1. INTRODUCTION

1.1. OVERVIEW

As fraudsters are increasing day by day and fallacious transactions are done by the credit card and there are various types of fraud. So to solve this problem combination of techniques are used.By this transaction is tested individually. And the foremost goal is to detect fraud by filtering the above techniques to get better result.Financial fraud is an ever growing menace with far consequences in the financial industry. Machine Learning has played an imperative role in the detection of credit card fraud in online transactions. Credit card fraud detection, which is a major problem, becomes challenging due to two main reasons - first, the profiles of normal and fraudulent behaviors change constantly and secondly, credit card fraud data sets are highly skewed. The performance of fraud detection in credit card transactions is greatly affected by the sampling approach on dataset, selection of variables and detection techniques used.

1.2. PURPOSE

Credit card frauds are easy and friendly targets. E-commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. Due to the increase in fraud rates, researchers started using different machine learning methods to detect and analyze frauds in online transactions. The main aim of the project is to design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyze the past transaction details of the customers and extract the behavioral patterns, where cardholders are clustered into different groups based on their transaction amount. Later different classifiers are trained over the groups separately. And then the classifier with better rating score can be chosen to be one of the best methods to predict frauds.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEM

Credit card generally refers to a card that is assigned to the customer (cardholder), usually allowing them to purchase goods and services within credit limit or withdraw cash in advance. Credit card provides the cardholder an advantage of the time, i.e., it provides time for their customers to repay later in a prescribed time, by carrying it to the next billing cycle. Credit card frauds are easy targets. Without any risks, a significant amount can be withdrawn without the owner's knowledge, in a short period. Fraudsters always try to make every fraudulent transaction legitimate, which makes fraud detection very challenging and difficult task to detect. In 2017, there were 1,579 data breaches and nearly 179 million records among which Credit card frauds were the most common form with 133,015 reports, then employment or tax-related frauds with 82,051 reports, phone frauds with 55,045 reports followed by bank frauds with 50,517 reports from the statics.With different frauds mostly credit card frauds are often in the news for the past few years, frauds are in the top of mind for most the world's population. Credit card dataset is highly imbalanced because there will be more legitimate transaction when compared with a fraudulent one. As advancement, banks are moving to EMV cards, which are smart cards that store their data on integrated circuits rather than on magnetic stripes, have made some on-card payments safer, but still leaving card-not-present frauds on higher rates. According to 2017, the US Payments Forum report, criminals have shifted their focus on activities related to CNP transactions as the security of chip cards were increased.Even then there are chances for thieves to misuse the credit cards. There are many machine learning techniques to overcome this problem.

2.2. PROPOSED SOLUTION

Multiple Supervised and Semi-Supervised machine learning techniques are used for fraud detection, but our aim is to overcome three main challenges with card frauds related dataset i.e., strong class imbalance, the inclusion of labelled and unlabelled samples, and to increase the ability to process a large number of transactions. Different Supervised machine learning algorithms like Decision Trees, Naive Bayes Classification, Least Squares Regression, Logistic Regression and SVM are used to detect fraudulent

transactions in real-time datasets. Two methods under random forests are used to train the behavioral features of normal and abnormal transactions. They are Random-tree-based random forest and CART-based. Even though random forest obtains good results on small set data, there are still some problems in case of imbalanced data. The future work will focus on solving the above-mentioned problem. The algorithm of the random forest itself should be improved. Performance of Logistic Regression, K-Nearest Neighbor, and Naïve Bayes are analyzed on highly skewed credit card fraud data where Research is carried out on examining meta-classifiers and meta-learning approaches in handling highly imbalanced credit card fraud data. Though supervised learning methods can be used they may fail in certain cases of detecting the fraudulent transactions.

Fraud Prediction using Auto AI

Automation and artificial intelligence (AI) are transforming businesses and will contribute to economic growth via contributions to productivity. They will also help address challenges in areas of healthcare, technology & other areas. At the same time, these technologies will transform the nature of work and the workplace itself. In this code pattern, we will focus on building state of the art systems for churning out predictions which can be used in different scenarios. We will try to predict fraudulent transactions which we know can reduce monetary loss and risk mitigation. The same approach can be used for predicting customer churn, demand and supply forecast and others. Building predictive models require time, effort and good knowledge of algorithms to create effective systems which can predict the outcome accurately. With that being said, IBM has introduced Auto AI which will automate all the tasks involved in building predictive models for different requirements. We will get to see how Auto AI can churn out great models quickly which will save time and effort and aid in faster decision making process.

When the reader has completed this code pattern, they will understand how to :

- Quickly set up the services on cloud for model building.
- Ingest the data and initiate the Auto AI process.
- Build different models using Auto AI and evaluate the performance.
- Choose the best model and complete the deployment.
- Generate predictions using the deployed model by making ReST calls.
- Compare the process of using Auto AI and building the model manually.

Architecture Diagram

1. User logs into Watson Studio, creates a project and initiates an instance of Auto AI & Object Storage.
2. User uploads the data file in the CSV format to the object storage.
3. User initiates the model building process using Auto AI and create pipelines.
4. User evaluates different pipelines from Auto AI and selects the best model for deployment.
5. User generates accurate predictions by making ReST call to the deployed model.

Included components

- IBM Watson Studio: Analyze data using RStudio, Jupyter, and Python in a configured, collaborative environment that includes IBM value-adds, such as managed Spark.
- IBM Auto AI:The AutoAI graphical tool in Watson Studio automatically analyzes your data and generates candidate model pipelines customized for your predictive modeling problem.
- IBM Cloud Object Storage: An IBM Cloud service that provides an unstructured cloud data store to build and deliver cost effective apps and services with high reliability and fast speed to market. This code pattern uses Cloud Object Storage.

Featured technologies

- Artificial Intelligence: Any system which can mimic cognitive functions that humans associate with the human mind, such as learning and problem solving.
- Data Science: Systems and scientific methods to analyze structured and unstructured data in order to extract knowledge and insights.
- Analytics: Analytics delivers the value of data for the enterprise.
- Python: Python is a programming language that lets you work more quickly and integrate your systems more effectively.

Steps using AutoAI

Follow these steps to setup and run this code pattern using Auto AI.

1. Create an account with IBM Cloud
2. Create a new Watson Studio project
3. Add Data
4. Add Asset as Auto AI
5. Create and define experiment
6. Import the csv file

7. Run experiment
8. Analyze results
9. Deploy to Cloud
10. Model testing

1. Create an account with IBM Cloud

Sign up for IBM Cloud. By clicking on create a free account you will get 30 days trial account.

2. Create a new Watson Studio project

Sign up for IBM's Watson Studio.

Click on New Project and select per below.

Define the project by giving a Name and hit 'Create'.

3. Add Data

Clone this repo Navigate to data and save the file on the disk. Review the data glossary from the data folder for more details. Note: Citation is needed to use this dataset for any other projects.

Click on Assets and select Browse and add the csv file from your file system.

4. Add Asset as Auto AI

Click on Add to project and select AutoAI experiment.

Note: The Lite account for AutoAI comes with 50 capacity units per month and AutoAI consumes 20 capacity units per hour.

5. Create and define experiment

Click on New AutoAI experiment and give a name to the experiment.

Click on Associate a Machine Learning service instance to this project and select the Machine Learning service instance and hit reload. If you do not have Machine Learning service instance, then follow the steps on your screen to get one.

The Create button at the bottom right gets highlighted, go ahead and hit Create.

6. Import the csv file

We need to import the csv file into the experiment. Note that, only csv file format is supported in AutoAI. Click on Browse or Select from project to choose the fraud_dataset.csv file to import.

7. Run experiment

We have to select the target variable, in this case it is Fraud_Risk. Notice that Prediction Type and Optimized Metric get highlighted which tells us that we are working on Binary Classification use case and the evaluation metric is ROC (Receiver Operating Characteristics) & AUC (Area Under The Curve) which is used for classification usecases.

We can click on experiment settings to adjust the holdout sample and training sample under source settings.

We can click on prediction setting to modify the Prediction type, Positive Class & Optimized metric if required. In this case, we will leave'em as is and hit save and close.

Click on Run experiment.

8. Analyze results

The AutoAI experiment has been completed in 97 seconds to generate four pipelines. The duration of experiment depends completly on the size of the dataset. AutoAI

selects the appropriate machine learning algorithm (in the fifth stage of the process under Model Selection) which is best suited for the dataset.

Each pipeline is run with different parameters, pipeline 3 is run on a sequence of HPO (hyper parameters optimization) & FE (feature engineering) where as pipeline 4 includes HPO (hyper parameters optimization), FE (feature engineering) and a combination of both. All these are done on the fly! Isn't it amazing that we just have to sit and watch while AutoAI takes care of things for us and generates awesome machine learning models!! There's very minimal intervention required to get things going and in no time we have the generated pipelines to choose from.

Click on pipeline 3 (which is ranked 1) to see the evaluation metrics on the left side.

Click on model evaluation to review the performance of the model on the hold out sample and cross validation score. We can observe that our model has done very well by scoring > 95% on Recall, average Precision scores & Area under the curve scores. These scores also mean that our model is able to remember and identify fraudulent transactions with great precision.

Click on feature importance to identify the significant features influencing the outcome. Any variable which starts with Newfeature is a variable generated on the fly by the model as part of feature engineering.

Click on feature transforms to understand the transformation of original features to new features. Feature engineering is one of the important factors in the model building process which has a direct impact on the overall accuracy of the model. We can observe that total features are 24 where as the original dataset had 13 variables which means 11 new features have been created by AutoAI which is one of the reasons for high accuracy of the model.

After all the analysis of model performance, its time to select the model for deployment. We will go ahead and select pipeline 3 which is Rank 1 and hit on Save as model. We can select any of the pipelines to be saved which has highest Accuracy or any other evaluation metrics.

9. Deploy to Cloud

The saved model can be found under Models under the project in Watson Studio. Click on three dots on the right side below Actions and hit Promote. Click the Promote to

deployment space. Choose an existing deployment space or create a new one. Click Add Deployment.

In the page that opens, fill in the fields: Specify a name for the deployment. Select “Web service” as the Deployment type. Click Save.

Define the deployment by giving a name and hit Save. Note that, the model will get deployed as web service as a ReST API.

After you save the deployment, click on the deployment name from the left navigation pane to view the deployment details page. The deployment will get initialized and the status will show as ready when it is complete.

We can click on deployed model to see three tabs, Overview, Implementation and Test. Overview tab will give all details about the deployment like name, type, status et'al. Implementation tab will give scoring endpoint and code snippets to invoke the model. Test tab will give options to test the model.

10. Model testing

Now that we have created and deployed the model as a web service, how do we test it?? We have to click on Test tab which will have two options which are form and json. We can use form if we are to test one record at a time where we can give the values to each attribute manually and hit Predict to generate predictions. The output of 0 under values indicate that it is a fraudulent transaction. The output can be either 0 or 1 as per the data glossary provided in the data folder.

For predicting multiple records, we have to update the values in the json file and use the option to input json data & then hit Predict to generate real time predictions.

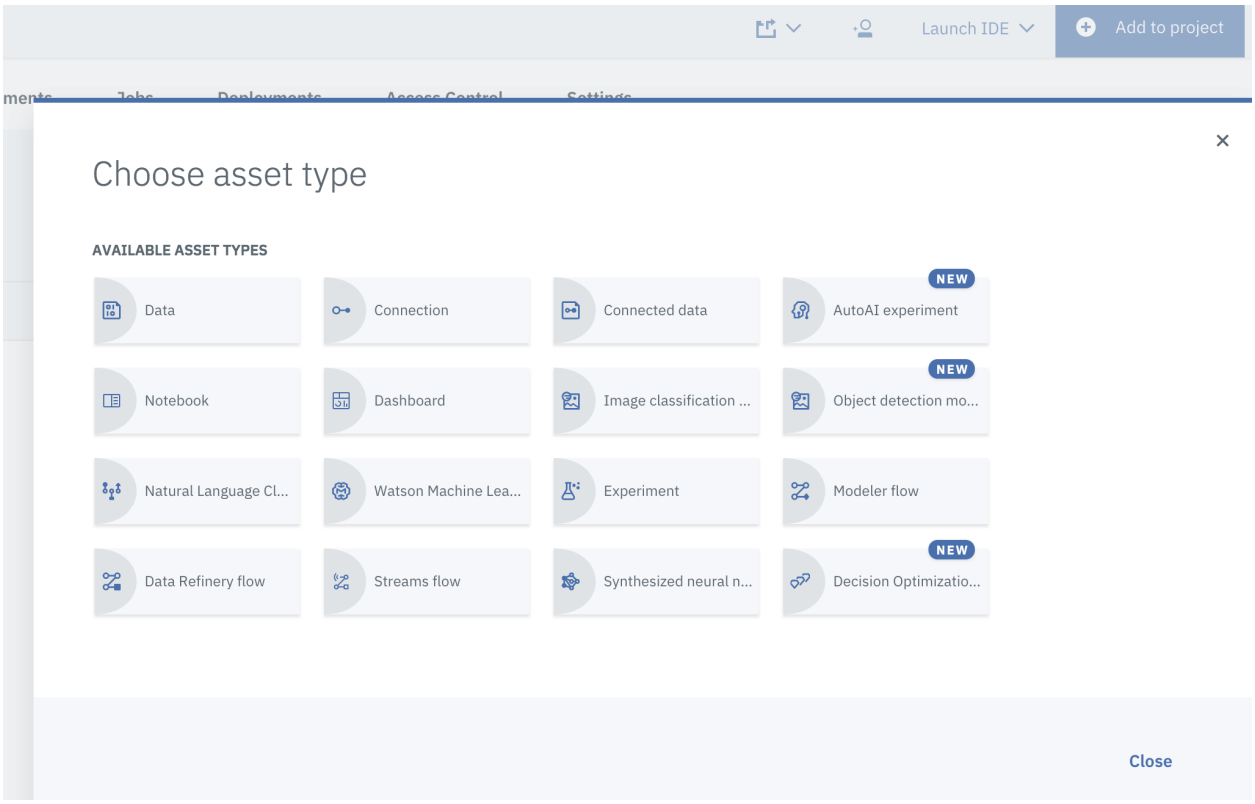
A sample json file has been provided for testing purpose. The format for scoring the model has to be same as given in json file. Navigate to data-for-testing and save the file on the disk. Copy and paste the values in the test tab as shown above to generate predictions.

