JJMALK Data Insights, Inc.

Model for Predicting Credit Card Fraud for BOMI

Milestone Status Report

Project Name: Model for Predicting Credit Card Fraud **Reporting Period:** December 8, 2023

Project Manager: Maricris Resma

Project Owner: JJMALK Data Insights, Inc.

Executive Summary

Overall Status: **ON TRACK**

	Status	Reason for Deviation
Budget	ON TRACK	n/a
Schedule	ON TRACK	Buffer was used but still ON TRACK for final milestone
Scope	ON TRACK	n/a

Comments:

- The first Project Phase: Planning has been done
- The project **Milestone - Model 2 Dev and Val Done** has been completed on December 8, 2023

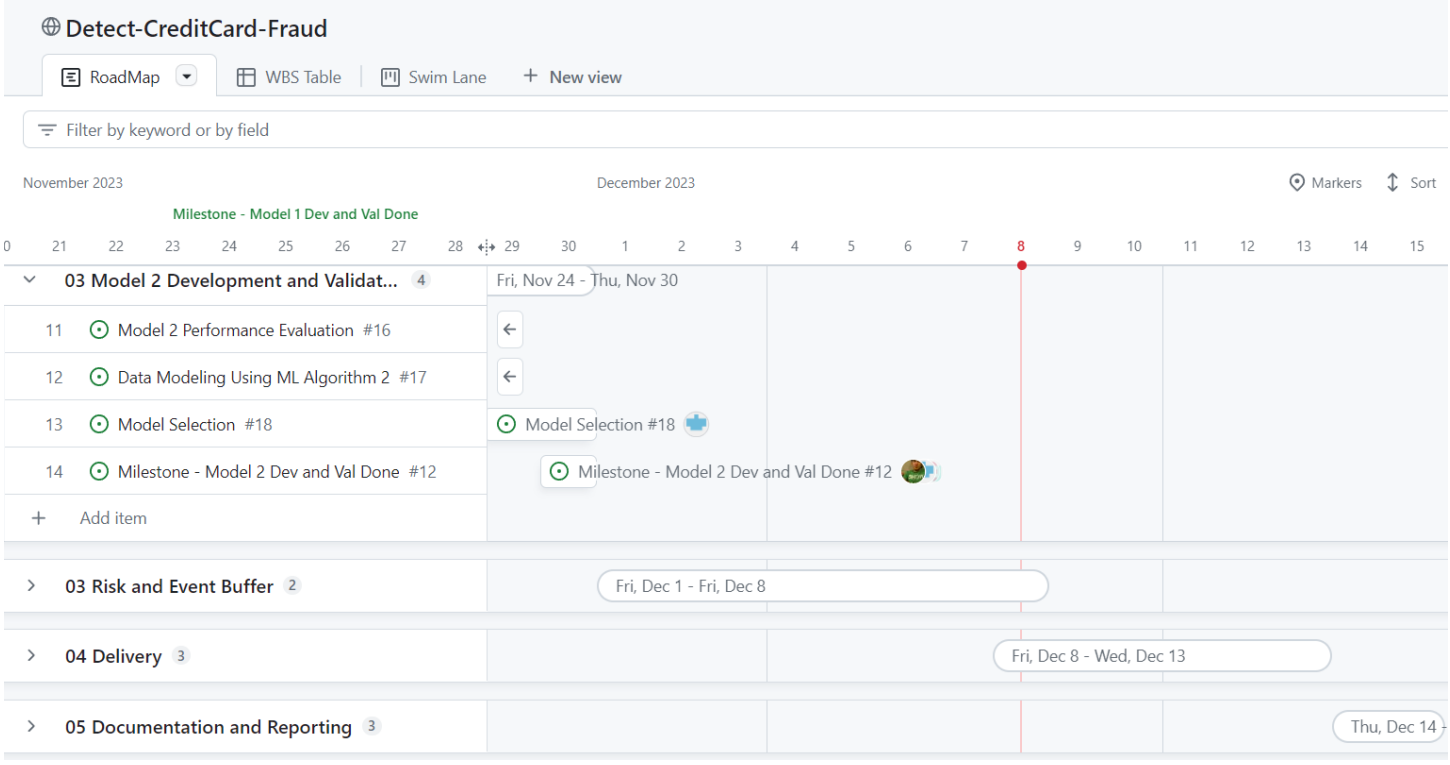
Project Definition

Project Definition

Project Objectives	▪ Develop a predictive modelling application that effectively predicts credit card fraudulent actions
Scope	▪ Retrieve and manage credit card transaction Data from Google Cloud Platform ▪ Create a machine learning classifier model capable of detecting whether credit card transaction is fraud or not(minimum 80% accuracy, recall and f1 score) using Decision Tree as the initial Model then Random Forest in case initial model does not reach desired accuracy. Use sklearn and imblearn libraries. ▪ Choose Model with best balance between reduced false positives to guarantee a seamless and trustworthy user experience but more importantly high precision to protect from financial loss ▪ Deliver the pickle file that can be integrated into Bank of Mississauga's organizational systems and databases
Assumptions	▪ Customer BOMI (Bank of Mississauga) will be responsible for integrating the delivered pickle file in their system

Scheduled Milestones / Deliverables

Gantt Chart of Project Timeline



Project Timeline and Deliverables

Project Tasks	Start	End	Deliverables	Status	Actual
Planning					
