10 Academy: Artificial Intelligence Mastery

Credit Risk Probability Model for Alternative Data

An End-to-End Implementation for Building, Deploying, and Automating a Credit Risk Model

Date: 25 Jun - 01 Jul 2025

Overview

Business Need

You are an Analytics Engineer at **Bati Bank**, a leading financial service provider with over 10 years of experience. Bati Bank is partnering with an upcoming successful eCommerce company to enable a **buy-now-pay-later** service - to provide customers with the ability to buy products by credit if they qualify for the service. You are assigned a project to create a **Credit Scoring Model** using the data provided by the eCommerce platform.

Credit scoring is the term used to describe the process of assigning a quantitative measure to a potential borrower as an estimate of how likely the default will happen in the future. Traditionally, creditors build credit scoring models using statistical techniques to analyze various information of previous borrowers in relation to their loan performance. Afterward, the model can be used to evaluate a potential borrower who applies for a loan by providing the similar information which has been used to build the model. The result is either a score which represents the creditworthiness of an applicant or a prediction of whether an applicant will default in the future.

The definition of default in the context of credit scoring may vary between each financial institution as long as it complies with the Basel II Capital Accord - you must read this reference to understand the factors the bank needs to take into account to start a new loan procedure. A quick summary of the Basel II Capital Accord can be found in this reference.

The key innovation lies in transforming behavioral data into a predictive risk signal. By analyzing customer Recency, Frequency, and Monetary (RFM) patterns, we engineer a proxy for credit risk. This allows us to train a model that outputs a risk probability score, a vital metric that can be used to inform loan approvals and terms.

Your job is to build a product that does the following

1. Defines a proxy variable that can be used to categorize users as high risk (bad) or low risk (good)
2. Select observable features that are good predictors (have high correlation) of the default variable defined in 1)
3. Develop a model that assigns risk probability for a new customer
4. Develop a model that assigns credit score from risk probability estimates
5. Develop a model that predicts the optimal amount and duration of the loan

Data and Features

The data for this challenge can be found here. Or you can find it also here: Xente Challenge | Kaggle

Data fields

- **TransactionId:** Unique transaction identifier on platform
- **BatchId:** Unique number assigned to a batch of transactions for processing
- **AccountId:** Unique number identifying the customer on platform
- **SubscriptionId:** Unique number identifying the customer subscription
- **CustomerId:** Unique identifier attached to Account
- **CurrencyCode:** Country currency
- **CountryCode:** Numerical geographical code of country
- **ProviderId:** Source provider of Item bought.
- **ProductId:** Item name being bought.
- **ProductCategory:** ProductIds are organized into these broader product categories.
- **ChannelId:** Identifies if customer used web,Android, IOS, pay later or checkout.
- **Amount:** Value of the transaction. Positive for debits from customer account and negative for credit into cus...
- **Value:** Absolute value of the amount
- **TransactionStartTime:** Transaction start time
- **PricingStrategy:** Category of Xente's pricing structure for merchants
- **FraudResult:** Fraud status of transaction 1 -yes or 0-No

Learning Outcomes

Skills:

- Advanced use of scikit-learn
- Feature Engineering
- ML Model building and fine-tuning
- CI/CD deployment of ML models
- Python logging
- Unit testing
- Model management
- MLOps with CML, and MLFlow

Knowledge:

- Reasoning with business context
- Data exploration
- Predictive analysis
- Machine learning
- Hyperparameter tuning
- Model comparison & selection

Communication:

- Reporting on statistically complex issues

Team

Tutors:

- Mahlet
- Rediet
- Kerod
- Rehmet

Key Dates

- Discussion on the case - 09:00 UTC on Wednesday 25 June 2025. Use #all-week5 to pre-ask questions.
- Interim Solution - 20:00 UTC on Sunday 29 June 2025.
- Final Submission - 20:00 UTC on Tuesday 01 July 2025

