) in Credit Risk Modelling. Springer, pp. 596–614.