VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018

SKILL DEVELOPMENT ACTIVITY REPORT ON

DATA SCIENCE AND MANAGEMENT(25MCS102)

TOPIC: CLASSIFICATION IN FINANCE DOMAIN

Submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Name**: B NAYANA**

USN**: 1KS25SCS01**

Under the guidance of

Dr. Rekha B Venkatapur

Professor and Head, CSE

Department of Computer Science and Engineering

(An Autonomous Institution under VTU)

#14, Raghuvanahalli, Kanakapura Road, Bengaluru- 560109

2025-2026

K. S. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU)

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the demonstration on Mini Project work entitled **"CLASSIFICATION IN FINANCE DOMAIN"** carried out by **Name: B NAYANA Bonafide student** of First Semester M.Tech CSE of **K.S. Institute of Technology, an autonomous Institution under Visvesvaraya Technological University, Belagavi,** during the year 2025-26. It is certified that all corrections/suggestions indicated for Internal Assessment of Data Science and Management (25MCS102) Course assignment have been incorporated in the report deposited in the departmental library. The Mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said degree for the First semester.

Dr. Rekha. B. Venkatapur
Prof & HOD, CS & E Department

Dr. Dilip Kumar K
Principal/Director, KSIT

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude to my college **K. S. Institute of Technology** for providing a supportive environment that enabled me to successfully complete this Skill Development Activity and prepare this report.

I also extend my heartfelt thanks to the **Management of K. S. Institute of Technology** for providing all the necessary resources required for the Skill Development Activity.

My sincere thanks to **Dr. Dilip Kumar K**, **Principal and Director,** K. S. Institute of Technology for his continuous support and encouragement.

I am grateful to my PG Coordinator, **Dr. Krishna Gudi, Associate professor,** Department of CSE for his guidance and support throughout the course of this work.

I would also like to extend my gratitude to **Dr. Rekha B Venkatapur, Professor and Head,** Department of Computer Science and Engineering for all the support forwarded to me in completing this Skill Development Activity successfully.

Finally, I express my appreciation to my family and friends for their Moral support and help prodding me to complete the Skill Development Activity.

B Nayana

1KS25SCS01

ABSTRACT

In the modern financial ecosystem, the increasing volume of transactional and customer data has created both opportunities and challenges for financial institutions. Accurate risk assessment is essential for minimizing financial losses, improving credit evaluation processes, detecting fraudulent activities, and ensuring regulatory compliance. Traditional rule-based and manual assessment methods are often inefficient when dealing with large-scale, complex, and dynamic financial datasets.

The system utilizes supervised learning algorithms to classify financial data into predefined risk categories such as low risk, medium risk, and high risk. Multiple classification models were implemented and compared to identify the most effective predictive approach. Model performance was evaluated using appropriate metrics such as precision, recall, F1-score, and ROC-AUC, ensuring balanced evaluation, particularly in scenarios involving imbalanced datasets where high-risk cases are relatively rare but critically important.

To enhance usability and practical applicability, the trained machine learning model was deployed using Streamlit to create an interactive web-based application. This interface allows users to input relevant financial parameters and receive real-time risk predictions. The integration of predictive modeling with a user-friendly dashboard transforms the system into a decision-support tool capable of assisting financial analysts, lending institutions, and risk managers.

The Financial Risk Intelligence System demonstrates how data science and machine learning can be effectively applied in the finance domain to automate risk classification, improve operational efficiency, and support informed decision-making. Furthermore, the project emphasizes the importance of model interpretability, scalability, and continuous monitoring to ensure long-term reliability in changing economic environments. Overall, the proposed system provides a scalable, adaptable, and intelligent framework for financial risk prediction, contributing to the advancement of technology-driven financial risk management solutions.

CONTENTS

INTRODUCTION

The rapid advancement of data science and machine learning has significantly transformed the financial industry. Financial institutions such as banks, insurance companies, investment firms, and financial technology (fintech) organizations generate enormous volumes of data from daily transactions, customer interactions, credit histories, trading activities, and digital platforms. The ability to analyze this data effectively has become a critical factor in improving operational efficiency, managing financial risks, ensuring regulatory compliance, and maintaining competitive advantage. As a result, predictive analytics techniques—particularly classification methods—have become central to modern financial decision-making systems.

Classification is a supervised machine learning technique that involves assigning data instances into predefined categories based on historical labeled data. In the finance domain, classification problems typically involve categorical outcomes such as “default” or “non-default,” “fraudulent” or “legitimate,” “approve” or “reject,” and “high risk,” “medium risk,” or “low risk.” These predictions directly influence high-stakes financial decisions, including loan approvals, credit limit assignments, fraud prevention, insurance underwriting, customer retention strategies, and investment recommendations. Because these decisions have significant financial and regulatory implications, the development of accurate and reliable classification models is essential.

One of the most prominent applications of classification in finance is credit risk modeling. Financial institutions use classification algorithms to estimate the probability that a borrower will fail to repay a loan. This probability of default helps lenders determine whether to approve or reject loan applications and how to price financial products. Similarly, fraud detection systems rely heavily on classification techniques to identify suspicious transaction patterns in real time. With the increasing growth of digital payments and online banking, fraud detection has become a major priority for financial institutions worldwide. Other important applications include anti-money laundering detection, customer churn prediction, customer segmentation, bankruptcy prediction, algorithmic trading signal classification, and insurance claim categorization.