Go ahead and give it a try on different datasets as per your requirement and realize the

ease of creating and deploying models quickly using AutoAI offering by IBM.

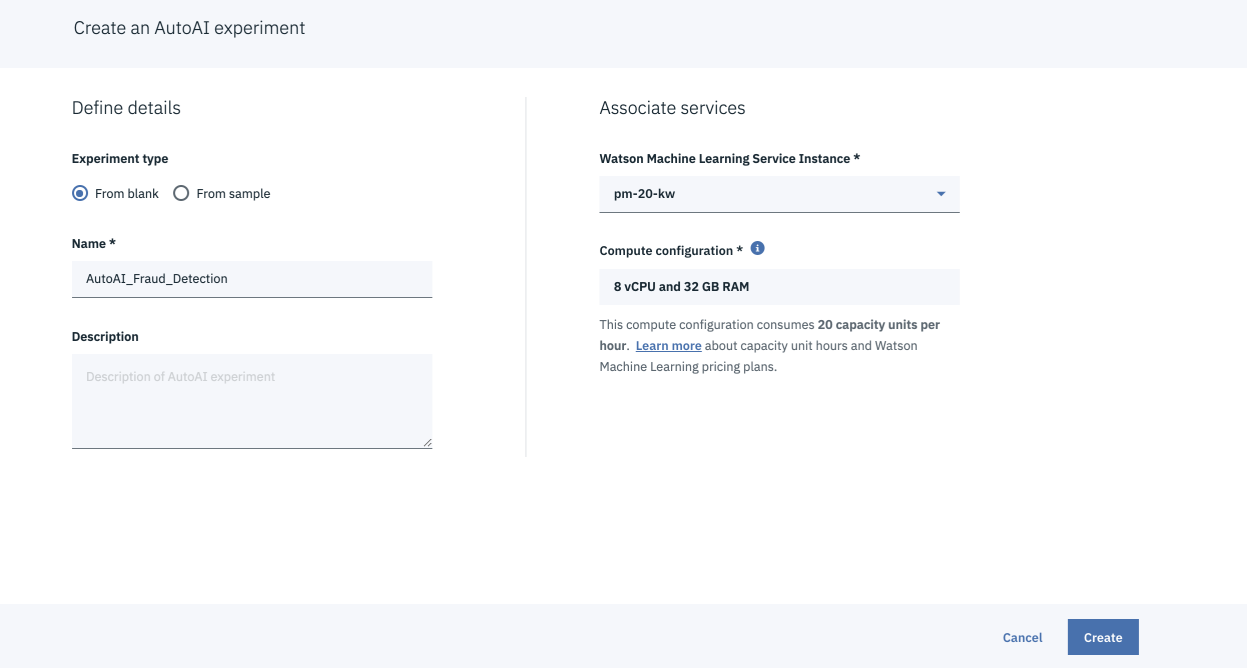
1. Add Asset as Auto AI

- Click on **Add to project** and select AutoAI experiment.



- Note: The Lite account for AutoAI comes with 50 capacity units per month and AutoAI consumes 20 capacity units per hour.
- Give a name to the experiment. If you have a Watson Machine Learning Service Instance associated with this project, it will automatically show up. The **Create** button at the bottom right gets highlighted, go ahead and hit Create.

Create Experiment



Note: If you dont have one, click on **Associate a Machine Learning service instance** to this project and select the Machine Learning service instance and hit reload. If you do not have Machine Learning service instance, then follow the steps on your screen to get one.

2. Import data

- We need to import the csv file into the experiment. Note that, only csv file format is supported in AutoAI. Click on **Browse** or **Select from project** to choose the *insurance_claims.csv* file to import.

Adding CSV file

Drop a .csv file here or **browse** for a file
to upload. Maximum file size is 100 MB.

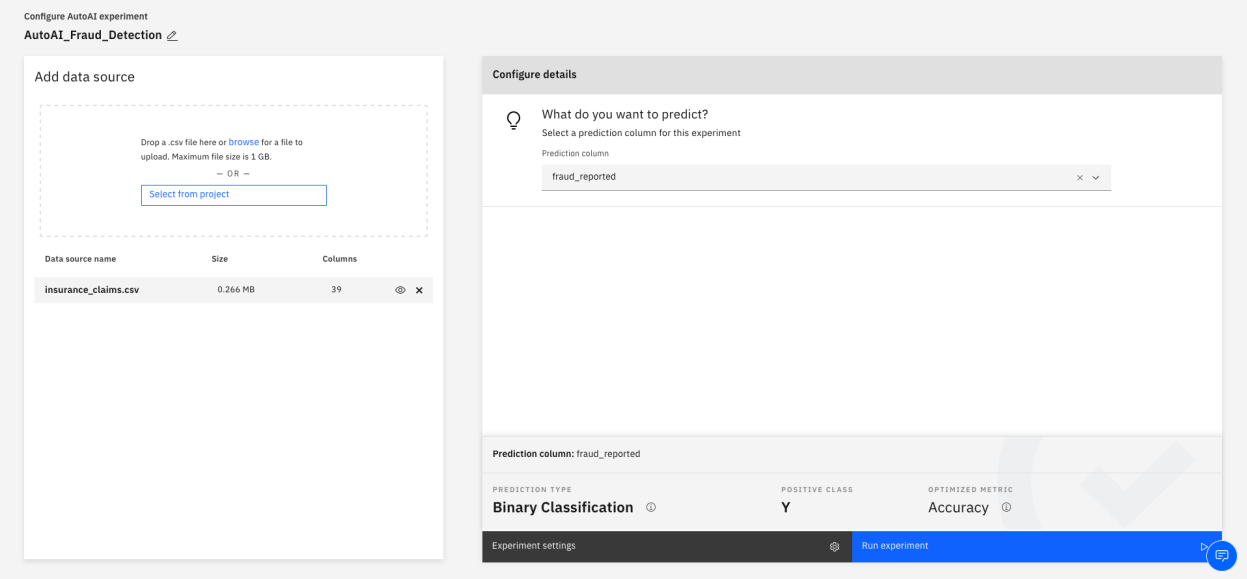
— OR —

Select from project

3. Configure AutoAI experiment

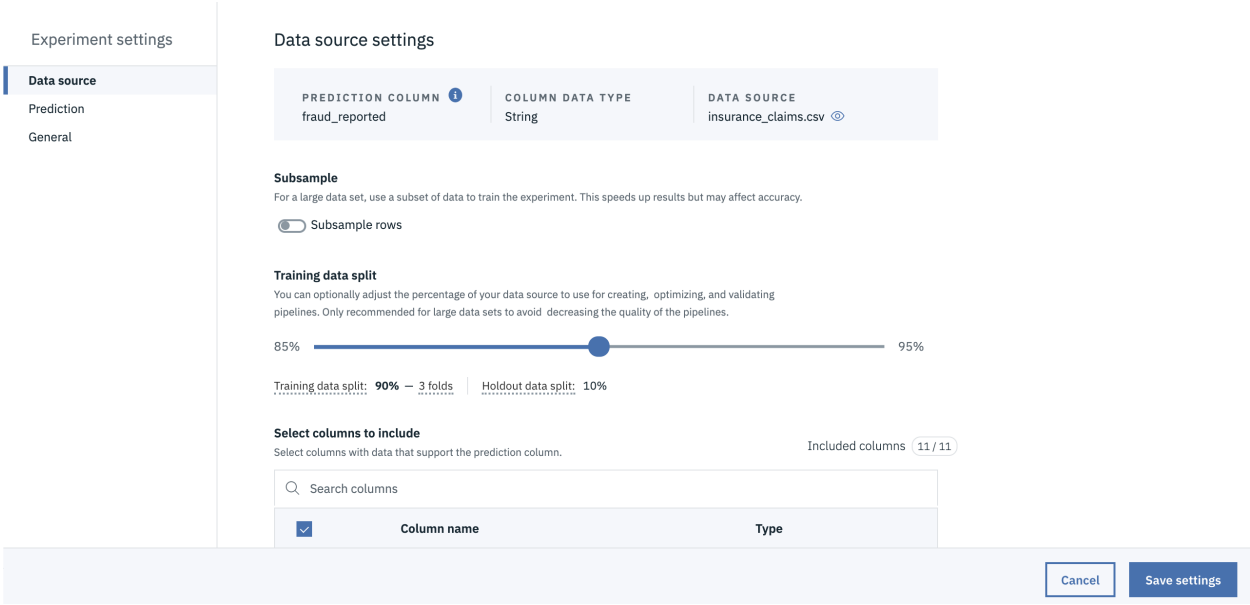
- We have to select the target variable, in this case it is **fraud_reported**. Notice that **Prediction Type** and **Optimized Metric** get highlighted which tells us that we are working on **Binary Classification** use case and the evaluation metric is **ROC** (Receiver Operating Characteristics) & **AUC** (Area Under The Curve) which is used for classification usecases.

Configure Experiment



- We can click on experiment settings for more customized configurations such as the training vs. holdout data split ratio, input columns etc. Save settings. Then click **Run experiment**.

Experiment Setting



4. Analyze results

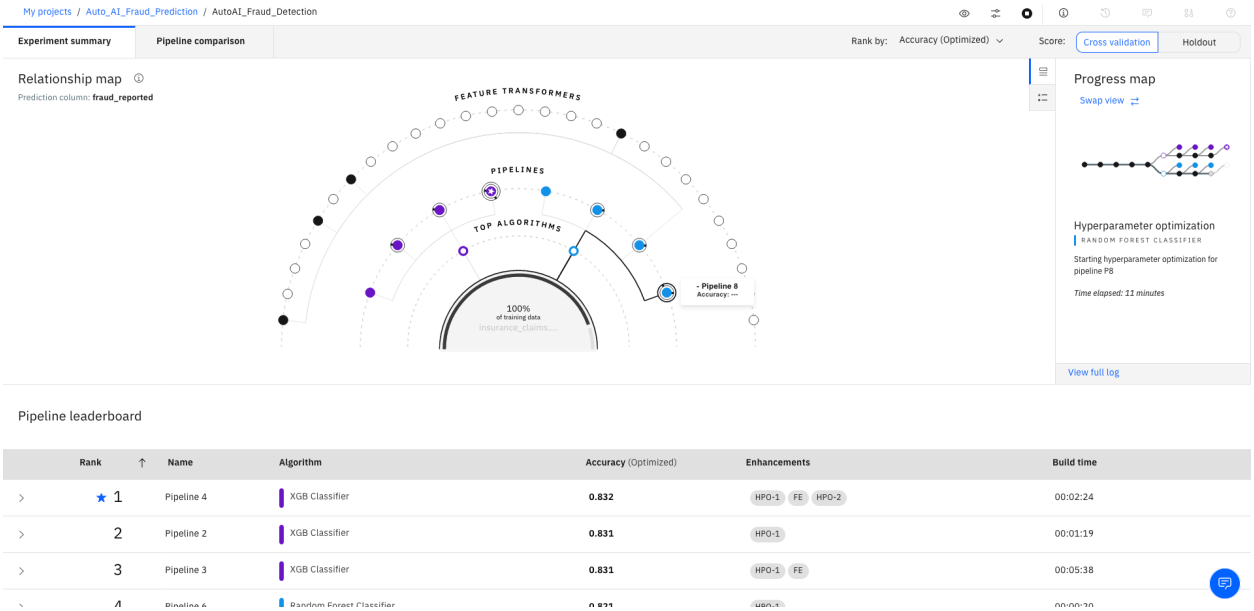
- The AutoAI experiment will generate four pipelines. which will be ranked based on the evaluation metrics. AutoAI selects the appropriate machine learning

algorithm which is best suited for the dataset.

- The duration of experiment depends completly on the size of the dataset.