Proposal Creation and Submission	06-Oct-2023	12-Oct-2023		**DONE**	12-Oct-2023
Data Gathering	13-Oct-2023	20-Oct-2023		**DONE**	20-Oct-2023
Model Research and Design	13-Oct-2023	20-Oct-2023		**DONE**	20-Oct-2023
◆ ***Milestone- Planning Done***		***20-Oct-2023***	***Proposal Doc***	**DONE**	***20-Oct-2023***
Model 1 Development and Validation					
GIT Creation	23-Oct-2023	23-Oct-2023		**DONE**	23-Oct-2023
Data Preprocessing	24-Oct-2023	03-Nov-2023		**DONE**	03-Nov-2023
Exploratory Data Analysis	06-Nov-2023	10-Nov-2023		**DONE**	10-Nov-2023
Data Modeling Using ML Algorithm 1	13-Nov-2023	22-Nov-2023		**DONE**	22-Nov-2023
Model 1 Performance Evaluation	13-Nov-2023	22-Nov-2023		**DONE**	22-Nov-2023
◆ ***Milestone - Model 1 Dev and Val Done***		***23-Nov-2023***		**DONE**	***23-Nov-2023***
Model 2 Development and Validation (if needed)					
Data Modeling Using ML Algorithm 2	24-Nov-2023	28-Nov-2023		**DONE**	28-Nov-2023
Model 2 Performance Evaluation	24-Nov-2023	28-Nov-2023		**DONE**	30-Nov-2023
Model Selection	29-Nov-2023	30-Nov-2023		**DONE**	8-Dec-2023
◆ ***Milestone - Model 2 Dev and Val Done***		***30-Nov-2023***		**DONE**	***8-Dec-2023***
Delivery					
Code Freeze	01-Dec-2023	01-Dec-2023		**NOT STARTED**	
Pickle File Generation and Delivery	04-Dec-2023	04-Dec-2023		**NOT STARTED**	
◆ ***Milestone - Customer Acceptance***		***04-Dec-2023***	***zip file of .ipynb and pickle file***	**NOT STARTED**	

Project Tasks	Start	End	Deliverables	Status	Actual
Documentation and Reporting					
Documentation	05-Dec-2023	07-Dec-2023		**NOT STARTED**	
Create Presentation	05-Dec-2023	07-Dec-2023		**NOT STARTED**	
◆ ***Milestone - Presentation***		***08-Dec-2023***	***.doc report and .ppt file***	**NOT STARTED**	
Risk and Event Buffer					
Risk Buffer	11-Dec-2023	13-Dec-2023		**DONE**	1-Dec-2023
Event Buffer	14-Dec-2023	15-Dec-2023		**DONE**	7-Dec-2023