Despite its widespread importance, classification in finance presents several unique challenges. Financial datasets are often highly imbalanced; for example, fraudulent transactions typically represent only a small fraction of total transactions. In such cases, traditional accuracy metrics may be misleading, making it necessary to rely on more informative performance measures such as precision, recall, F1-score, ROC-AUC, Gini coefficient, and the Kolmogorov–Smirnov (KS) statistic. Additionally, financial institutions operate under strict regulatory frameworks that require model transparency, fairness, and explainability. Decisions made by automated systems must be interpretable and justifiable to regulators and customers. Consequently, there is a strong emphasis on explainable artificial intelligence (XAI) techniques in financial modeling.

Another significant challenge is concept drift, where changes in economic conditions, market behavior, or customer patterns reduce the effectiveness of previously trained models. Economic crises, policy changes, inflation, and technological innovations can alter financial behavior over time, requiring continuous monitoring, validation, and periodic retraining of classification models. Ensuring model robustness and stability in dynamic environments is therefore a crucial aspect of financial data science.

Various algorithms are used to address classification problems in finance. Traditional statistical approaches such as logistic regression remain widely adopted due to their simplicity, interpretability, and regulatory acceptance. However, more advanced machine learning methods—including decision trees, random forests, gradient boosting techniques (such as XGBoost and LightGBM), support vector machines, and neural networks—are increasingly utilized for their superior predictive performance. The choice of model depends on factors such as data characteristics, business requirements, interpretability needs, and computational efficiency.

Recent developments in artificial intelligence have further enhanced the role of classification in finance. Deep learning models are being applied to large-scale transaction data and alternative data sources, including text, social media, and behavioral data. Moreover, the integration of big data technologies, cloud computing, and real-time analytics has enabled financial institutions to deploy classification models at scale. At the same time, growing awareness of ethical AI and algorithmic bias has encouraged the adoption of fairness-aware machine learning frameworks to prevent discriminatory outcomes.

Problem Statement

Financial institutions face increasing challenges in managing risk, detecting fraud, and making accurate credit decisions in a data-rich and highly regulated environment. Traditional rule-based systems are often insufficient to handle complex and evolving financial patterns. Therefore, there is a need for robust, accurate, interpretable, and scalable classification models that can support automated decision-making while ensuring compliance with regulatory standards.

Objectives of the Project

The primary objective of this project is to design and develop a Financial Risk Intelligence System that utilizes machine learning classification techniques to predict financial risk accurately and efficiently. The system aims to transform raw financial data into meaningful insights that can support intelligent decision-making in areas such as credit risk assessment, loan approval evaluation, and financial stability analysis.

Another key objective is to study and implement supervised classification algorithms suitable for financial datasets and compare their performance using appropriate evaluation metrics. The project seeks to identify the most effective model for predicting financial risk categories while ensuring reliability, robustness, and generalization capability. Special emphasis is placed on handling challenges such as imbalanced data, feature selection, and model validation to ensure realistic and accurate predictions.

The project also aims to perform comprehensive data preprocessing and feature engineering to enhance the predictive power of the models. This includes cleaning inconsistent data, handling missing values, transforming categorical variables, and constructing meaningful financial indicators that significantly influence risk assessment.

An additional objective is to evaluate model performance using financial domain-specific metrics beyond simple accuracy. Metrics such as precision, recall, F1-score, and ROC-AUC are considered to ensure that the system performs effectively, particularly in high-risk detection scenarios where misclassification can result in financial losses.

Furthermore, the project seeks to develop a user-friendly and interactive web application using Streamlit to deploy the trained model. This objective focuses on bridging the gap between theoretical machine learning models and real-world applications by enabling users to input financial parameters and receive real-time risk predictions through an accessible interface.

The system also aims to demonstrate scalability and adaptability by creating a modular architecture that allows future enhancements, integration with databases, and potential cloud deployment. Ensuring interpretability and transparency of predictions is another important objective, as financial systems require explainable and justifiable decision-making processe

TOOLS AND TECHNOLOGIES

1. Hardware Requirements

The Financial Risk Intelligence System was developed and tested on the following hardware configuration:

Development System Specifications

- Processor: Intel Core i5 / i7 (or equivalent)
- RAM: Minimum 8 GB (16 GB recommended for handling large financial datasets)
- Storage: 256 GB SSD or higher
- System Architecture: 64-bit system
- Internet Connection: Required for package installation and deployment (if cloud-based)

The system does not require high-end computing resources for traditional machine learning models. However, increased RAM improves performance when processing large transaction datasets.

2. Software Requirements

Operating System

- Windows10/11
- Ubuntu
- macOS

The application is platform-independent and can run on any operating system that supports Python.

Programming Language

- Python 3.8 or above

  Python was used for:

- Data preprocessing
- Feature engineering
- Model training
- Risk prediction
- Backend integration with the user interface

3. Frameworks and Libraries

Frontend & Web Application Framework

Streamlit

Streamlit was used to develop an interactive web-based interface for the Financial Risk Intelligence System. It enabled:

- Real-time risk prediction
- User input forms for financial parameters
- Visualization of risk scores
- Display of classification results
- Easy deployment as a web application

Data Processing Libraries

- NumPy – Numerical computations
- Pandas – Data manipulation and preprocessing

Machine Learning Libraries

- Scikit-learn – Implementation of classification models such as:
  - Logistic Regression
  - Decision Trees
  - Random Forest
  - Support Vector Machine
- XGBoost / LightGBM (if used)

Data Visualization

- Matplotlib
- Seaborn
- Streamlit built-in visualization components

4. Database (If Applicable)

- MySQL / PostgreSQL (for structured financial records)
- CSV-based dataset (for model training and testing)

5. Execution Environment

The system operates in the following environment:

1. Data is loaded and preprocessed using Python.
2. The trained classification model is saved using Pickle / Joblib.
3. The Streamlit application loads the trained model.
4. Users input financial parameters via the web interface.
5. The model generates a risk prediction (e.g., Low / Medium / High Risk).
6. The predicted risk score is displayed on the dashboard.

