**Feature Document**

**Problem Statement** Detecting and preventing credit card fraud is critical for maintaining customer trust and reducing financial losses. Accurate detection of fraudulent transactions can significantly reduce unauthorized transactions and enhance security measures.

**Business Value**

* Improved detection of fraudulent activities
* Reduced financial losses due to fraud
* Enhanced customer trust and satisfaction
* Reduction in manual review of transactions
* Streamlined fraud management processes

**Project Stakeholders**

1. **Product Owner**
2. **Data Engineer**
3. **Data Science Team**
4. **Mlops**
5. **DevOps**
6. **Compliance**
7. **Scrum Master**

**Created by:**

1. **XXX**  - Version V 0.1:
2. **Edited by:**

**Model Details:**

1. Unique ID / RunID
2. Created by
3. Data version
4. Metrics
5. Parameters

**Used Features:**

| **Inference Feature** | **Used for Inference** | **Used for Training** | **Part of Batch Inference Request/File** | **Raw Source** | **Load From (How/From where we plan to load the feature)** | **Finally Agreed Source (Who is providing data)** |
| --- | --- | --- | --- | --- | --- | --- |
| Transaction\_ID | Yes | Yes | Yes | Transaction Database | Batch Process from Database | Transaction Database / Data Engineering Team |
| User\_Account\_ID | Yes | Yes | Yes | User Account Database | Batch Process from Database | User Account Database / Data Engineering Team |
| Transaction\_Amount | Yes | Yes | Yes | Transaction Database | Batch Process from Database | Transaction Database / Data Engineering Team |
| Transaction\_DateTime | Yes | Yes | Yes | Transaction Database | Batch Process from Database | Transaction Database / Data Engineering Team |
| Merchant\_Category | Yes | Yes | Yes | Merchant Database | Batch Process from Database | Merchant Database / Data Engineering Team |

**Testing:**

a. **Unit Testing:** Data processing and transformation validations.

b. **Integration Testing:** End-to-end model performance testing.

c. **API Testing:** Input and output validation through simulated API calls.

d. **Load Testing:** Simulated data volume to evaluate system performance under stress.

**Input Payload:**

json

Copy code

[

{

"transaction\_id": "123456789",

"user\_account\_id": "987654321",

"transaction\_amount": 250.00,

"transaction\_datetime": "2023-06-10T14:00:00Z",

"merchant\_category": "Electronics"

}

]

**Output Values:**

json

Copy code

{

"Fraud\_Status": [

{

"Transaction\_ID": "123456789",

"Is\_Fraudulent": true

}

]

}

**Error Section:**

| **Code** | **Meaning** | **Description** |
| --- | --- | --- |
| 200 | OK | The requested action was successful. |
| 201 | Created | A new resource was created. |
| 202 | Accepted | The request was received, but no modification has been made yet. |
| 204 | No Content | The request was successful, but the response has no content. |
| 400 | Bad Request | The request was malformed. |
| 401 | Unauthorized | The client is not authorized to perform the requested action. |
| 404 | Not Found | The requested resource was not found. |
| 415 | Unsupported Media Type | The request data format is not supported by the server. |
| 422 | Unprocessable Entity | The request data was properly formatted but contained invalid data. |
| 500 | Internal Server Error | The server encountered an error while processing the request. |