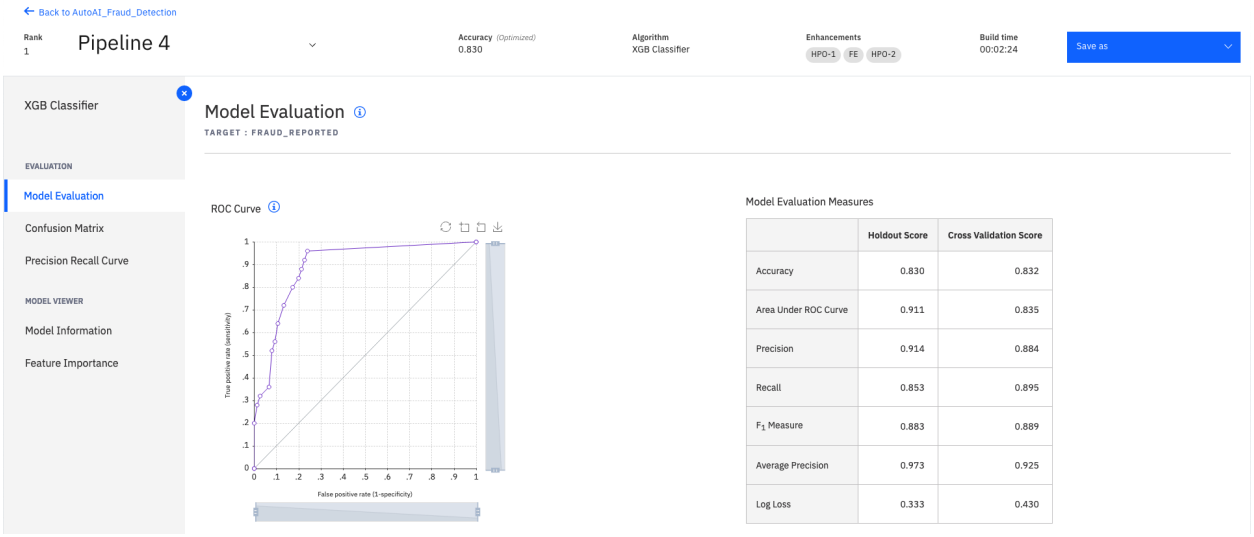
Each pipeline is run with different parameters, pipeline 3 is run on a sequence of HPO (hyper parameters optimization) & FE (feature engineering) where as pipeline 4 includes HPO (hyper parameters optimization), FE (feature engineering) and a combination of both. All these are done on the fly! Isn't it amazing that we just have to sit and watch while AutoAI takes care of things for us and generates awesome machine learning models!! There's very minimal intervention required to get things going and in no time we have the generated pipelines to choose from.

Relationship Map



- Click on **Pipeline 4** (which is ranked 1) to check model details.

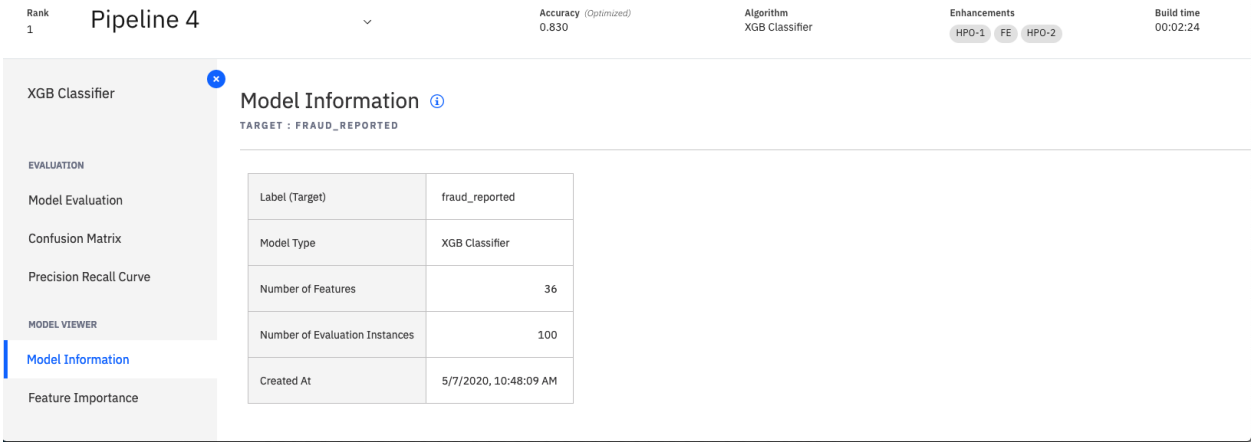
Model Evaluation



The model evaluation tab shows the performance of the model on the hold out sample and cross validation score. We can observe that our model has done very well by scoring > 95% on Recall, average Precision scores & Area under the curve scores. These scores also mean that our model is able to remember and identify fraudulent claims with great precision.

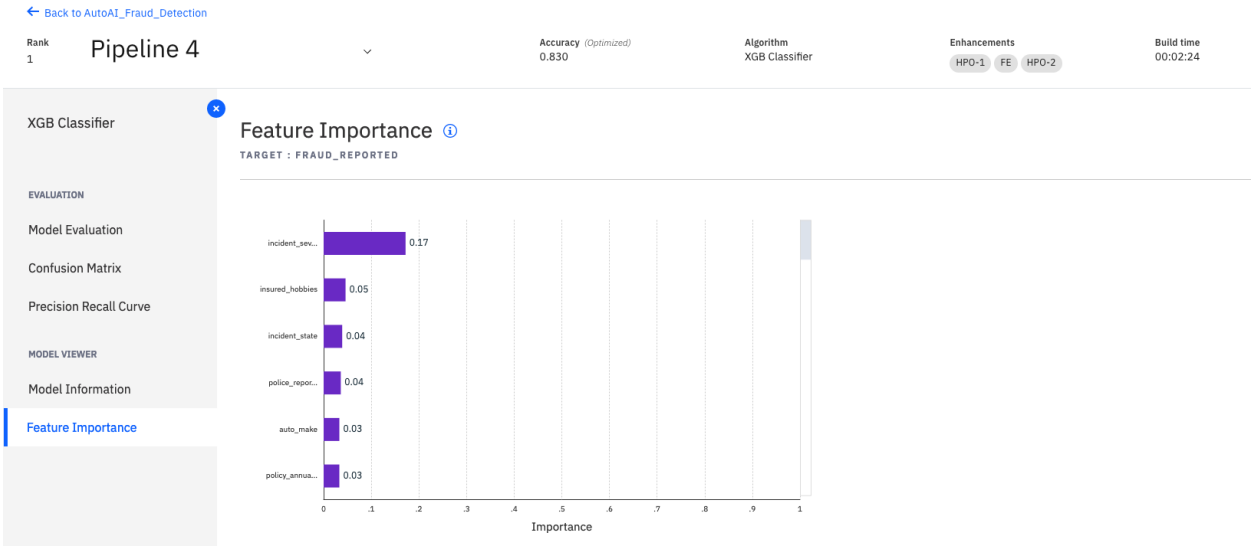
- Click on the **Model Information** tab, we can observe that XGB Classifier has been used with total features of 22 where as the original dataset had 10 variables which means 12 new features have been created by AutoAI which is one of the reasons for high accuracy of the model. Check feature transformations to understand the transformation of original features to new features. **Feature engineering** is one of the important phases in the model building process which has a direct impact on the overall accuracy of the model.

Model Information



- Click on **feature importance** to identify the significant features influencing the outcome. Any variable which starts with Newfeature is a variable generated on the fly by the model as part of feature engineering.

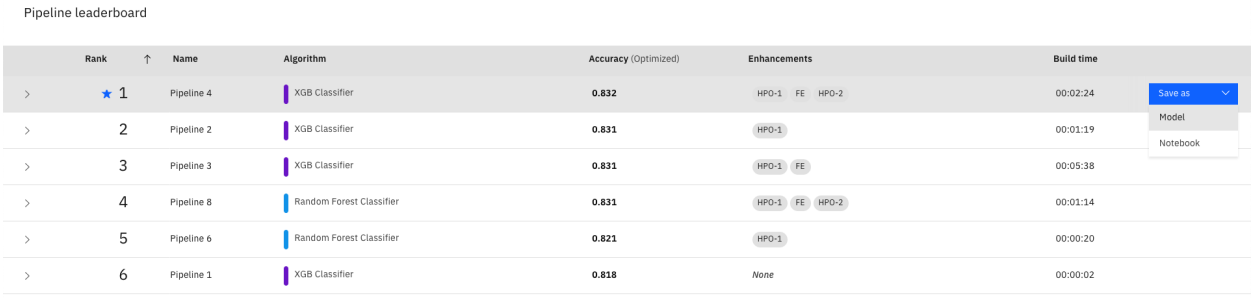
Feature Importance



5. Save Model

- After all the analysis of model performance, it's time to select the model for deployment. We will go ahead and select pipeline 4 which is Rank 1 and click on **Save as model**. We can select any of the pipelines to be saved which has highest Accuracy or any other evaluation metrics.

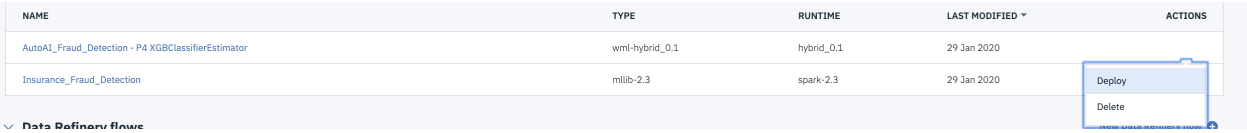
Save Model



Note: You can also save the model as a Notebook, which will generate all the python code of the model.

- The saved model can be found under **Models** under the project in Watson Studio. Click on three dots on the right side below **Actions** and hit **Deploy**.

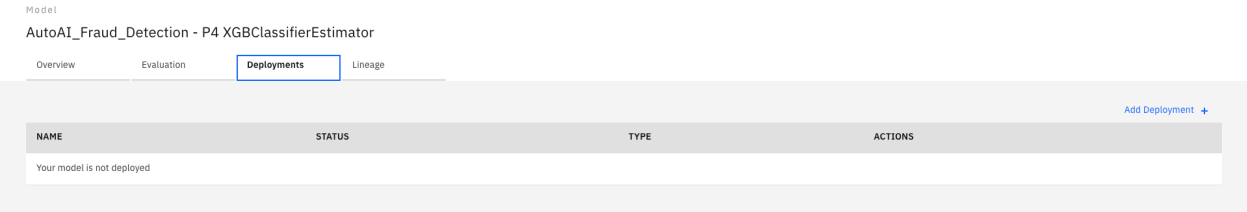
Deploy Model



6. Deploy Model

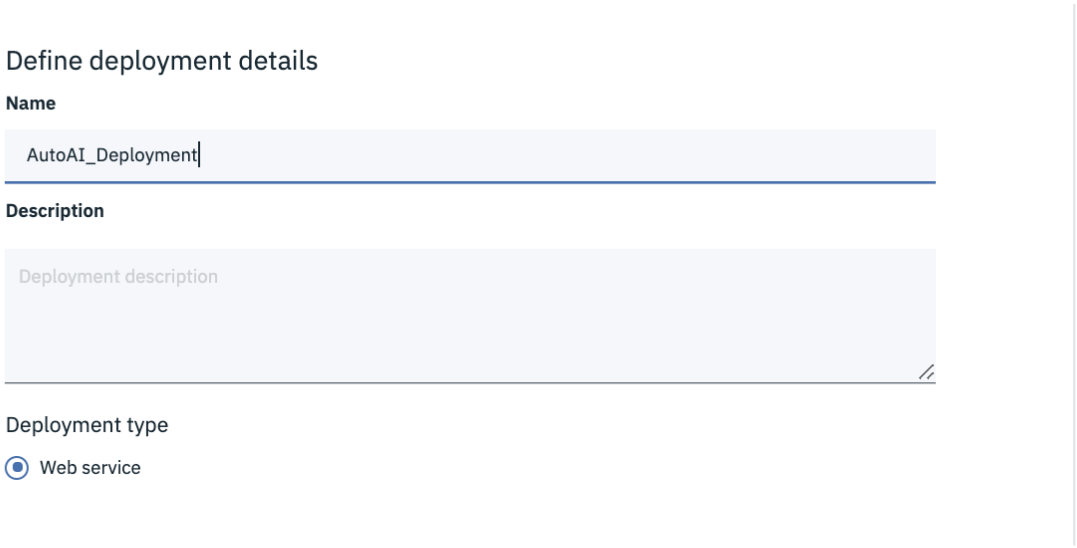
- Click on **Add Deployment** on the right side above **Actions**.

Add Deployment



- Define the deployment by giving a name and hit **Save**. Note that, the model will get deployed as web service as a ReST API. The deployment should be ready soon. Wait until the deployment status is *DEPLOY_SUCCESS*. Fresh the page if needed.