   The application can be executed using: streamlit run app.py

IMPLEMENTATION

```
import streamlit as st

import numpy as np

import random

from credit_risk import credit_risk_model

from decision_engine import decision_engine


# ---------------- PAGE CONFIG ----------------

st.set_page_config(

    page_title="Financial Risk Intelligence System",

    page_icon="💰",

    layout="centered"

)


# ---------------- HEADER ----------------

st.markdown(

    """

    <h1 style='text-align: center;'>💰 Financial Risk Intelligence System</h1>

    <p style='text-align: center; font-size: 18px;'>

    AI-powered credit risk assessment with system-estimated market conditions
```

```
    </p>
    """,
    unsafe_allow_html=True
)

st.markdown("---")

# ---------------- LOAD MODELS ----------------
@st.cache_resource
def load_models():
    credit_model, scaler, feature_count, employment_index = credit_risk_model()
    return credit_model, scaler, feature_count, employment_index

with st.spinner("🔄 Loading AI models..."):
    credit_model, scaler, feature_count, employment_index = load_models()

st.success("✅ Models loaded successfully")

# ---------------- INPUT SECTION ----------------
st.markdown("## 🧾 Customer Information")
```

```
col1, col2 = st.columns(2)

with col1:
    age = st.number_input("Age", min_value=18, max_value=75, value=30)
    duration = st.number_input("Loan Duration (months)", min_value=6,
value=24)

with col2:
    credit_amount = st.number_input("Credit Amount", min_value=100,
value=5000)
    employment_option = st.selectbox(
        "Employment Duration",
        ["Unemployed", "< 1 year", "1–4 years", "≥ 4 years"]
    )

employment_mapping = {
    "Unemployed": 0,
    "< 1 year": 1,
    "1–4 years": 2,
    "≥ 4 years": 3
```

```
}
employment_value = employment_mapping[employment_option]


st.markdown("---")


# ---------------- RUN BUTTON ----------------
run = st.button("🔍 Run Risk Analysis", use_container_width=True)


# ---------------- ANALYSIS ----------------
if run:
    # Create feature vector
    user_features = np.zeros((1, feature_count))
    user_features[0, 0] = age
    user_features[0, 1] = credit_amount
    user_features[0, 2] = duration
    user_features[0, employment_index] = employment_value


    user_features = scaler.transform(user_features)


    credit_prediction = credit_model.predict(user_features)[0]
```

```
    # Hybrid override for extreme cases
    if credit_amount >= 25000 and duration >= 48 and employment_value == 0:
        credit_prediction = 1

    # ---------------- MARKET SENTIMENT ----------------
    market_contexts = {
        1: "Markets are showing strong growth with stable economic indicators.",
        0: "Markets remain stable with no major fluctuations.",
       -1: "Markets are under stress due to inflation and economic slowdown."
    }

    sentiment_prediction = st.selectbox(
    "Market Condition",
    [1, 0, -1],
    format_func=lambda x: {
        1: "Positive Market",
        0: "Neutral Market",
       -1: "Negative Market"
    }[x]
)
    market_summary = market_contexts[sentiment_prediction]
```

```
    decision = decision_engine(
    credit_prediction,
    sentiment_prediction,
    age,
    employment_value
)

    st.markdown("---")
    st.markdown("## 📊 Risk Assessment Result")

    # ---------------- RESULT CARDS ----------------
    col_risk, col_market = st.columns(2)

    with col_risk:
        if credit_prediction == 0:
            st.success("✅ **LOW CREDIT RISK**")
            st.caption("Customer is likely to repay the loan.")
        else:
            st.error("🚨 **HIGH CREDIT RISK**")
```

```
        st.caption("Customer has a higher probability of default.")


with col_market:
    if sentiment_prediction == 1:
        st.success("📈 **Positive Market**")
    elif sentiment_prediction == -1:
        st.error("📉 **Negative Market**")
    else:
        st.info("➖ **Neutral Market**")


st.markdown("#### 📰 Market Insight")
st.write(market_summary)


st.markdown("---")


# ---------------- FINAL DECISION ----------------
st.markdown("## 🧠 Final Decision")


if "REJECT" in decision:
    st.error(f"❌ **{decision}**")
```

```
    elif "APPROVE" in decision:
        st.success(f"✅ **{decision}**")
    else:
        st.warning(f"⚠️ **{decision}**")
```

-----------credit.risk.py----------

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
def credit_risk_model():

    # ---------------- Load Dataset ----------------
    df = pd.read_csv("credit_data.csv")
    print("\nTotal Rows in Dataset:", len(df))

    # ---------------- Encode Categorical Columns ----------------
    encoders = {}
    for col in df.select_dtypes(include='object').columns:
        encoder = LabelEncoder()
```

```
    df[col] = encoder.fit_transform(df[col])
    encoders[col] = encoder


# ---------------- Target Cleaning ----------------
df["kredit"] = df["kredit"].astype(int)
df = df[df["kredit"].isin([1, 2])]
df["kredit"] = df["kredit"].map({1: 0, 2: 1})


print("\nClass Distribution:")
print(df["kredit"].value_counts())