Note: ◆ *signify Key Milestones*

Accomplishments during this Reporting Period:

1. Libraries and built-in functions used:
   - Data visualization:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

   - Data Preprocessing

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```

   - Feature Engineering

```
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import SMOTE, SMOTENC
```

   - Data Modelling

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
```

   - Performance Metrics

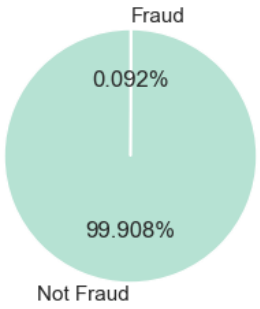
```
from sklearn.metrics import accuracy_score, confusion_matrix,
```

   - Cross Validation

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from imblearn.pipeline import Pipeline
```

2. Improve Data Preprocessing
   - Data profiling

Data Quality	**Description**
Format	.CSV
Shape	15 features 24,386,900 records Y (Dependent variable) – **Is Fraud?** X (Independent variables): # Column Dtype --- ------ ----- 0 User int64 1 Card int64 2 Year int64 3 Month int64

	4 Day int64 5 Time object 6 Amount object 7 Use Chip object 8 Merchant Name int64 9 Merchant City object 10 Merchant State object 11 Zip float64 12 MCC int64 13 Errors? object
Observations	- Imbalanced Y (29757 – Yes and 24357143 – No) - 3 features have missing values ([‘Errors?’, ‘Zip’, ‘Merchant State’])
Sampling	Recent information will be used to capture latest patterns in fraudulent transactio ns. - Sample latest 3 years(2019,2018,2017). - Retain successful transactions only (no error). - reduced to 5,087,066 records with following class distribution **Fraud No 5,082,411** **Fraud Yes 4,655** 

- Data preprocessing

Data	Description of cleaning done
Convert features data type	- Assign Merchant Code Groupings - Found significance in grouping as online, foreign and local
Null values	- Investigated NaN values for Zip were related to non USA local states - Found significance in grouping as online, foreign and local
Time	- Convert to float - extracted Day of Week
Data Encoding	- Convert non ordinal categorical features using One Hot Encoding

3. Data Split Train and Test
    - Applied stratify to maintain the same class distribution for both train and test

4. Feature Engineering
    - Data Transformation - Applying PCA resulted to worse performance on the test precision therefore it is not recommended to use this
    - Sampling Imbalanced Data
        - on train data, applied undersampling(70/30) which performed better then oversampling using SMOTE KNN Approach resulting to reduced records with following class distribution:

```
After Undersampling
No. of Observations: 11,015
      Fraud No    0.704222
      Fraud Yes    0.295778
```

        - This resulted to very high accuracy and precision but 0nly 10% recall
        - Note disadvantages of undersampling are loss of data meanwhile oversampling leads to synthetic records

    - Class Weight
        - Use original sample without oversampling or undersampling but applying calss wait for the model as {NonFraud:.01 ,Fraud:.99 }
        - This is a better choice instead of balancing original data

5. Feature Selection
    - Feature Importance in ascending order

	Feature	Feature Importance
36	Zip_100.0	8.005694e-01
3	Hour	1.518322e-01
33	Day of Week_6	1.639918e-02
2	Amount	1.066587e-02
13	MCC_Group_Miscellaneous Stores	9.701689e-03
0	Month	5.830603e-03
1	Day	1.367146e-03
12	MCC_Group_Lodging	9.426547e-04
31	Day of Week_4	8.684024e-04
16	MCC_Group_Transportation Services	7.575979e-04
4	Use Chip_Online Transaction	5.342982e-04
32	Day of Week_5	5.091367e-04
18	Card_0	1.484378e-05
25	Card_7	5.937562e-06
9	MCC_Group_Clothing Stores	1.052573e-06
23	Card_5	4.860596e-13