Define Deployment Details

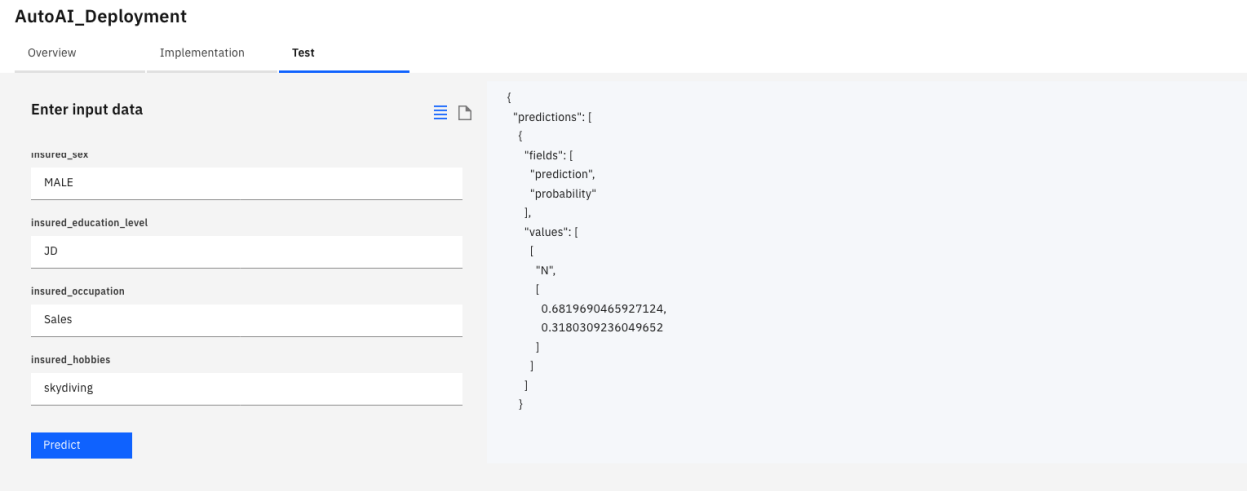


7. Test Deployment Endpoint

Now, the model is deployed and can be used for prediction. However, before using it in a production environment it might be worthwhile to test it using real data. You can do this interactively or programmatically using the API for the IBM Machine Learning Service. For now, we test it interactively.

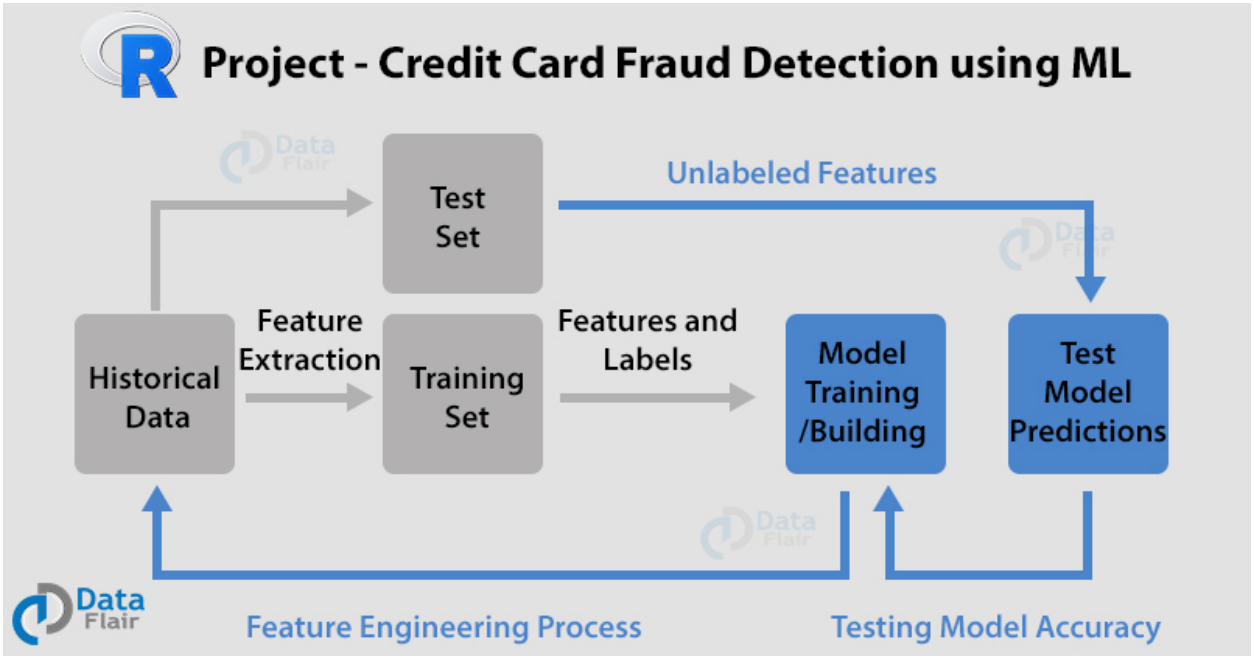
The UI provides two options for testing the prediction: by entering the values one by one in distinct fields (one for each feature) or by specifying the feature values using a JSON object. We usually use the second option because it is the most convenient one when tests are performed more than once (which is usually the case), and when a large set of feature values is needed.