# ---------------- Features & Target ----------------
X = df.drop("kredit", axis=1)
y = df["kredit"]


feature_columns = X.columns.tolist()
employment_index = feature_columns.index("beszeit")


# ---------------- Train Test Split (STRATIFIED) ----------------
X_train, X_test, y_train, y_test = train_test_split(
    X,
```

```
    y,
    test_size=0.3,
    random_state=42,
    stratify=y
)


# ---------------- Scaling (After Split) ----------------
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


# ---------------- Controlled Random Forest (Prevent Overfitting) ----------------
model = RandomForestClassifier(
n_estimators=100,
max_depth=6,
min_samples_split=10,
min_samples_leaf=5,
class_weight='balanced',   # IMPORTANT FIX
random_state=42
)
```

```
model.fit(X_train, y_train)


# ---------------- Evaluation ----------------
predictions = model.predict(X_test)


accuracy = accuracy_score(y_test, predictions) * 100
precision = precision_score(y_test, predictions) * 100
recall = recall_score(y_test, predictions) * 100
f1 = f1_score(y_test, predictions) * 100


print("\n--- Credit Risk Model Performance ---")
print(f"Accuracy  : {accuracy:.2f}%")
print(f"Precision : {precision:.2f}%")
print(f"Recall    : {recall:.2f}%")
print(f"F1-Score  : {f1:.2f}%")


# ---------------- Feature Importance ----------------
print("\n--- Feature Importances ---")
for feature, importance in zip(feature_columns, model.feature_importances_):
    print(f"{feature}: {importance:.4f}")
```

```
    return model, scaler, X.shape[1], employment_index



if __name__ == "__main__":
    credit_risk_model()
    import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score



def credit_risk_model():


    # ---------------- Load Dataset ----------------
    df = pd.read_csv("credit_data.csv")
    print("\nTotal Rows in Dataset:", len(df))


    # ---------------- Encode Categorical Columns ----------------
    encoders = {}
```

```
for col in df.select_dtypes(include='object').columns:
    encoder = LabelEncoder()
    df[col] = encoder.fit_transform(df[col])
    encoders[col] = encoder


# ---------------- Target Cleaning ----------------
df["kredit"] = df["kredit"].astype(int)
df = df[df["kredit"].isin([1, 2])]
df["kredit"] = df["kredit"].map({1: 0, 2: 1})


print("\nClass Distribution:")
print(df["kredit"].value_counts())


# ---------------- Features & Target ----------------
X = df.drop("kredit", axis=1)
y = df["kredit"]


feature_columns = X.columns.tolist()
employment_index = feature_columns.index("beszeit")


# ---------------- Train Test Split (STRATIFIED) ----------------
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.3,
    random_state=42,
    stratify=y
)


# ---------------- Scaling (After Split) ----------------
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


# ---------------- Controlled Random Forest (Prevent Overfitting) ----------------
model = RandomForestClassifier(
    n_estimators=100,
    max_depth=6,
    min_samples_split=10,
    min_samples_leaf=5,
    random_state=42
)
```

```
model.fit(X_train, y_train)


# ---------------- Evaluation ----------------
predictions = model.predict(X_test)


accuracy = accuracy_score(y_test, predictions) * 100
precision = precision_score(y_test, predictions) * 100
recall = recall_score(y_test, predictions) * 100
f1 = f1_score(y_test, predictions) * 100


print("\n--- Credit Risk Model Performance ---")
print(f"Accuracy  : {accuracy:.2f}%")
print(f"Precision : {precision:.2f}%")
print(f"Recall    : {recall:.2f}%")
print(f"F1-Score  : {f1:.2f}%")


# ---------------- Feature Importance ----------------
print("\n--- Feature Importances ---")
for feature, importance in zip(feature_columns, model.feature_importances_):
    print(f"{feature}: {importance:.4f}")
```

```
    return model, scaler, X.shape[1], employment_index



if __name__ == "__main__":
    credit_risk_model()
---------decision.engine.py---------
defdecision_engine(credit_prediction,sentiment_prediction,age,
employment_value):
    # Rule 1: Age > 50 AND Unemployed → Reject
    if age > 50 and employment_value == 0:
        return "REJECT LOAN"
    # Rule 2: High risk (others) → Manual review
    elif credit_prediction == 1:
        return "MANUAL REVIEW REQUIRED"
    # Rule 3: Low risk + positive market → Approve
    elif credit_prediction == 0 and sentiment_prediction == 1:
        return "APPROVE LOAN"
    # Everything else → Manual
    else:
        return "MANUAL REVIEW REQUIRED"
```

-------------main.py------------

```
from credit_risk import credit_risk_model

from sentiment_analysis import sentiment_model

from decision_engine import decision_engine

import numpy as np


print("\n🔍 Training Credit Risk Model...")

credit_model = credit_risk_model()


print("\n📰 Loading Financial Sentiment Model...")

sentiment_predictor = sentiment_model()


# --- SAMPLE INPUT ---

sample_customer_data = np.random.rand(1, 20)  # mock input

sample_news = "The company reported strong quarterly earnings and market optimism."


