**Credit Risk Modeling – AMEX Case Study**

This project presents an end-to-end machine learning solution to predict **credit default risk**, using a real-world industrial-scale dataset provided by **American Express** as part of their Kaggle competition: [AMEX Default Prediction](https://www.kaggle.com/competitions/amex-default-prediction).

**🧭 Introduction**

In the evolving landscape of consumer lending, accurately identifying creditworthy individuals is crucial for minimizing risk and maximizing customer satisfaction. This project leverages machine learning techniques to enhance credit default prediction, enabling financial institutions like **American Express** to offer credit products to the **right customers at the right time**.

By modeling customer behavior using **time-series transactional data** and anonymized profile information, we can build a more **robust, explainable, and high-performing model** that improves upon traditional risk models. This ensures better approval decisions, reduces potential losses, and ultimately **enhances the customer experience**.

🔗 **Data Source:** [Kaggle Competition – AMEX Default Prediction](https://www.kaggle.com/competitions/amex-default-prediction)

**🧠 Key Contributions**

* ✅ Handled a **large-scale dataset (5.5M+ rows)** with efficient chunking
* ✅ Performed **missing value treatment**, **categorical encoding**, and **temporal aggregation** over 12-month periods
* ✅ Engineered features like mean, min, max, and last for numeric time-series variables
* ✅ Applied **one-hot encoding** to categorical variables, generating 45+ new features
* ✅ Built and compared two models: **XGBoost** and a **Neural Network**, using extensive grid search
* ✅ Selected the final model based on **AUC performance** and **variance stability** across training and test splits
* ✅ Used **SHAP values** for model explainability and to understand feature contributions to risk prediction

**📁 Project Structure**

📦 Credit\_Risk\_Modeling

┣ 📊 AMEX\_Credit\_Risk\_Analysis.ipynb

┣ 📈 Credit Risk Model.pptx

┣ 🧠 credit\_risk\_modeling.py

┣ 📄 final\_data\_for\_project.csv

┗ 📘 README.md

**🎯 Objective**

To build an accurate and explainable credit risk model using machine learning that can challenge the existing production model used by American Express. The project implements two modeling strategies and selects the best based on AUC performance:

✅ **XGBoost**  
🧪 **Neural Network**

**📂 Data Overview**

* **Total Customers:** 91,783
* **Time Span:** 13 months of historical data
* **Categories of Features:**
  + D\_ - Delinquency
  + B\_ - Balance
  + S\_ - Spend
  + P\_ - Payment
  + R\_ - Risk

**⚙️ Feature Engineering**

* 📅 Aggregated temporal features: mean, min, max, last over 12 months
* 🏷️ One-hot encoded 11 categorical columns → 45 dummy variables
* 🧼 Missing values handled based on type:
  + Binary → 0
  + Categorical → Mode
  + Numeric → Mean

**🧪 Model Training**

**1️⃣ XGBoost**

* Grid search on:
  + Trees: 50, 100, 300
  + Learning rates: 0.01, 0.1
  + Row subsample: 50%, 80%
  + Feature subsample: 50%, 100%
  + Class weight: 1, 5, 10
* ✅ **72 models trained**
* 📈 **Best AUC performance on Test 2**
* 💡 SHAP analysis used for model interpretability

**2️⃣ Neural Network**

* Outliers capped to 99th percentile
* Normalized using StandardScaler
* Grid search on:
  + Hidden layers: 2, 4
  + Neurons: 4, 6
  + Activation: ReLU, Tanh
  + Dropout: 50%, 100%
  + Batch size: 100, 10000
* Trained 32 models
* ❌ Did **not outperform** XGBoost

**📊 SHAP Value Insights (XGBoost)**

| **Feature** | **Impact** |
| --- | --- |
| P\_2\_last | Negative |
| B\_1\_last | Negative |
| D\_39\_max | Positive |
| S\_3\_mean | Positive |

SHAP values helped explain **how each feature contributes** to a prediction.

**🏆 Final Model Selection**

* ✅ **XGBoost outperformed NN** across all sets (Train, Test1, Test2)
* Highest AUC with lowest variance
* Chosen as the final model for production-ready implementation

**🚀 Future Enhancements**

* Streamlit dashboard for real-time scoring
* AutoML model comparison
* Model monitoring for drift detection

**📈 Results Summary**

| **Model** | **AUC (Test2)** | **Remarks** |
| --- | --- | --- |
| XGBoost | ✅ High | Best performer overall |
| Neural Net | Moderate | Underperformed |

**🛠 Tech Stack**

* Python, Pandas, NumPy
* Matplotlib, Seaborn
* LightGBM, XGBoost, Scikit-Learn
* SHAP, GridSearchCV
* PowerPoint for stakeholder communication

**📣 Let's Connect**

For questions, improvements, or collaboration, feel free to reach out or fork the repo!