Instructions

Project Structure: Mandate a standardized project structure from the beginning. This is a core engineering discipline.

```
credit-risk-model/
├── .github/workflows/ci.yml   # For CI/CD
├── data/                       # add this folder to .gitignore
│   ├── raw/                   # Raw data goes here
│   └── processed/             # Processed data for training
├── notebooks/
│   └── 1.0-eda.ipynb          # Exploratory, one-off analysis
├── src/
│   ├── __init__.py
│   ├── data_processing.py     # Script for feature engineering
│   ├── train.py               # Script for model training
│   ├── predict.py             # Script for inference
│   └── api/
│       ├── main.py            # FastAPI application
│       └── pydantic_models.py # Pydantic models for API
├── tests/
│   └── test_data_processing.py # Unit tests
├── Dockerfile
├── docker-compose.yml
├── requirements.txt
├── .gitignore
└── README.md
```

Task 1 - Understanding Credit Risk

Focus on understanding the concept of Credit Risk. Read the provided references on Credit Risk and the Basel II Capital Accord.

Key references

- https://www3.stat.sinica.edu.tw/statistica/oldpdf/A28n535.pdf
- https://www.hkma.gov.hk/media/eng/doc/key-functions/financial-infrastructure/alternative_credit_scoring.pdf

    - https://thedocs.worldbank.org/en/doc/935891585869698451-0130022020/original/CREDITSCORINGAPPROACHESGUIDELINESFINALWEB.pdf
    - https://towardsdatascience.com/how-to-develop-a-credit-risk-model-and-scorecard-91335fc01f03
    - https://corporatefinanceinstitute.com/resources/commercial-lending/credit-risk/
    - https://www.risk-officer.com/Credit_Risk.htm

In your README.md file, create a section titled "Credit Scoring Business Understanding."
In this section, write a concise summary that answers the following three questions:

- How does the Basel II Accord's emphasis on risk measurement influence our need for an interpretable and well-documented model?
- Since we lack a direct "default" label, why is creating a proxy variable necessary, and what are the potential business risks of making predictions based on this proxy?
- What are the key trade-offs between using a simple, interpretable model (like Logistic Regression with WoE) versus a complex, high-performance model (like Gradient Boosting) in a regulated financial context?

Deliverables:

- The updated README.md file in your GitHub repository containing the completed "Credit Scoring Business Understanding" section.

Task 2 - Exploratory Data Analysis (EDA)

To explore the dataset to uncover patterns, identify data quality issues, and form hypotheses that will guide your feature engineering. Use the Jupyter Notebook for all your exploratory work. This notebook is for exploration only; it is not for production code. **Based on your findings, summarize your top 3-5 most important insights.**

1. **Overview of the Data:**
    - Understand the structure of the dataset, including the number of rows, columns, and data types.
2. **Summary Statistics**
    - Understand the central tendency, dispersion, and shape of the dataset's distribution.
3. **Distribution of Numerical Features**
    - Visualize the distribution of numerical features to identify patterns, skewness, and potential outliers.
4. **Distribution of Categorical Features**

  - Analyzing the distribution of categorical features provides insights into the frequency and variability of categories.
5. **Correlation Analysis**
   - Understanding the relationship between numerical features.
6. **Identifying Missing Values**
   - Identify missing values to determine missing data and decide on appropriate imputation strategies.
7. **Outlier Detection**
   - Use box plots to identify outliers.

Task 3 - Feature Engineering

Build a robust, automated, and reproducible data processing script that transforms raw data into a model-ready format.