# Predict credit risk

credit_prediction = credit_model.predict(sample_customer_data)[0]
```

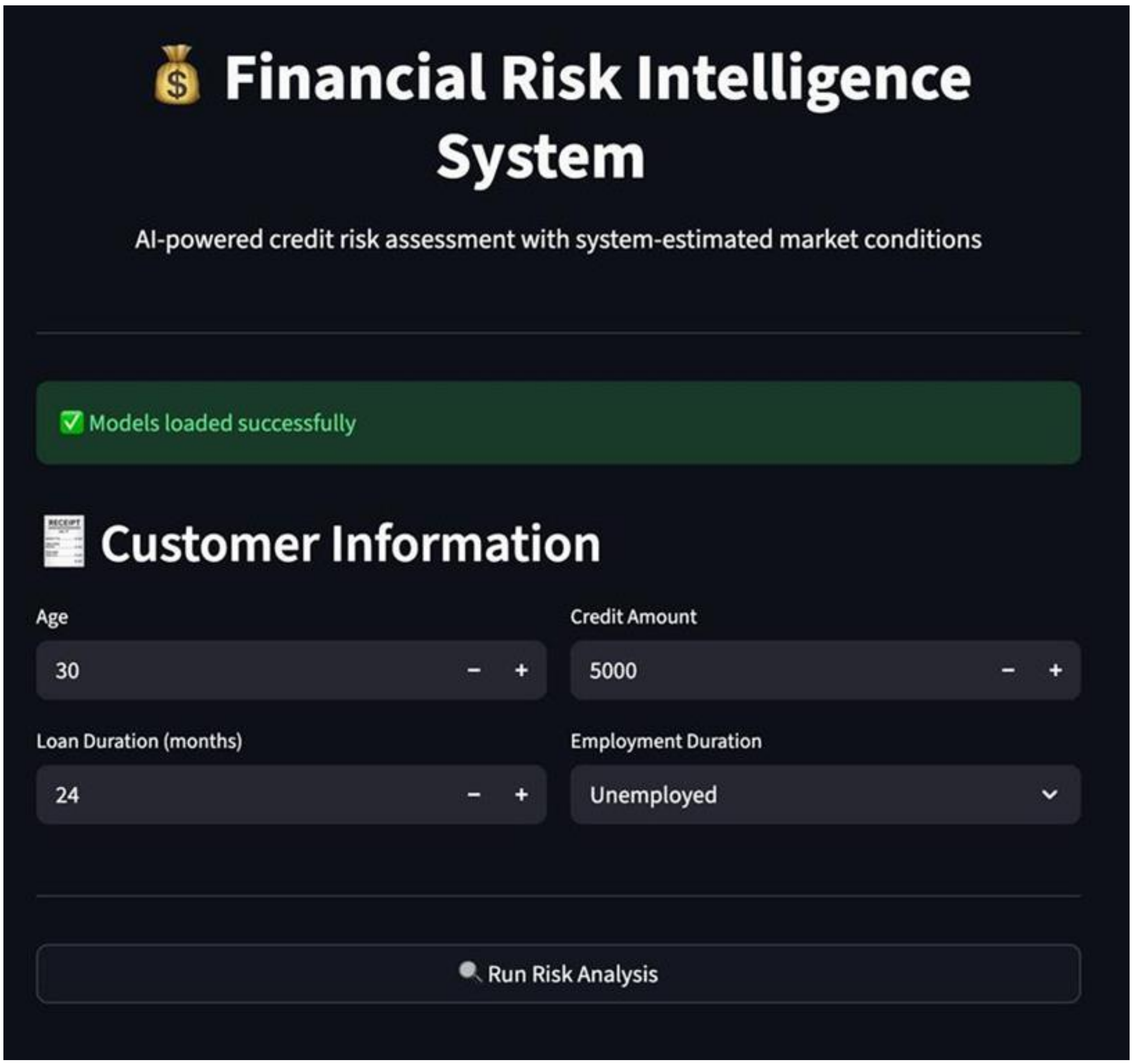
```
# Predict sentiment

sentiment_prediction = sentiment_predictor(sample_news)


# Final Decision

decision = decision_engine(credit_prediction, sentiment_prediction)


print("\n--- FINAL AI DECISION ---")

print(f"Credit Risk Class : {credit_prediction}")

print(f"Market Sentiment  : {sentiment_prediction}")

print(f"Decision          : {decision}")
```

-----------sentiment.analysis.py----------

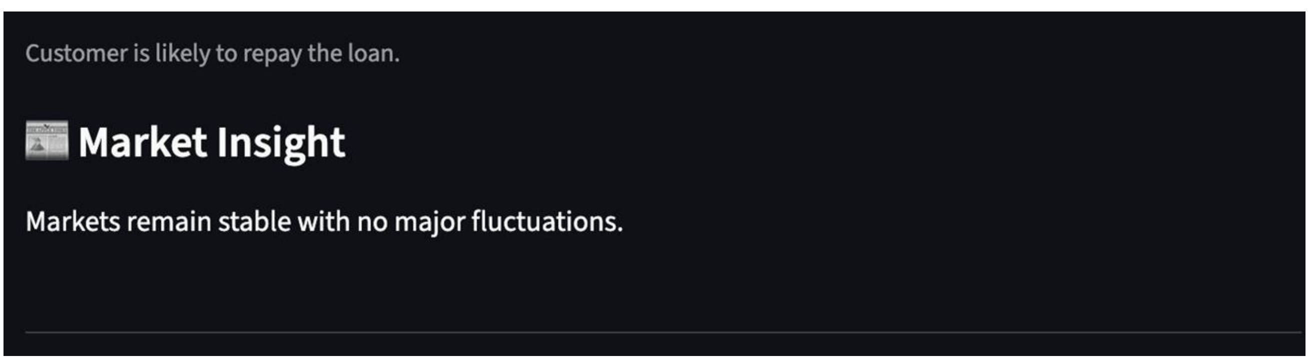
```
from transformers import pipeline
def sentiment_model():
    sentiment_pipeline = pipeline(
        "sentiment-analysis",
        model="ProsusAI/finbert"
    )