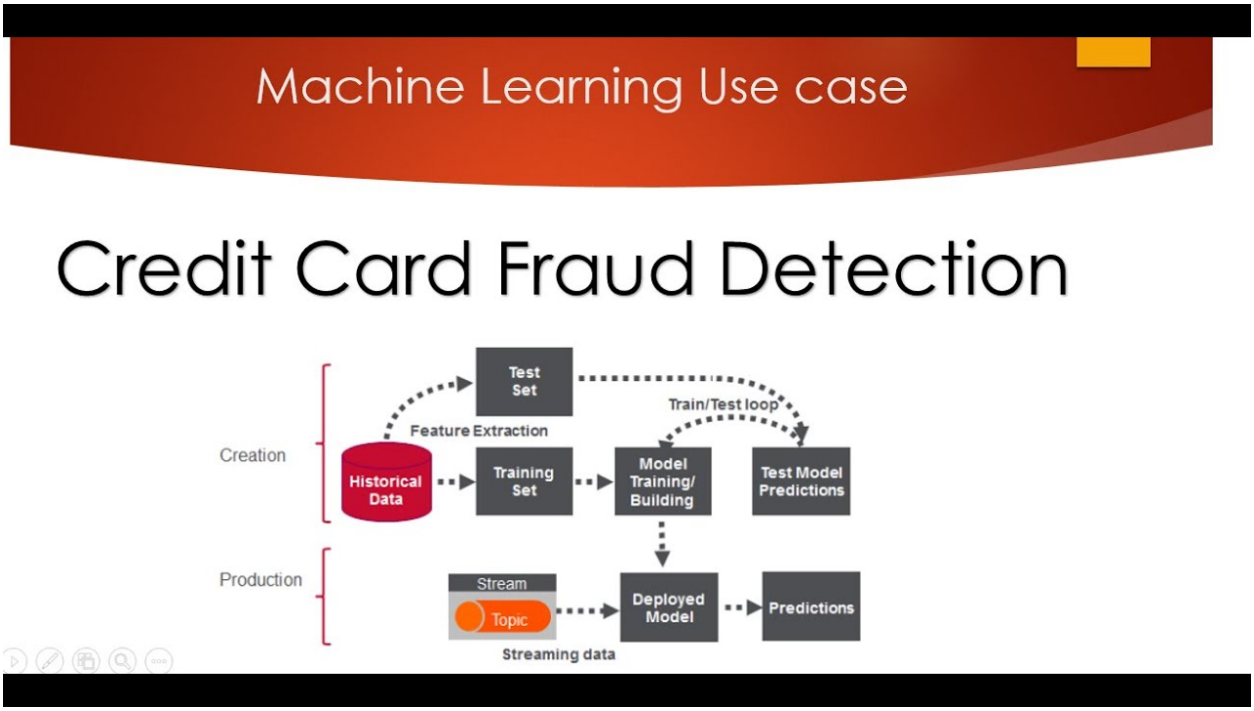
Predict with distinct fields

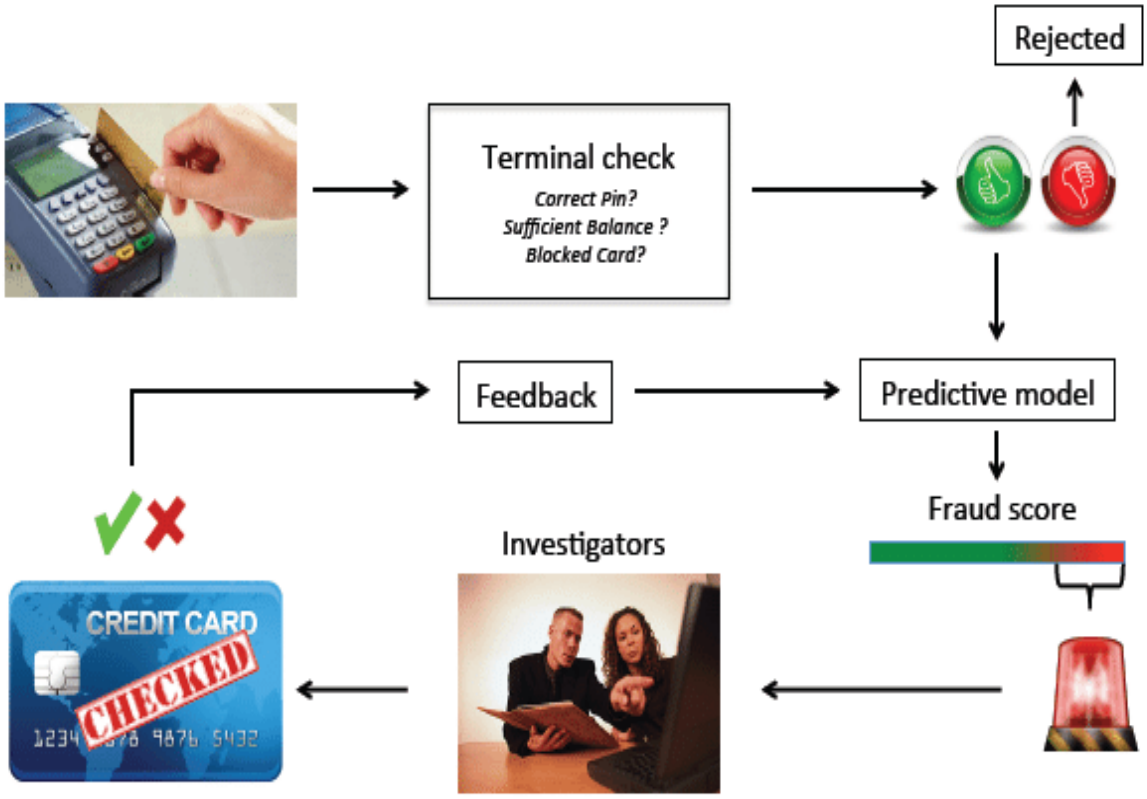


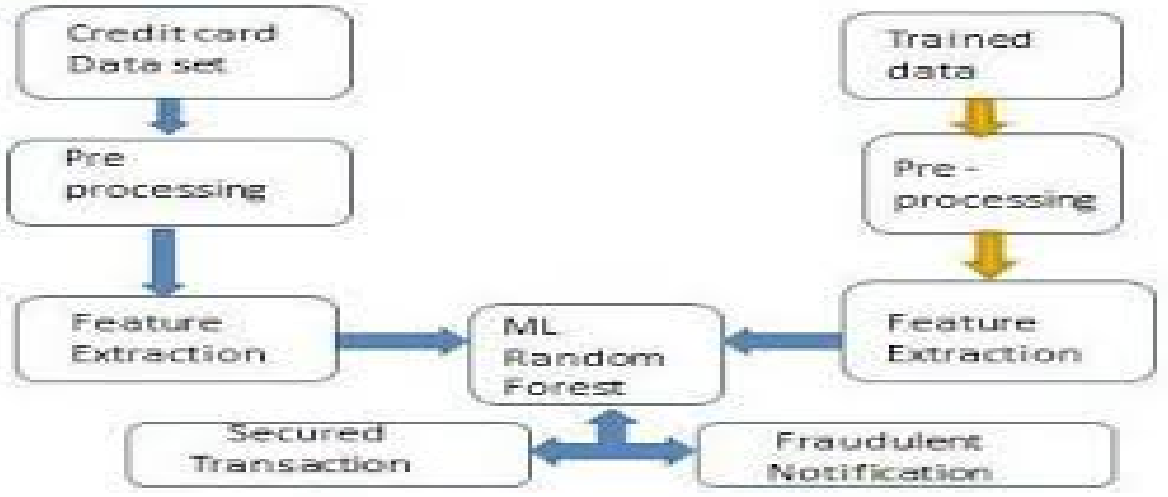
3. THEORITICAL ANALYSIS

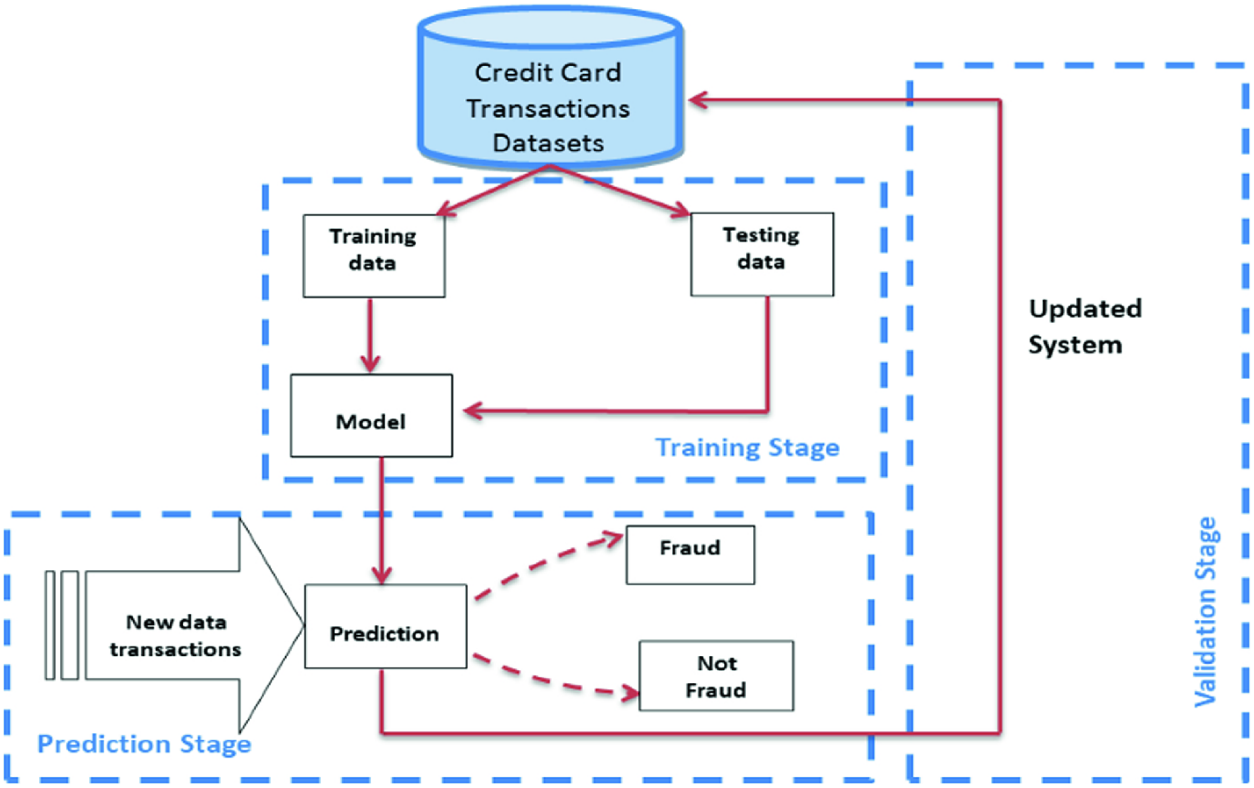
3.1. BLOCK DIAGRAM



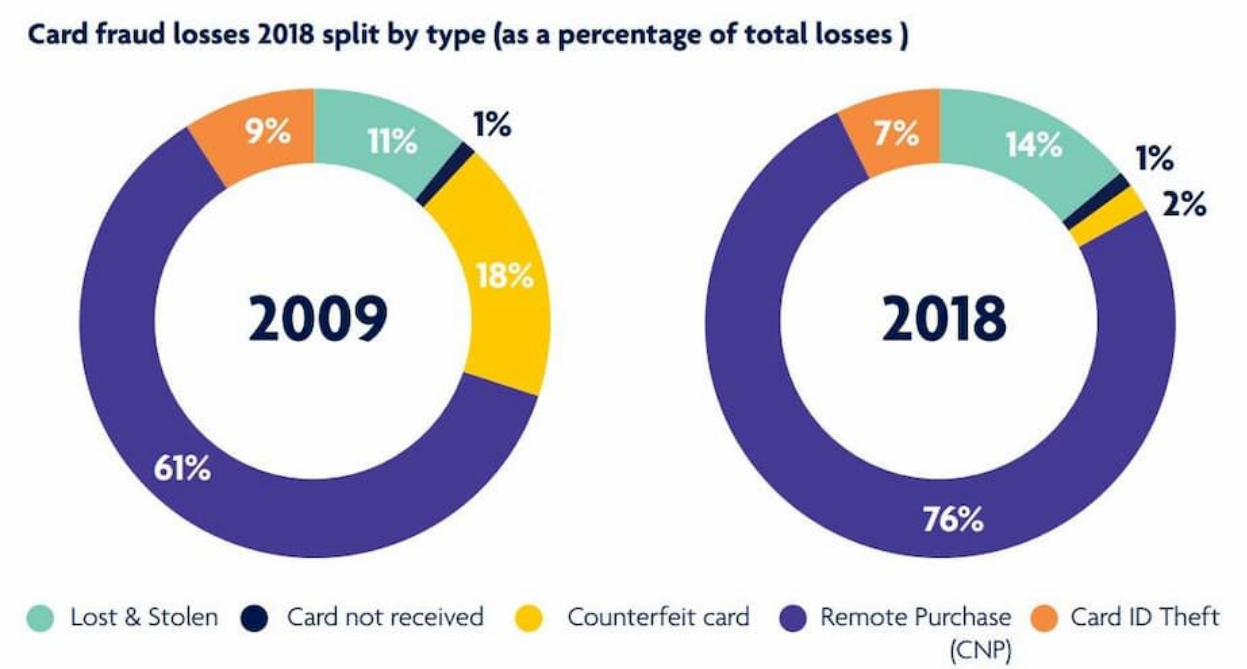



Rejected
Terminal check
Correct Pin?
Sufficient Balance ?
Blocked Card?
Feedback
Predictive model
Fraud score
Investigators
CREDIT CARD
CHECKED

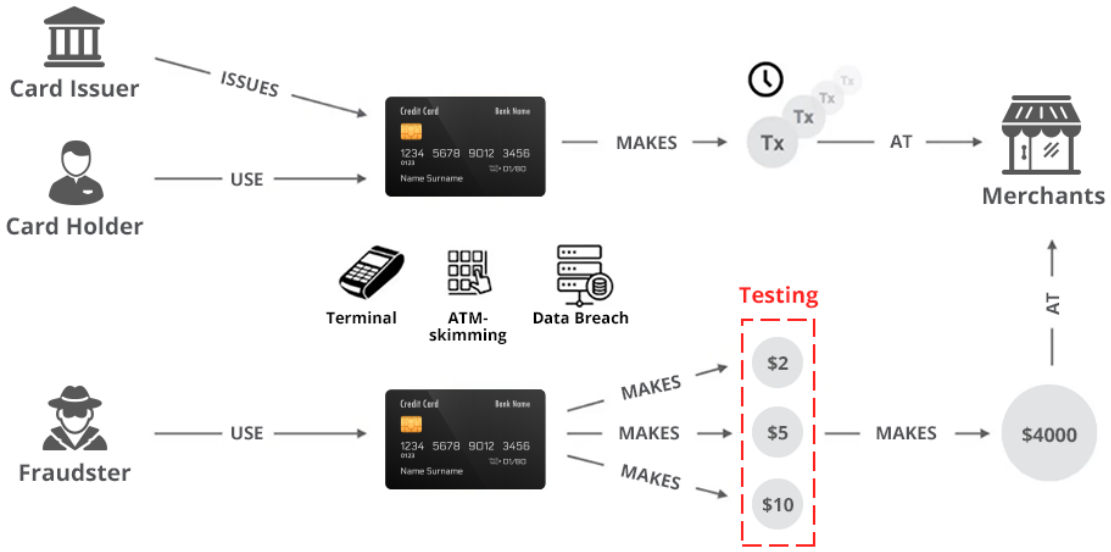

Credit card Data set
Pre processing
Feature Extraction
Trained data
Pre - processing
Feature Extraction
ML Random Forest
Secured Transaction
Fraudulent Notification