1. All feature engineering logic must be implemented in the src/ .py scripts.
2. Inside the script, use sklearn.pipeline.Pipeline to chain together all transformation steps
3. **Create Aggregate Features**
   Example:
   - **Total Transaction Amount:** Sum of all transaction amounts for each customer.
   - **Average Transaction Amount**: Average transaction amount per customer.
   - **Transaction Count:** Number of transactions per customer.
   - **Standard Deviation of Transaction Amounts:** Variability of transaction amounts per customer.
4. **Extract Features**
   Example:
   - **Transaction Hour:** The hour of the day when the transaction occurred.
   - **Transaction Day:** The day of the month when the transaction occurred.
   - **Transaction Month:** The month when the transaction occurred.
   - **Transaction Year:** The year when the transaction occurred.
5. **Encode Categorical Variables**
   Convert categorical variables into numerical format by using:
   - **One-Hot Encoding:** Converts categorical values into binary vectors.
   - **Label Encoding:** Assigns a unique integer to each category.
6. **Handle Missing Values**
   Use imputation or Removal to handle missing values
   - **Imputation**: Filling missing values with mean, median, mode, or using more methods like KNN imputation.
   - **Removal**: Removing rows or columns with missing values if they are few.
7. **Normalize/Standardize Numerical Features**

Normalization and standardization are scaling techniques used to bring all numerical features onto a similar scale.

- **Normalization**: Scales the data to a range of [0, 1].
- **Standardization**: Scales the data to have a mean of 0 and a standard deviation of 1.

Feature Engineering using:

- https://pypi.org/project/xverse/
- https://pypi.org/project/woe/
- WEIGHT OF EVIDENCE (WOE) AND INFORMATION VALUE (IV) EXPLAINED

Task 4 - Proxy Target Variable Engineering

You do not have a pre-existing "credit risk" column in your data. The goal of this task is to create one. You will do this by programmatically identifying a group of "disengaged" customers and labeling them as high-risk proxies. High-risk groups are those with high likelihood of default - those who do not pay the loan principal and interest in the specified time frame.

1. **Calculate RFM Metrics:** For each CustomerId, calculate their Recency, Frequency, and Monetary (RFM) values from the transaction history. You will need to define a snapshot date to calculate Recency consistently.
2. **Cluster Customers:** Use the **K-Means clustering** algorithm to segment customers into **3 distinct groups** based on their RFM profiles.
    - Remember to pre-process (e.g., scale) the RFM features appropriately before clustering to ensure meaningful results.
    - Set a random_state during clustering to ensure your work is reproducible.
3. **Define and Assign the "High-Risk" Label:** Analyze the resulting clusters to determine which one represents the least engaged and therefore highest-risk customer segment (typically characterized by low frequency and low monetary value).
    - Create a new binary target column named is_high_risk.
    - Assign a value of 1 to customers in this high-risk cluster and 0 to all others.
4. **Integrate the Target Variable:** Ensure this new is_high_risk column is merged back into your main processed dataset, so it is available for model training.

Task 5 - Model Training and Tracking

To develop a structured model training process that includes experiment tracking, model versioning, and unit testing.

1. Add mlflow and pytest to your requirements.txt.
2. Model Selection and Training
    a. **Split the Data**
    Splitting the data into training and testing sets helps evaluate the model's performance on unseen data.
    b. **Choose Models**
    Choose at least two models from the following:
        - Logistic Regression
        - Decision Trees
            - Random Forest
        - Gradient Boosting Machines (GBM)

    c. **Train the Models**
    Train the models on the training data
    d. **Hyperparameter Tunning**
    Improve model performance using hyperparameter tuning, use techniques like:
        - Grid Search
        - Random Search
3. Model Evaluation
    Assess model performance using the following metrics
    a. **Accuracy**: The ratio of correctly predicted observations to the total observations.
    b. **Precision:** The ratio of correctly predicted positive observations to the total predicted positives.
    c. **Recall (Sensitivity):** The ratio of correctly predicted positive observations to all observations in the actual class.
    d. **F1 Score:** The weighted average of Precision and Recall.
    e. **ROC-AUC:** Area Under the Receiver Operating Characteristic Curve, which measures the ability of the model to distinguish between classes.
4. After experimenting, identify your best model and register it in the MLflow Model Registry.
5. Write Unit Tests:
    In the tests/test_data_processing.py file, write at least two unit tests for a helper function within your scripts.

Task 6 - Model Deployment and Continuous Integration

To package the trained model into a containerized API and set up a CI/CD pipeline to automate testing and ensure code quality.