    def predict_sentiment(text):
        result = sentiment_pipeline(text)[0]
        label = result['label']

        if label == "positive":
            return 1
        elif label == "negative":
```

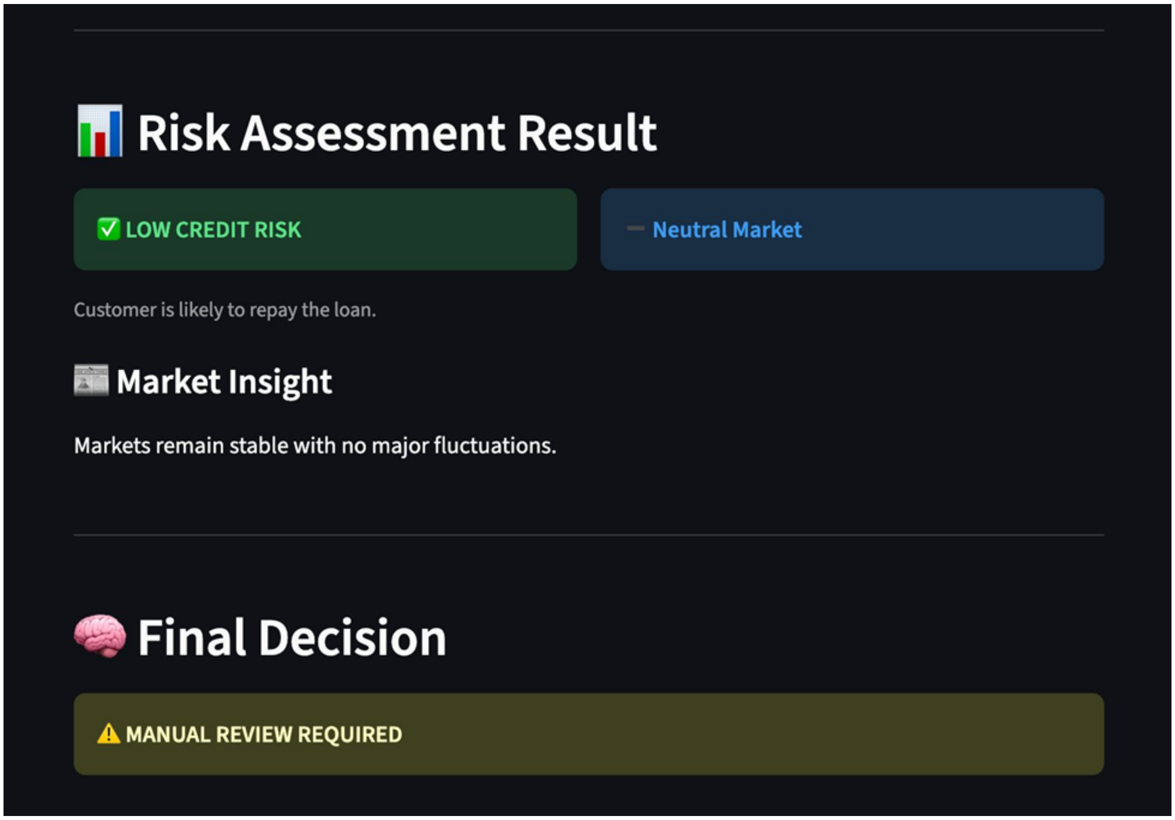
```
        return -1
    else:
        return 0