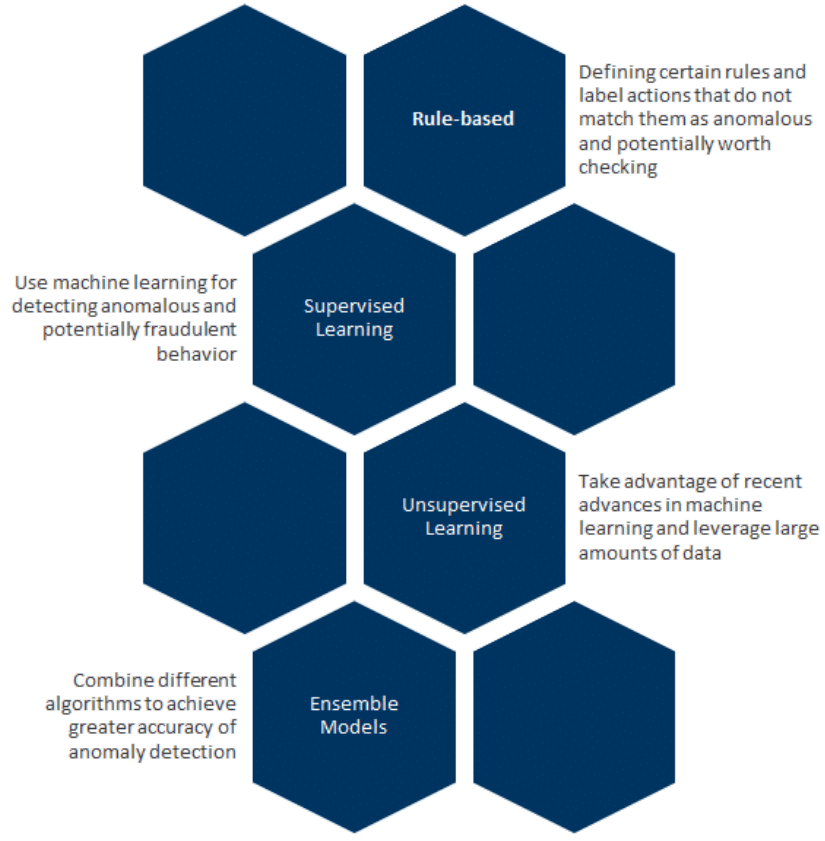


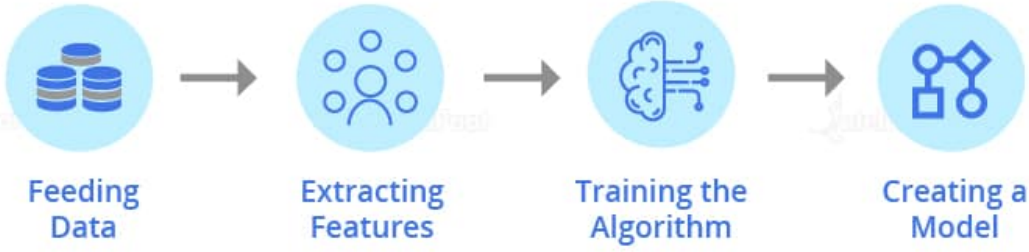
3.2. HARDWARE/SOFTWARE DESIGNING

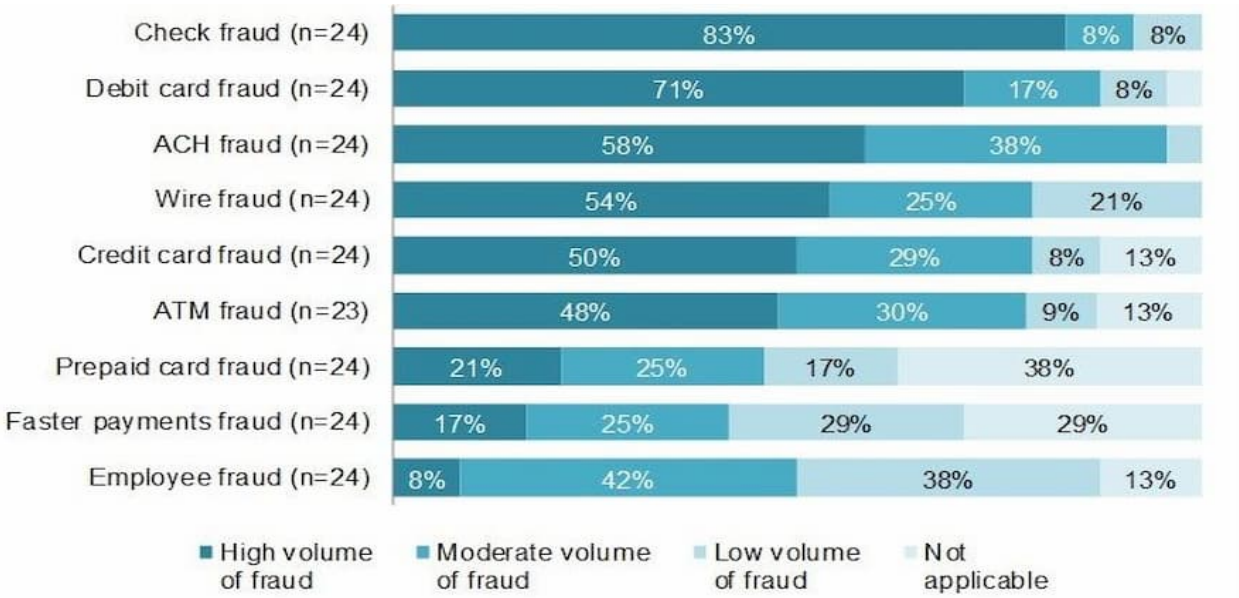


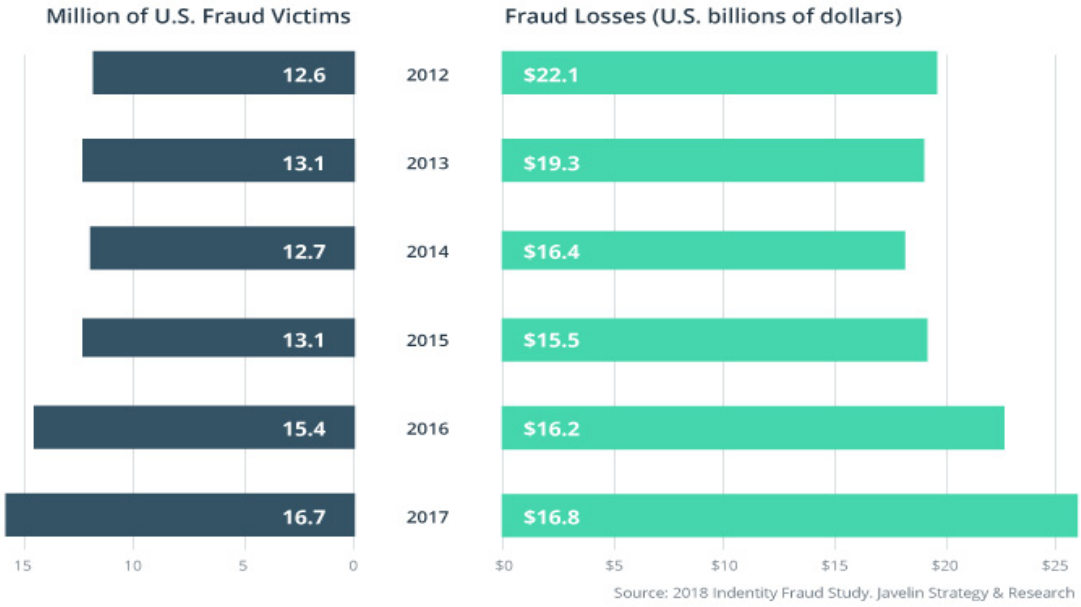
Credit Card Testing

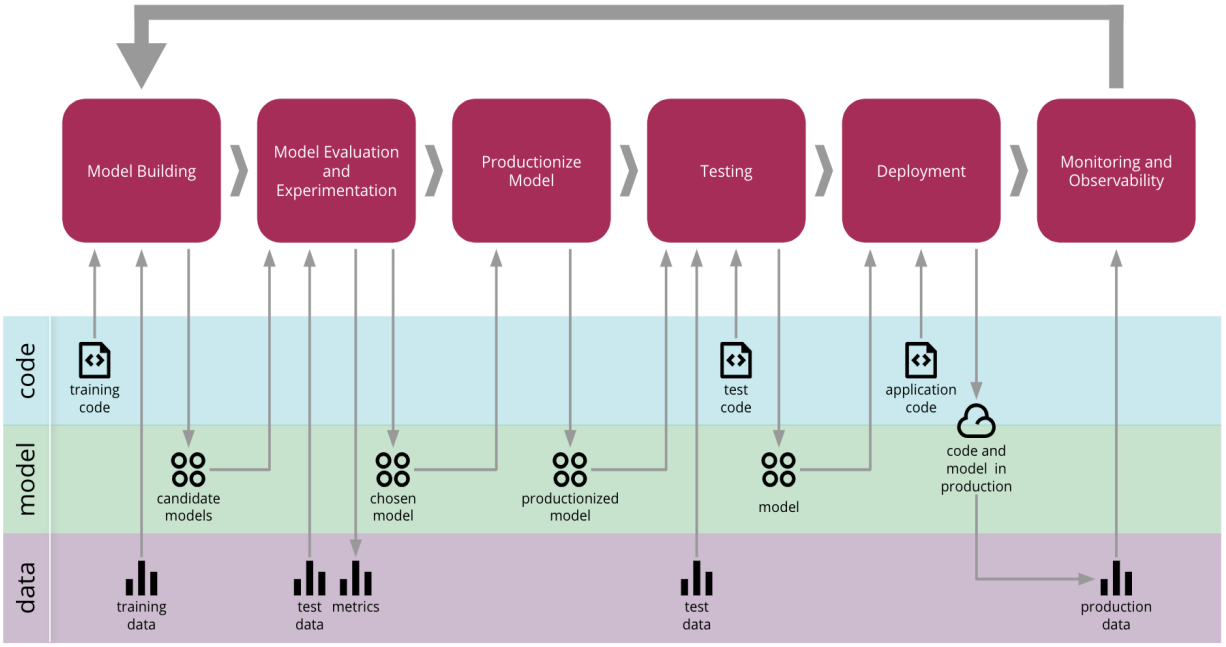





Feeding Data
Extracting Features
Training the Algorithm
Creating a Model


Check fraud (n=24) 83% 8% 8%
Debit card fraud (n=24) 71% 17% 8%
ACH fraud (n=24) 58% 38%
Wire fraud (n=24) 54% 25% 21%
Credit card fraud (n=24) 50% 29% 8% 13%
ATM fraud (n=23) 48% 30% 9% 13%
Prepaid card fraud (n=24) 21% 25% 17% 38%
Faster payments fraud (n=24) 17% 25% 29% 29%
Employee fraud (n=24) 8% 42% 38% 13%
High volume of fraud
Moderate volume of fraud
Low volume of fraud
Not applicable


Million of U.S. Fraud Victims
Fraud Losses (U.S. billions of dollars)
12.6 2012 $22.1
13.1 2013 $19.3
12.7 2014 $16.4
13.1 2015 $15.5
15.4 2016 $16.2
16.7 2017 $16.8
15 10 5 0
$0 $5 $10 $15 $20 $25
Source: 2018 Indentity Fraud Study. Javelin Strategy & Research



63% of businesses have experienced the same or more fraud losses in the past 12 months

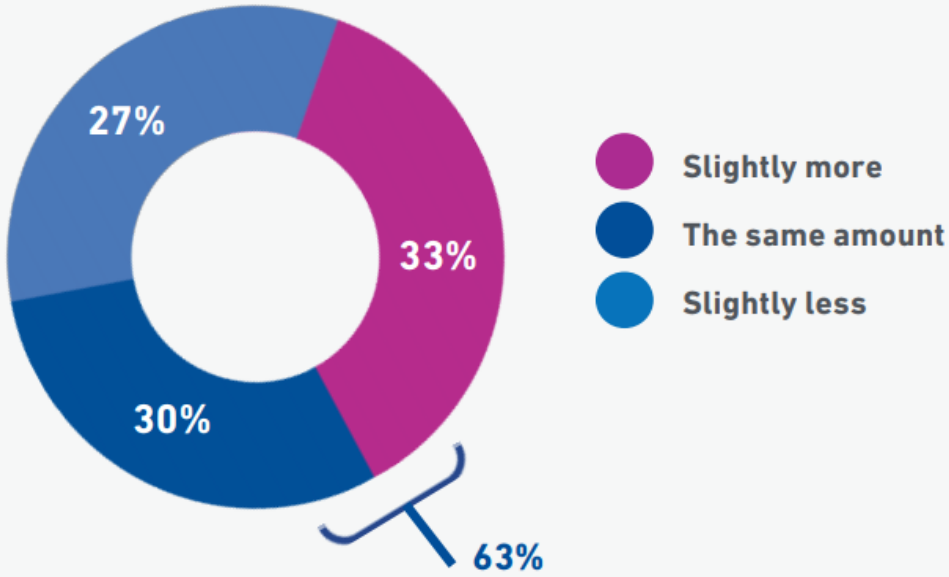
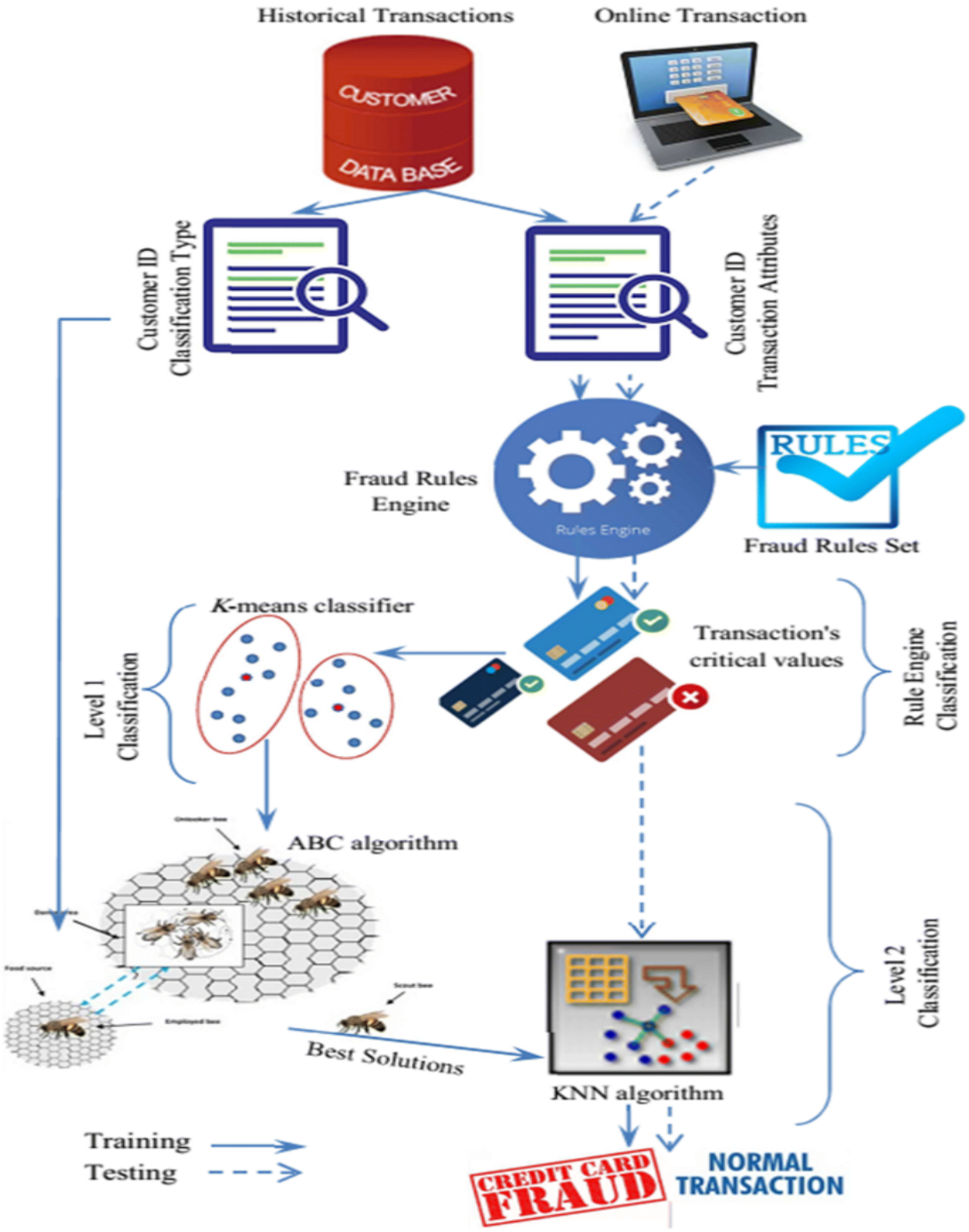
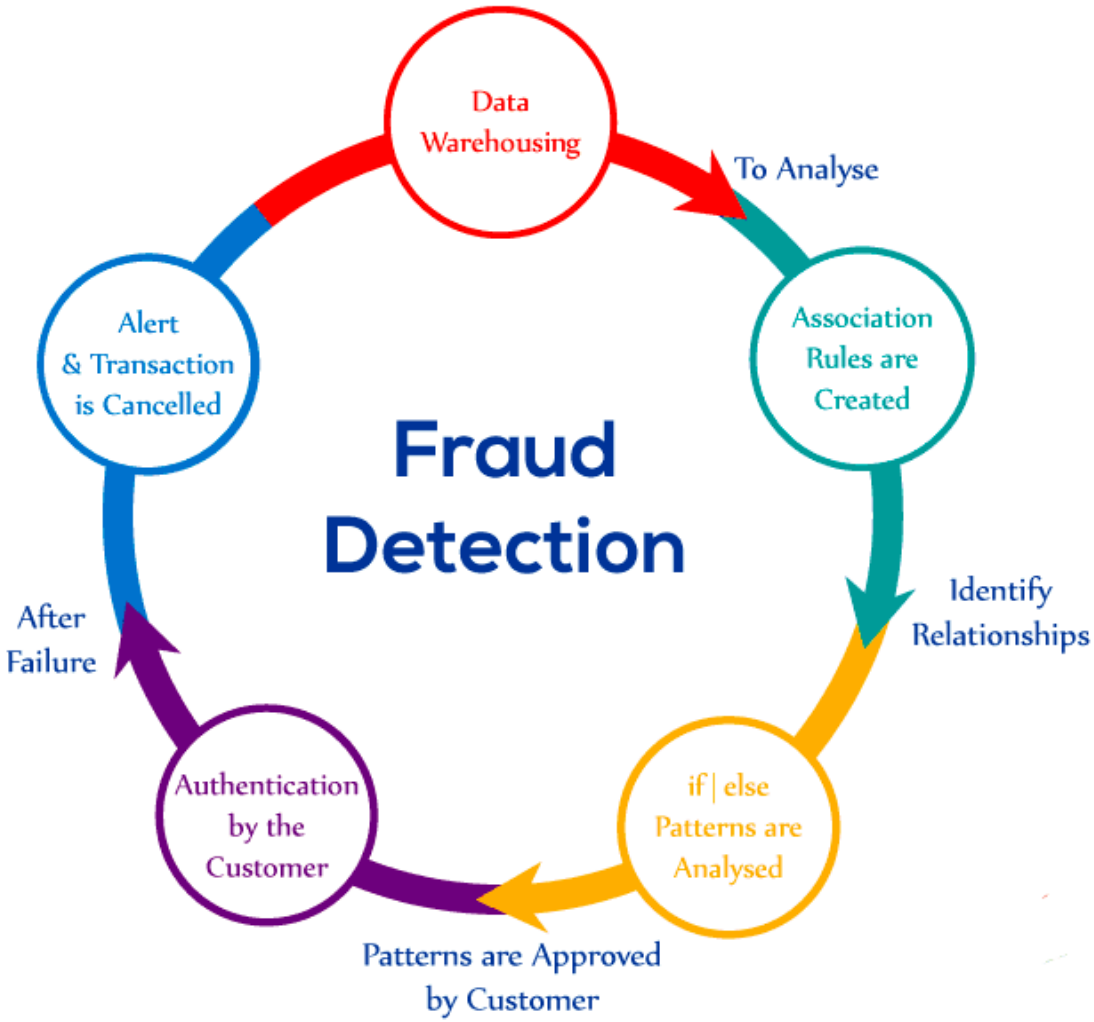
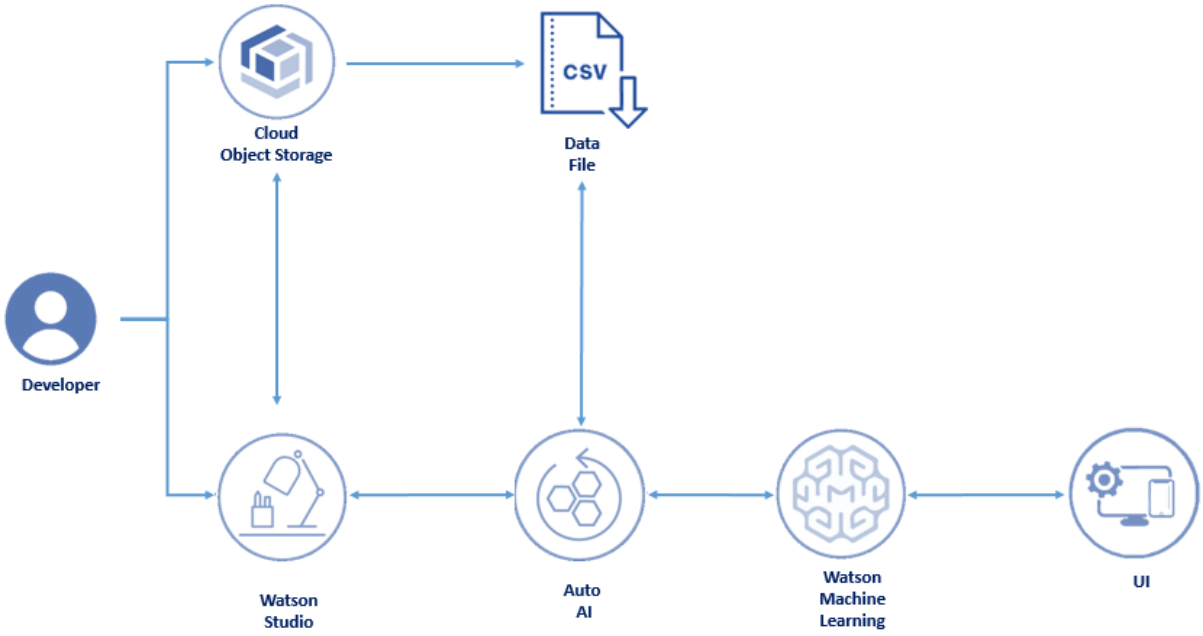