- Add fastapi, uvicorn, and a linter (flake8 or black) to your requirements.txt.
- **Create the API:** In src/api/main.py, build a REST API using **FastAPI**.
    - The API should load your best model from the MLflow registry.
    - Create a /predict endpoint that accepts new customer data (matching the model's features) and returns the risk probability.
    - Use Pydantic models in src/api/pydantic_models.py for request and response data validation.
- **Containerize the Service:**
    - Write a Dockerfile that sets up the environment and runs the FastAPI application using uvicorn.
    - Write a docker-compose.yml file to easily build and run your service.
- **Configure CI/CD:**
    - In the .github/workflows/ci.yml file, create a GitHub Actions workflow that triggers on every push to your main branch.
    - The workflow must have at least two steps:
        - A step to run a code linter (like flake8) to check for code style issues.
        - A step to run pytest to execute your unit tests.
    - The build should fail if either the linter or the tests fail.

Tutorials Schedule

Overview

In the following, the colour **purple** indicates morning sessions, and **blue** indicates afternoon sessions.

Wednesday:

- Introduction to the challenge(Mahlet)
- Introduction to Credit Risk Analysis and Modeling (Kerod)

Thursday:

- Feature Engineering, WEIGHT OF EVIDENCE(WoE), and INFORMATION VALUE (IV) (Rediet)
- Model Training, Hyperparameter Tuning, and Evaluation (Rehmet)

Friday:

- Model Serving and Deployment with docker (Rediet)
- Q&A

Deliverables

Interim Submission

- To verify that you have correctly set up your project foundation, understood the business context, and completed the initial data exploration and processing.
    - A review report of your reading and understanding of Task 1 and any progress you made in other tasks.
    - Link to your GitHub.

Feedback

You may not receive detailed comments on your interim submission but will receive a grade.

Final Submission

- A blog post entry (which you can submit for example to Medium publishing) or a pdf report.
- Link to your Github code, and make sure to include screenshots demonstrating anything else you have done.

Feedback

You will receive comments/feedback in addition to a grade.

References

1. Loss functions
2. Sklearn pipelines
3. Merging dataframes
4. RandomForests

Credit Risk

1. https://www.investopedia.com/terms/c/creditrisk.asp
2. https://investopedia.com/terms/c/creditspread.asp
3. https://cleartax.in/glossary/credit-risk/
4. https://corporatefinanceinstitute.com/resources/commercial-lending/credit-risk/
5. https://www.risk-officer.com/Credit_Risk.htm

   Publications:
   a. Credit Risk Determinants in Selected Ethiopian Commercial Banks: A Panel Data Analysis
   b. Factors Affecting Credit Risk Exposure of Commercial Banks in Ethiopia: An Empirical Analysis
   c. Credit Risk Analysis of Ethiopian Banks: A Fixed Effect Panel Data Model.
6. https://drive.google.com/drive/folders/1pAXmJ_Sl46D4Ex-nV0pDGvpxa7HD5erW?usp=drive_link
7. https://shichen.name/scorecard/

MLOps

1. Auto-sklearn — AutoSklearn 0.12.7 documentation (automl.github.io)
2. https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/
3. Hyperparameter tuning. Grid search and random search
4. https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

Kaggle kernels

1. https://www.kaggle.com/datasets/atwine/xente-challenge

Feature Engineering in Credit Scoring

1. https://pypi.org/project/xverse/
2. https://pypi.org/project/woe/
3. https://github.com/JGFuentesC/woe_credit_scoring
4. https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.htm
l
5. https://shichen.name/scorecard/

Related Optional References

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8860138/
- Credit Risk Determinants in Selected Ethiopian Commercial Banks: A Panel Data Analysis
- Factors Affecting Credit Risk Exposure of Commercial Banks in Ethiopia: An Empirical Analysis
- Credit Risk Analysis of Ethiopian Banks: A Fixed Effect Panel Data Model.