return predict_sentiment
```

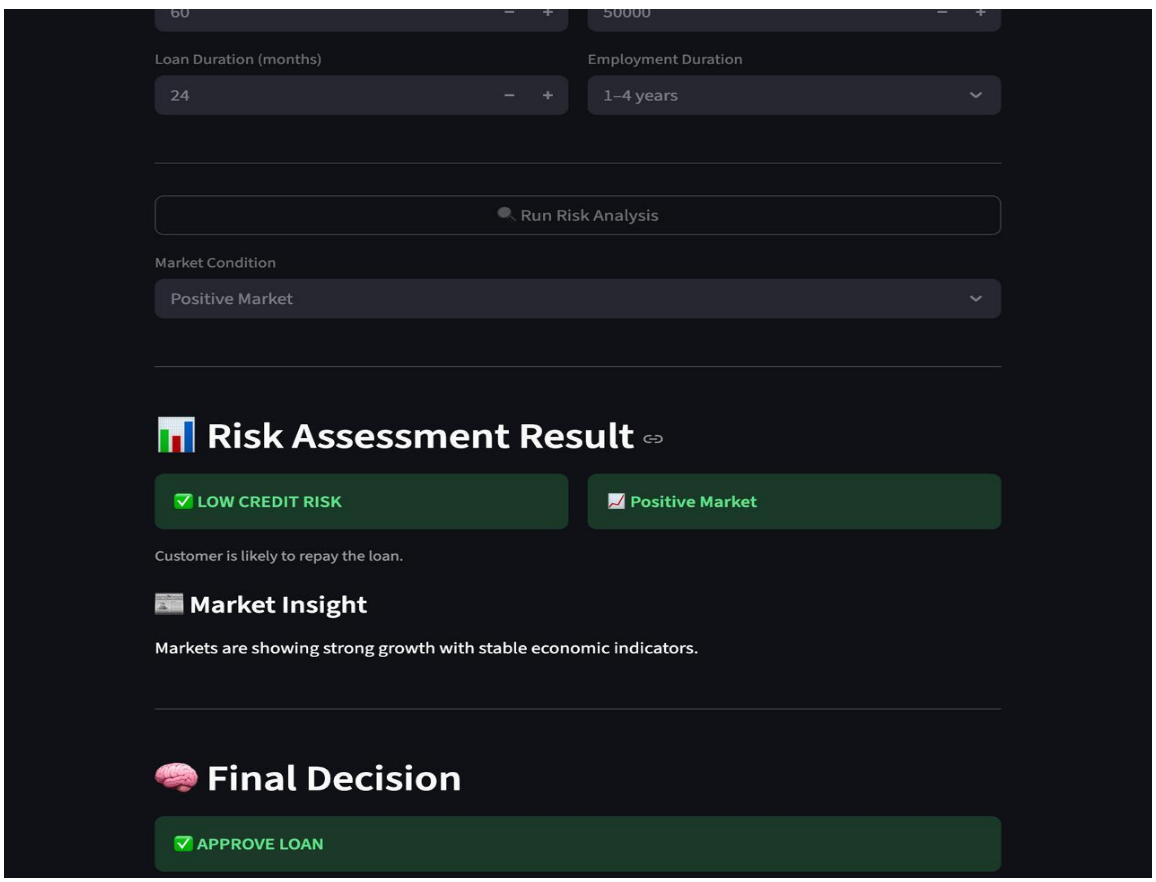
SNAPSHOTS


Financial Risk Intelligence System
AI-powered credit risk assessment with system-estimated market conditions
Models loaded successfully
Customer Information
Age
30
Credit Amount
5000
Loan Duration (months)
24
Employment Duration
Unemployed
Run Risk Analysis


Customer Information
Age
30
Credit Amount
5000
Loan Duration (months)
24
Employment Duration
Unemployed


Customer is likely to repay the loan.
Market Insight
Markets remain stable with no major fluctuations.


Risk Assessment Result
LOW CREDIT RISK
Neutral Market
Customer is likely to repay the loan.
Market Insight
Markets remain stable with no major fluctuations.
Final Decision
MANUAL REVIEW REQUIRED


Loan Duration (months)
24
Employment Duration
1–4 years
Run Risk Analysis
Market Condition
Positive Market
Risk Assessment Result
LOW CREDIT RISK
Positive Market
Customer is likely to repay the loan.
Market Insight
Markets are showing strong growth with stable economic indicators.
Final Decision
APPROVE LOAN

CONCLUSION

The Financial Risk Intelligence System developed in this project successfully demonstrates the practical implementation of classification techniques within the finance domain. In today's rapidly evolving financial ecosystem, institutions face increasing challenges in managing credit risk, detecting fraudulent activities, and making accurate lending decisions. Traditional rule-based systems are often insufficient to handle large-scale, complex, and dynamic financial data. This project addresses these challenges by leveraging machine learning-based classification models to build an intelligent, data-driven risk assessment system.

Throughout the project, financial data was carefully preprocessed, cleaned, and transformed to ensure accuracy and consistency. Feature engineering techniques were applied to extract meaningful indicators that significantly influence risk prediction. Multiple classification algorithms were implemented and evaluated to determine the most effective model for predicting financial risk categories. Appropriate performance metrics such as precision, recall, F1-score, and ROC-AUC were used instead of relying solely on accuracy, ensuring reliable evaluation especially in scenarios involving imbalanced datasets.

One of the key strengths of this project is the integration of the trained machine learning model with a Streamlit-based web application. The deployment of the model into an interactive web interface transformed a theoretical predictive model into a practical financial tool. Users can input relevant financial parameters and instantly receive a predicted risk classification, demonstrating how machine learning solutions can be effectively implemented in real-world financial environments. This enhances usability, accessibility, and operational efficiency.

The system also highlights important considerations in financial analytics, including model interpretability, scalability, and adaptability. Since financial decisions have significant regulatory and ethical implications, the selection of transparent and explainable models plays a crucial role. Additionally, the project emphasizes the importance of continuous monitoring and periodic model retraining to address concept drift caused by changing economic conditions, customer behavior, and market trends.

Overall, the Financial Risk Intelligence System bridges the gap between academic machine learning concepts and industry-level financial applications. It showcases how classification models can assist financial institutions in reducing potential losses, improving credit decision processes, enhancing fraud detection mechanisms, and strengthening overall risk management strategies. The project not only reinforces the importance of predictive analytics in finance but also demonstrates the potential of integrating machine learning with user-friendly web technologies to create intelligent and scalable financial solutions.

In conclusion, this project provides a strong foundation for intelligent risk prediction systems and illustrates how advanced data science techniques can transform traditional financial decision-making processes into automated, efficient, and reliable systems

FUTURE SCOPE AND ENANCEMENT

The Financial Risk Intelligence System developed in this project provides a strong foundation for intelligent financial risk prediction; however, there are several opportunities for further enhancement and expansion. As financial environments continue to evolve, integrating more advanced technologies and broader data sources can significantly improve the system's predictive accuracy, scalability, and real-world applicability.

In the future, the system can be enhanced by incorporating larger and more diverse datasets, including real-time transaction data, alternative financial data, behavioral indicators, and macroeconomic variables. The inclusion of external data sources such as credit bureau reports, market trends, and economic indicators could improve the robustness of risk predictions. Additionally, integrating real-time data streaming capabilities would allow the system to perform dynamic risk assessment, making it more suitable for live financial environments such as banking and fintech platforms.

Another important enhancement would be the implementation of advanced machine learning and deep learning models. While traditional classification algorithms provide reliable performance, more sophisticated techniques such as ensemble learning, gradient boosting optimization, and neural networks could further increase prediction accuracy, especially when handling high-dimensional or complex financial data. Hyperparameter optimization and automated machine learning (AutoML) approaches could also be introduced to streamline model selection and improve efficiency.

Model interpretability and explainability can be further strengthened by integrating explainable AI techniques. As financial decisions are subject to strict regulatory requirements, incorporating tools that provide clear reasoning behind risk predictions would enhance transparency and trust. This would make the system more suitable for deployment in regulated financial institutions.

Scalability is another key area for improvement. The system can be deployed on cloud platforms to handle large-scale financial data and multiple concurrent users. Containerization and microservices architecture could be adopted to improve performance, flexibility, and maintainability. Furthermore, integrating secure authentication mechanisms and role-based access control would enhance system security, especially if deployed in a real organizational environment.