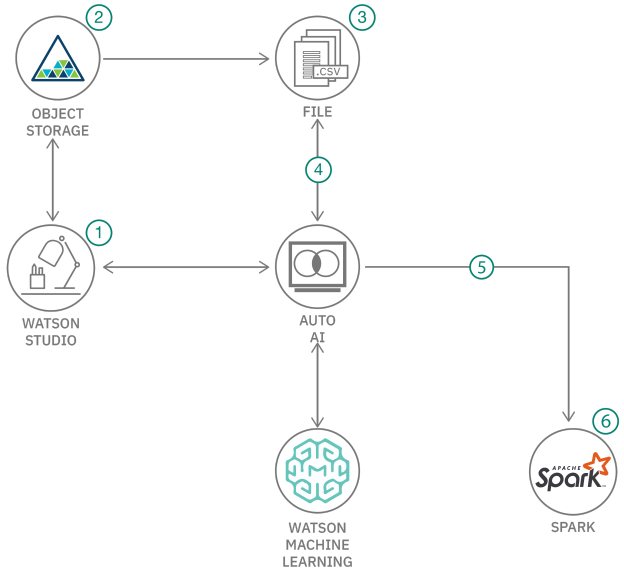
Figure 5

4. FLOWCHART


Historical Transactions
Online Transaction
CUSTOMER
DATA BASE
Customer ID
Classification Type
Customer ID
Transaction Attributes
Fraud Rules
Engine
Rules Engine
RULES
Fraud Rules Set
K-means classifier
Level 1
Classification
Transaction's
critical values
Rule Engine
Classification
ABC algorithm
Best Solutions
KNN algorithm
Level 2
Classification
Training
Testing
CREDIT CARD
FRAUD
NORMAL
TRANSACTION


Data Warehousing
To Analyse
Association Rules are Created
Identify Relationships
if | else Patterns are Analysed
Patterns are Approved by Customer
Authentication by the Customer
After Failure
Alert & Transaction is Cancelled
Fraud Detection





- Quickly set up the model building services on IBM Cloud
- Ingest the data and initiate the AutoAI tool
- Build different models using AutoAI and evaluate the performance
- Choose the best model and complete the deployment

- Generate predictions using the deployed model by making REST calls
- Compare the process of using AutoAI and building the model manually

1. Log in to Watson Studio, create a project, and initiate an instance of AutoAI and Cloud Object Storage.
2. Upload the .csv data file to Object Storage.
3. Initiate the model building process using AutoAI and create pipelines.
4. Evaluate different pipelines from AutoAI and select the best model for deployment.
5. Generate accurate predictions by making REST calls to the deployed model.

1. Create an account with IBM Cloud.
2. Create a new Watson Studio project.
3. Add data to the project.
4. Add an asset as AutoAI.
5. Create and define an experiment.
6. Import the .csv file.
7. Run the experiment.
8. Analyze the results.
9. Deploy to IBM Cloud.
10. Test the model.

5. EXPERIMENTAL INVESTIGATIONS

The methodology used in this project is fairly straightforward and consists of the following steps:

- First, we follow a “standard” data science approach. We apply several combinations of feature engineering techniques with different degrees of complexity: from basic *NaN* imputation and encoding to domain- and dataset-specific transformations, as well as several machine learning models.

- Second, we load the dataset into AutoAI and let it do its magic. Then, AutoAI will come up with eight pipelines consisting of different combinations of feature transformations and hyperparameters for the model that best fits the data. We can then choose whichever pipeline we prefer (AutoAI will give us metrics to compare them) and predict the targets of the dataset, whose real values we can’t see. There are several ways of doing this: we can save the pipeline as a model on IBM Cloud and then input it the test set or generate a Jupyter notebook and produce the predictions via code — we use the first option.

- Third, we proceed to mix the two previous steps. We take the feature engineering and transformations from the first step and we apply the machine learning model form the AutoAI pipeline. If we want to make sure that the model does not overfit the data, we can apply early stopping when we train it. After training the model on this data, we are again ready to generate and submit our predictions. While the optimization problem that yielded the pipeline did not include these transformations (but rather used others), we have seen that in both cases this led to better results.

The aim of this project is to identify fraud in credit card transactions. Consequently, the

problem was framed as a binary classification problem (whether or not the transaction was fraudulent). With a training set including the target, each observation represents one single transaction. The test set has a similar size (though it obviously doesn't include the target variable). The dataset is highly imbalanced, with only 3% of entries labeled as fraud.

Given the imbalance of the target variable, we apply random oversampling on the dataset.

Once we have our the dataset, we proceed to train several classification algorithms. These include a Logistic Regression, a Support Vector Machine, a Random Forest, an XGBoost, and a Light GBM. We then predict the target values on the test set.

In the case of AutoAI, we just need to upload the original dataset to our project in Watson Studio or Cloud Pak for Data, start a new AutoAI experiment with it, and select the target variable. When it finishes, based on the criteria and metrics that AutoAI has computed, we can select the best pipeline of the eight that have been generated and predict the target variable on the test set.

Finally, we take the Light GBM generated by AutoAI and manually train it on the dataset that resulted from our manual feature engineering process. We also apply early stopping during training.

6. RESULT

As we can see, the manual process can yield very disperse and diverse results. It is the source of some really poor models (ROC-AUC of 0.5) and some really good ones (0.929512). AutoAI's best pipeline already performs quite well, with a score of 0.900256, but it is still far from the top manual algorithm in a competition where each decimal point matters.

As mentioned above, we achieve the best results when we combine the feature engineering based on domain and dataset knowledge and the model generated by AutoAI. In fact, the **AutoAI model applied to the dataset from the manual feature engineering process outscores the AutoAI submission and all manual models except for the top** one, with a public score of 0.935207. While higher scores were achieved in the competition, these results already point in one direction: **when used in conjunction with domain knowledge, AutoAI provides a much better starting point than when a team of data scientists has to build a model from scratch**. Furthermore, the AutoAI submission matches (and even outscores) some of the top manual algorithms. This means that **even analysts with little to no experience in Data Science can already get high-quality results by pressing a button**.

7. ADVANTAGES & DISADVANTAGES

7.1. ADVANTAGES

- An end-to-end data science solution to scale analysis across your entire organization. Ability to speed modeling, training and deployment time while simplifying collaboration and adhering to a strong governance and security posture.

- A platform that supports visual programming in order to upskill your team. More data science for more people, faster discovery and deployment, and deep learning and advanced analytics allow you to move from detection to prediction.

- Make faster, more accurate decisions. Leverage unstructured data and enable deep learning and neural networks to reduce false positives. Use pre-trained API's and one-click tooling to train and develop models faster.

- The IBM Watson® Studio platform provides an end-to-end data science solution that quickly puts AI to work, helping your organization stay ahead of fraudsters. As a single platform for collaborative model development, Watson Studio makes it possible for cross-functional business and technical teams to work together quickly and seamlessly.
- Watson Studio enables different sets of users to work together regardless of the

types of users they are. It supports teams with easy tooling to help automate tasks and with more advanced tools such as deep learning and neural networks.

- **Quickly put AI to work**

Watson Studio provides a true business environment on a single platform for users to develop, train and manage models that support fraud detection and prediction.

- **Enable more employees**

Tools such as AutoAI, data refinery and low-code visual modeling help automate tasks for data scientists, and simplify modeling for business users.

- **Stay ahead of fraudsters**

Detect and prevent fraud with a platform that supports advanced tools such as deep learning, neural network frameworks and models in a security-rich, governed environment.

- **Get deeper customer insights**

Easy tooling and access to more types of data — including unstructured data — enables more in-depth investigation and prediction.

- **Boost your ROI**

An open source, flexible platform lowers implementation and ownership costs to avoid vendor lock-in and optimize existing investments.

- **Speed:** Machine Learning is widely used because of its fast computation. It analyzes and processes data and extracts new patterns from it within no time. For human beings to evaluate the data, it will take a lot of time and evaluation time will increase with the amount of data.
- **Scalability:** As more and more data is fed into the Machine Learning-based model, the model becomes more accurate and effective in prediction.
- **Efficiency:** Machine Learning algorithms perform the redundant task of data analysis and try to find hidden patterns repetitively. Their efficiency is better in giving results in comparison with manual efforts. It avoids the occurrence of false positives which counts for its efficiency.

7.2. DISADVANTAGES

- Difficulty to confirm the structure/high processing time for large neural networks and excessive training/ poor explanation capability/ difficult to setup and operate/high expense/ non numerical data need to be converted and normalized/Sensitivity to data format.
- Requires extensive tool knowledge to set up and operate and difficult to understand.
- Highly expensive/ low accuracy/not scalable to large size data sets.
- Poor in process large dataset/expensive/has low speed of detection/ medium accuracy/lack of transparency of results.
- Excessive training need/ expensive.
- Poor in handling missing information or unexpected data values/poor in process different data types /knowledge representation languages do not approach human flexibility/ poor in build and operate/ poor in integration.
- Has low predictive accuracy/extremely sensitive to noise/ their performance deteriorates rapidly in the presence of spurious data.
- May suffer from the problem of incomplete or noisy data.
- Requirements to check each condition one by one. In fraud detection condition is transaction.

1. Enormous Data is processed every day and the model build must be fast enough to respond to the scam in time.
2. Imbalanced Data i.e most of the transactions *(99.8%)* are not fraudulent which makes it really hard for detecting the fraudulent ones
3. Data availability as the data is mostly private.
4. Miss-classified Data can be another major issue, as not every fraudulent transaction is caught and reported.
5. Adaptive techniques used against the model by the scammers.

