Credit Score Classification Using Machine Learning and Deep Learning Models

1. Overview of the Project

The core objective of this project was to develop and compare several ML and DL models with regard to performance for multi-class credit score classification: Poor, Standard, and Good. Accurate prediction of credit scores may enable financial institutions to judge the creditworthiness of an individual with more assurance and, hence, facilitate decision-making processes regarding loan approvals and risk management.
This classification task was done based on the dataset taken from Kaggle that had all the details concerning the financial and demographic attributes of the individuals: income, payment history, debt, credit utilization, and a variety of credit behavior indicators.

2. Description of Dataset

The dataset had 26 features between financial and demographic data for individuals. These were:
Demographic Information: ID, Customer_ID, Month, Name, Age, SSN
Financial Attributes: Annual_Income, Monthly_Inhand_Salary, Num_Bank_Accounts, Num_Credit_Card, Interest_Rate, Num_of_Loan, Outstanding_Debt, Credit_Utilization_Ratio, Total_EMI_per_month, Amount_invested_monthly, Monthly_Balance
Credit Behavioural Attributes: No_of_Deliquent_Payment, Credit_Mix, Delay_from_due_date, Changed_Credit_Card, No_Credit_Inquiries, Payment_of_Min_Amount, Payment_Behaviour
Target Variable: Credit_Score -> Poor, Standard or Good.

3. Data Preprocessing and Feature Engineering

Missing Value Treatment: imputed numerical columns with mean, for categorical columns, mode.
Aggregation: Aggregation by Customer_ID was done in order to aggregate the multiple records of each person by taking an average of the numerical features and mode in the case of categorical features.
Feature Encoding: The categorical features are encoded using LabelEncoder, which replaces categorical features into their numerical versions.
Feature Scaling: Numerical columns have been standardized using StandardScaler to scale all features to the same size.
Further cleaning the dataset, a cleaned one was created and further cleaned by removing columns like ID, Customer_ID, SSN, and Name, since it is not required to keep such information private or leak.

4. Model Development and Evaluation

The dataset was divided into training sets of 85% and testing sets of 15%. The following models were implemented:
Machine Learning Models:
Logistic Regression
Random Forest Classifier
Support Vector Machine (SVM)

XGBoost Classifier

Deep Learning Models:
Multi-Layer Perceptron (MLP)
Convolutional Neural Network (CNN)
Long Short-Term Memory Network (LSTM)

5. Model Performance

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.65	0.66	0.65	0.65	0.82
Random Forest	**0.75**	**0.76**	**0.75**	**0.75**	**0.88**
SVM	0.47	0.22	0.47	0.30	0.79
XGBoost	0.73	0.73	0.73	0.73	0.88
MLP	0.61	0.68	0.61	0.61	0.81
CNN	0.43	0.59	0.43	0.43	0.73
LSTM	0.71	0.73	0.71	0.72	0.85

The Random Forest Classifier had the best performance in all key metrics and proved to be the best model in this task.

6. Key Insights

Feature Importance: Using the Random Forest feature importance, there were a number of important features that contribute toward the prediction of credit score. Some important ones include Monthly_Inhand_Salary, Credit_Utilization_Ratio, Outstanding_Debt, and Num_of_Delayed_Payment.
Comparing the Models: Among ML models, Random Forest and XGBoost outperformed others such as Logistic Regression and SVM. This hinted at the superiority of ensemble methods for this particular classification problem.
DL Model Challenges: The convolutional neural network performed much worse, even compared to the LSTM. Poor performance by the models probably occurred because the nature of the data is tabular data that doesn't show either sequential or spatial relationships. However, MultiLayerPerceptron, which is actually a fully connected network, presented pretty reasonable performance.

7. Conclusion

In this project, we deal with credit score classification, which also ranges from ensemble learning methods. Taking into consideration the fact that financial applications have a requirement for interpretability, the Random Forest model has been not only high regarding accuracy but also interpretable. Future work could be hyperparameter tuning, feature selection, and other ensemble techniques testing to further tune the model's performance.

Visualizations
In addition to model development, visualizations were created in Tableau to explore and gain insights from the data. [Tableau Visualization]