Continuous monitoring and automated model retraining mechanisms could also be implemented to address concept drift caused by changing economic conditions and customer behavior. By building a feedback loop that tracks prediction performance over time, the system could adapt automatically to new financial patterns, ensuring long-term reliability.

In addition, the user interface developed using Streamlit can be enhanced with advanced visualization dashboards, risk trend analysis, downloadable reports, and decision-support analytics. Providing graphical insights and risk explanations would make the system more interactive and beneficial for financial analysts and decision-makers.

Overall, the future scope of the Financial Risk Intelligence System lies in expanding its data capabilities, improving predictive performance, enhancing interpretability, strengthening security, and enabling scalable deployment. With these improvements, the system has the potential to evolve into a comprehensive, enterprise-level financial risk management solution capable of supporting real-time intelligent decision-making in modern financial institutions.

REFERENCES

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

[2] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794).

[3] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273

[4] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874.

[5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[6] Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

[7] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An Introduction to Statistical Learning (2nd ed.). Springer.

[8] Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

USER GUIDE

Introduction

This User Guide provides detailed instructions for installing, running, and using the Financial Risk Intelligence System. The system is a Streamlit-based web application designed to predict financial risk levels using machine learning classification models. It allows users to input financial parameters and receive real-time risk predictions through an interactive interface.

System Requirements

To run the Financial Risk Intelligence System, the user must have a system with at least 8 GB RAM, a 64-bit operating system (Windows, macOS, or Linux), and Python version 3.8 or higher installed. An internet connection may be required for installing dependencies if they are not already available on the system.

Installation Process

Before running the application, required libraries must be installed. The user should first install Python and ensure that it is properly configured in the system environment variables. It is recommended to use Anaconda or a virtual environment to manage dependencies.

After setting up Python, the required packages can be installed using the following command in the terminal or command prompt:

pip install -r requirements.txt

If a requirements file is not provided, the user can manually install the necessary libraries such as Streamlit, NumPy, Pandas, Scikit-learn, Matplotlib, and Seaborn using pip.

Running the Application

Once the installation is complete, the user can navigate to the project directory in the terminal and execute the following command:

streamlit run app.py

After running this command, the Streamlit application will automatically open in the default web browser, typically at:

http://localhost:8501

The system is now ready for use.

Using the Financial Risk Intelligence System

Upon launching the application, the user is presented with an interactive dashboard. The interface contains input fields where financial parameters must be entered. These parameters may include attributes such as income level, loan amount, credit score, debt-to-income ratio, employment status, transaction frequency, or other risk-related features depending on the dataset used in the project.

The user enters the required financial details into the respective fields and submits the form by clicking the prediction button. The system processes the input data, applies the trained classification model, and generates a risk prediction.

The output is displayed on the screen, indicating the predicted risk category, such as Low Risk, Medium Risk, or High Risk. In some cases, the system may also display probability scores or additional visualizations to provide better insight into the prediction result.

Interpreting the Results

The predicted risk category helps users understand the likelihood of financial risk associated with the provided input. A low-risk prediction suggests a lower probability of default or financial instability, whereas a high-risk prediction indicates a greater potential risk. These predictions can support decision-making in loan approvals, credit assessments, or fraud detection scenarios.

It is important to note that the system provides predictive assistance and should be used as a decision-support tool rather than a sole determinant for critical financial decisions.

Error Handling

If invalid or incomplete data is entered, the system may display an error message requesting the user to provide valid inputs. Users should ensure that numerical fields contain appropriate numeric values and required fields are not left empty.

If the application fails to run, users should verify that all dependencies are correctly installed and that the correct Python environment is activated.

Maintenance and Updates

The system should be periodically updated with new financial data to maintain prediction accuracy. Model retraining may be required to adapt to changing financial trends and economic conditions. Additionally, updating library versions and monitoring application performance ensures long-term reliability.

Conclusion

The Financial Risk Intelligence System provides an intuitive and efficient platform for predicting financial risk using machine learning classification techniques. With its Streamlit-based interface, the system enables users to interact easily with complex predictive models and obtain real-time insights, making it a practical tool for financial risk assessment.

COURSE COMPLETION CERTIFICATE

Infosys
Navigate your next

COURSE COMPLETION CERTIFICATE

The certificate is awarded to

Suma .

for successfully completing the course

Software Engineering

on February 17, 2026

Infosys | Springboard

Congratulations! You make us proud!

Issued on: Tuesday, February 17, 2026
To verify, scan the QR code at https://verify.onwingspan.com

Satheesha B.N.
Satheesha B. Nanjappa
Senior Vice President and Head
Education, Training and Assessment
Infosys Limited

GEO TAGGED PHOTO DURING PRESENTATION