6. Data Modelling Using ML Algorithm 2: Random Forest
    - Updates on Decision Tree
        i. minimum leaf size = 10
        ii. tree depth = 7 - has the good combination of accuracy, recall, precision, ROC AUC score
        iii. Class_weight – 'balanced' (very high recall 99% but low precision 10%)
        Class_weight – {0 : .1, 1 : .9} (better precision 30% but lower recall 85%)

    - Random Forest
        i. minimum leaf size = 10
        ii. tree depth = 7 -
        iii. Class_weight – {0 : .1, 1 : .9}
        iv. estimators = 200

    - XGBoost
        i. minimum leaf size = 10
        ii. tree depth = 7 - has the good combination of accuracy, recall, precision, ROC AUC score
        iii. estimators = 50 – improved overfitting in precision score vs Decision Tree
        iv. learning_rate=.1

7. Model 2 Performance Evaluation

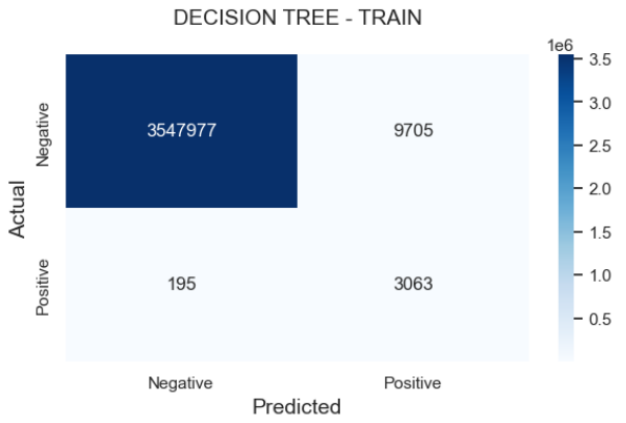
RANDOM FOREST

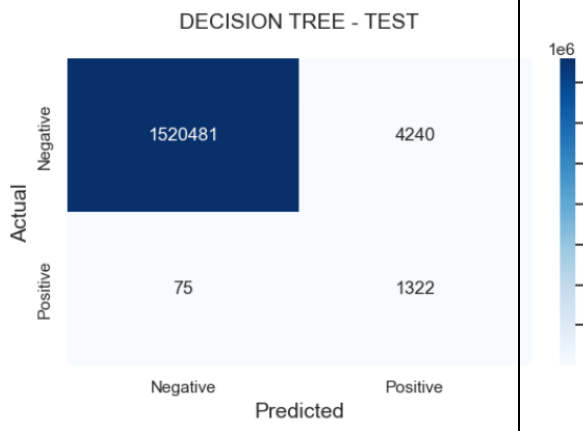
	1 - Yes	0 - No	weighted ave
TRAIN Model Score			
classification accuracy	NaN	NaN	1.00
precision	0.24	1.00	1.00
recall	0.93	1.00	1.00
f1 score	0.38	1.00	1.00
ROC AUC	0.96	0.96	0.96

	1 - Yes	0 - No	weighted ave
TEST Model Score			
classification accuracy	NaN	NaN	1.00
precision	0.24	1.00	1.00
recall	0.93	1.00	1.00
f1 score	0.38	1.00	1.00
ROC AUC	0.97	0.97	0.97

- Recall score for the 'YES Fraud' is 93%, precision of 24% and overall accuracy of 100%

- Result shows good fit with 100% average accuracy but relatively low precision





8. Model Selection
    - The updated performance of the Decision Tree model and Random Forest are compared based on latest data
    - The team has chose **Decision Tree** since it has an overall better performance while also being the least complex

DECISION TREE

TRAIN Model Score	1 - Yes	0 - No	weighted ave
classification accuracy	NaN	NaN	1.00
precision	0.24	1.00	1.00
recall	0.94	1.00	1.00
f1 score	0.38	1.00	1.00
ROC AUC	0.97	0.97	0.97

TEST Model Score	1 - Yes	0 - No	weighted ave
classification accuracy	NaN	NaN	1.00
precision	0.24	1.00	1.00
recall	0.95	1.00	1.00
f1 score	0.38	1.00	1.00
ROC AUC	0.97	0.97	0.97

9. Further Investigation to explain very low precision
    - The 99:1 Class Imbalance greatly affects the result of the precision. In order prove this, we also tried to have the test data balanced which then resulted to both high precision and recall. Hence if we are testing on a balanced data, it would have high precision and recall as is shown in below result:

DECISION TREE

TRAIN Model Score	1 - Yes	0 - No	weighted ave
classification accuracy	NaN	NaN	0.99
precision	0.99	1.00	0.99
recall	1.00	0.99	0.99
f1 score	0.99	0.99	0.99
ROC AUC	0.99	0.99	0.99

TEST Model Score	1 - Yes	0 - No	weighted ave
classification accuracy	NaN	NaN	0.99
precision	0.99	0.99	0.99
recall	0.99	0.99	0.99
f1 score	0.99	0.99	0.99
ROC AUC	0.99	0.99	0.99

- We tried Xgboost which increased precision 70% but brought the recall down to 40% which is not good. But the effect shows that we may need a more complex model if we want to have better score for both precision and recall

XGBoost Results:
XGBOOST

TRAIN Model Score	1 - Yes	0 - No	weighted ave
classification accuracy	NaN	NaN	1.00
precision	0.81	1.00	1.00
recall	0.21	1.00	1.00
f1 score	0.34	1.00	1.00
ROC AUC	0.61	0.61	0.61

TEST Model Score	1 - Yes	0 - No	weighted ave
classification accuracy	NaN	NaN	1.0
precision	0.73	1.0	1.0
recall	0.20	1.0	1.0
f1 score	0.32	1.0	1.0
ROC AUC	0.60	0.6	0.6

Next Steps

Plans for the next Reporting Period:

Delivery

1. Code Freeze
2. Pickle File Generation
3. Deployment
    - Create basic UI to test new data

Issue Status:

- No Issues occurred

Change Status:

- No change needed

Risk Status:

Risk	Impact Level	Mitigation Plan	Status
Imbalanced data results in a data model that may lead to bias toward the majority class and poor predictions for the minority class	High	Apply resampling to make up for the imbalanced data	***Closed***
Poor accuracy in data model due to poor quality of data may lead to difficulty that can cause delay	Medium	Apply rigorous data preprocessing to ensure removal of outliers and irrelevant data	***Closed***
The resulting performance metrics of initial modelling algorithms to be used may not reach required accuracy, causing delay in remodelling	Low	Extensive pre-study on the data and research on best model to use must be done beforehand	***Closed***
There is a chance that the project won't be completed in the allotted time due to a team member's illness	Low	The team will apply effective time management by developing efficient work breakdown where there are primary and secondary members assigned to each task	***Closed***

Notes: Status Definition

Planned *- An identified risk with a risk response plan.*

In-Process *- A risk where the risk response is being executed*

Closed *- A risk that occurred and is transfered to an issue or the risk was solved/avoided*

Not occurred *- A risk that was identified but that did not occur*

Rejected *- Created and kept for tracking pruposes but considered not to be used yet*

Budget Status as of 20/10/2023

Budget Category	Original Budget Cost	Spent to Date	Budget Left
Software Costs			
Application software	$500	$500	$0
Operating system	$1,000	$1,000	$0
Hardware Costs			
PCs	$10,000	$10,000	$0
Network Costs			
Internet	$1,000	$950	$50
Team Resources			
Project Manager ($30/hr) ($240/day)	$12,240	$9,000	$2,240
Technical Lead ($30/hr)	$12,240	$9,000	$2,240
Developer ($30/hr)	$12,240	$9,000	$2,240
Researcher ($30/hr)	$12,240	$9,000	$2,240
Validator ($30/hr)	$12,240	$9,000	$2,240
Total	**$ 73,700**	$57,550	$16,150

Budgetary Comments:

- Still within budget