6. Ability to handle class imbalance. This is incorporated in the model by creating two separate pattern databases for fraud and legal transactions. Both customer and fraudulent behaviors are found to be changing gradually over a longer period of time. This may degrade the performance of fraud detection model. Therefore the fraud

detection model should be adaptive to these behavioral changes. These behavioral changes can be incorporated into the proposed model by updating the fraud and legal pattern databases.

Introduction to Fraud Detection Algorithms in Machine Learning

For years, fraud has been a major issue in sectors like banking, medical, insurance, and many others. Due to the increase in online transactions through different payment options, such as credit/debit cards, PhonePe, Gpay, Paytm, etc., fraudulent activities have also increased. Moreover, fraudsters or criminals have become very skilled in finding escapes so that they can loot more. Since no system is perfect and there is always a loophole them, it has become a challenging task to make a secure system for authentication and preventing customers from fraud. So, Fraud detection algorithms are very useful for preventing frauds.

Here comes Machine Learning which can be used for creating a fraud detection algorithm that helps in solving these real-world problems.

Manual Review and Transaction Rules

Nowadays, Machine Learning in Artificial Intelligence resolves most of the issues that human beings find difficult to deal with. Previously, industries were using a rule-based approach for fraud detection. But due to the popularity and acceptance of Artificial Intelligence Tutorial, especially by students and Machine Learning in every industry vertical, organizations have moved from the ruled-based fraud detection to ML-based solutions.

Now, we will look at the rule-based fraud detection system and ML-based systems.

Rule-based Approach or Traditional Approach in Fraud Detection Algorithms

In the rule-based approach, the algorithms are written by fraud analysts. They are based on strict rules. If any changes have to be made for detecting a new fraud, then they are done manually either by making those changes in the already existing algorithms or by creating new algorithms. In this approach, with the increase in the number of customers

and the data, human effort also increases. So, the rule-based approach is time-consuming and costly. Another drawback of this approach is that it is more likely to have false positives. This is an error condition where an output of a test specifies the existence of a particular condition that does not even exist. The output of a transaction depends upon the rules and guidelines made for training the algorithm for non-fraudulent transactions. So, for a fixed risk threshold, if a transaction is rejected where it should not be, it will generate a condition of high rates of false positives. This false-positive condition will result in losing genuine customers.

ML-based Fraud Detection Algorithms

In the rule-based approach, the algorithms cannot recognize the hidden patterns. Since they are based on strict rules, they cannot predict fraud by going beyond these rules. But in real world, fraudsters are very skilled and can adopt new techniques every time to commit a crime. Therefore, there is a need for a system that can analyze patterns in data and predict and respond to new situations for which it is not trained or explicitly programmed.

Hence, we use Machine Learning for detecting fraud. Here, a machine tries to learn by itself and becomes better by experience. Also, it is an efficient way of detecting fraud because of its fast computing. It does not even require the guidance of a fraud analyst. It helps in reducing false positives for transactions as the patterns are detected by an automated system for streaming transactions that are in huge volume.

Now, we will look at the two most commonly used Machine Learning models for detecting fraud in transactions.

Supervised Learning Used in Fraud Detection Algorithms

Supervised Learning models are trained on tagged outputs. If a transaction occurs, it is tagged as either ‘fraud’ or ‘non-fraud.’ Large amounts of such tagged data are fed into the supervised learning model in order to train it in such a way that it gives a valid output. Also, the accuracy of the model’s output depends on how well-organized your data is.

Unsupervised Learning Used in Fraud Detection Algorithm

Unsupervised learning models are built to detect unusual behavior in transactions which is not detected previously. Unsupervised learning models involve self-learning that helps

in finding hidden patterns in transactions. In this type, the model tries to learn by itself, analyzes the available data, and tries to find the similarities and dissimilarities between the occurrences of transactions. This helps in detecting fraudulent activities.

So, both these models, supervised and unsupervised, can be used independently or in combination for detecting anomalies in transactions.

Need for the Fraud Detection Machine Learning Algorithms

Human beings always search for methods, tools, or techniques that reduce the human effort for performing a certain task efficiently. In Machine Learning, algorithms are designed in such a way that they try to learn by themselves using past experience. After learning from the past experience, the algorithms become quite capable of reacting and responding to conditions for which they are not explicitly programmed. So, Machine Learning helps a lot when it comes to fraud detection. It tries to identify hidden patterns that help in detecting a fraud which is not been previously recognized. Also, its computation is fast as compared to the traditional rule-based approaches.

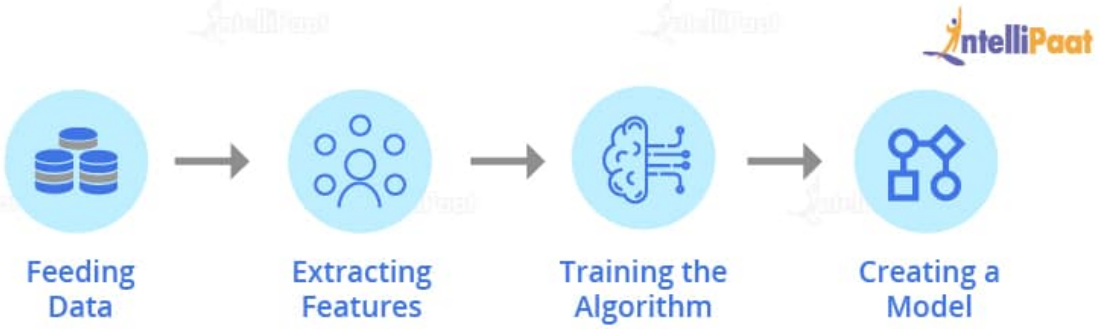
Why do we use Machine Learning in Fraud Detection?

Here are some factors for why Machine Learning techniques are so popular and widely used in industries for detecting frauds:

- Speed: Machine Learning is widely used because of its fast computation. It analyzes and processes data and extracts new patterns from it within no time. For human beings to evaluate the data, it will take a lot of time and evaluation time will increase with the amount of data.
- Scalability: As more and more data is fed into the Machine Learning-based model, the model becomes more accurate and effective in prediction.
- Efficiency: Machine Learning algorithms perform the redundant task of data analysis and try to find hidden patterns repetitively. Their efficiency is better in giving results in comparison with manual efforts. It avoids the occurrence of false positives which counts for its efficiency.

How does a Machine Learning system work for Fraud Detection?

The below picture shows the basic structure of the working of fraud detection algorithms using Machine Learning:



Feeding Data: First, the data is fed into the model. The accuracy of the model depends on the amount of data on which it is trained, more data better the model performs.

For detecting frauds specific to a particular business, you need to input more and more amounts of data into your model. This will train your model in such a way that it detects fraud activities specific to your business perfectly.

Extracting Features: Feature extraction basically works on extracting the information of each and every thread associated with a transaction process. These can be the location from where the transaction is made, the identity of the customer, the mode of payments, and the network used for transaction.

- Identity: This parameter is used to check a customer's email address, mobile number, etc. and it can check the credit score of the bank account if the customer applies for a loan.
- Location: It checks the IP address of the customer and the fraud rates at the customer's IP address and shipping address.
- Mode of Payment: It checks the cards used for the transaction, the name of the cardholder, cards from different countries, and the rates of fraud of the bank account used.
- Network: It checks for the number of mobile numbers and emails used within a network for the transaction.

Training the Algorithm: Once you have created a fraud detection algorithm, you need to train it by providing customers data so that the fraud detection algorithm learns how to distinguish between 'fraud' and 'genuine' transactions.

Creating a Model: Once you have trained your fraud detection algorithm on a specific dataset, you are ready with a model that works for detecting 'fraudulent' and

'non-fraudulent' transactions in your business.

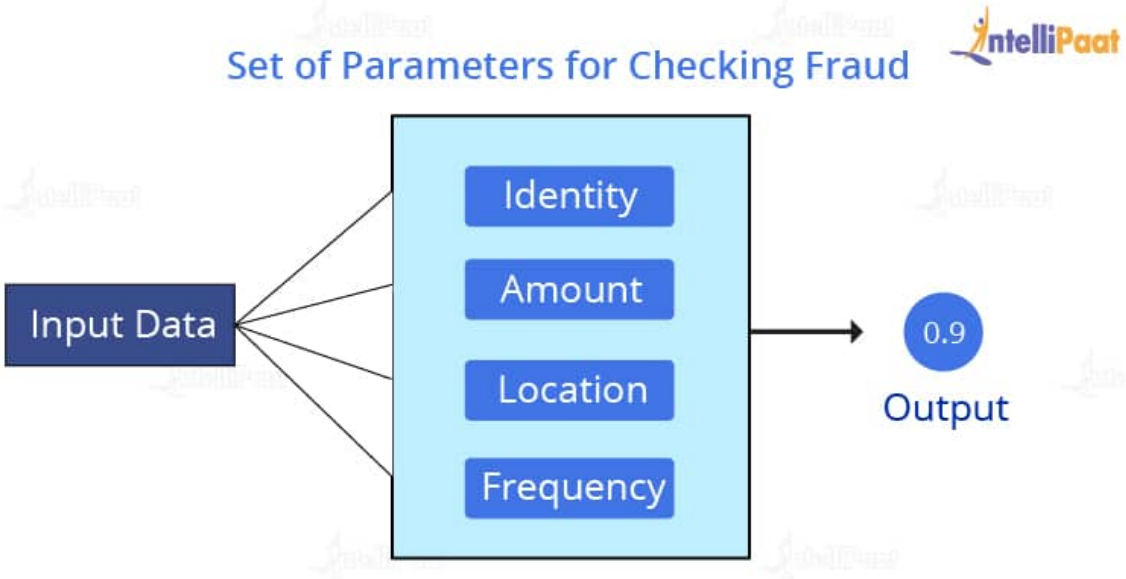
The advantage of Machine Learning in fraud detection algorithms is that it keeps on improving as it is exposed to more data.

There are many techniques in Machine Learning used for fraud detection. Here, with the help of some use cases, we will understand how Machine Learning is used in fraud detection.

Techniques of Machine Learning for Fraud Detection Algorithms

1. Fraud Detection Machine Learning Algorithms Using Logistic Regression: Logistic Regression is a supervised learning technique that is used when the decision is categorical. It means that the result will be either 'fraud' or 'non-fraud' if a transaction occurs.

Use Case: Let us consider a scenario where a transaction occurs and we need to check whether it is a 'fraudulent' or 'non-fraudulent' transaction. There will be given set of parameters that are checked and, on the basis of the probability calculated, we will get the output as 'fraud' or 'non-fraud.'

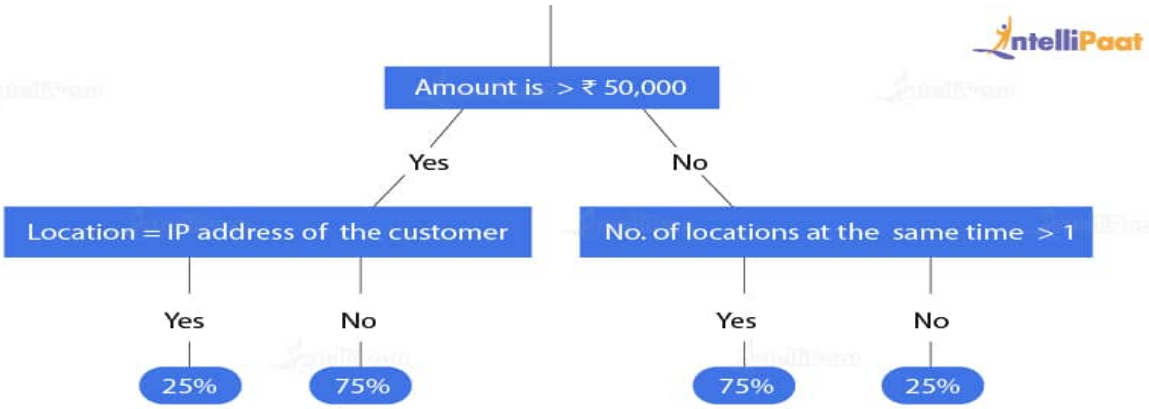


In the above diagram, we can see that the probability calculated is 0.9. This means that there is a 90 percent chance that the transaction is 'genuine' and there is a 10 percent probability that it is a 'fraud' transaction.

2. Fraud Detection Machine Learning Algorithms Using Decision Tree: Decision Tree algorithms in fraud detection are used where there is a need for the classification

of unusual activities in a transaction from an authorized user. These algorithms consist of constraints that are trained on the dataset for classifying fraud transactions.

Use Case: Let us consider a scenario where a user makes transactions. We will build a decision tree to predict the probability of fraud based on the transaction made.



First, in the decision tree, we will check whether the transaction is greater than ₹50,000. If it is 'yes,' then we will check the location where the transaction is made.

And if it is 'no,' then we will check the frequency of the transaction.

After that, as per the probabilities calculated for these conditions, we will predict the transaction as 'fraud' or 'non-fraud.'

Here, if the amount is greater than ₹50,000 and location is equal to the IP address of the customer, then there is only a 25 percent chance of 'fraud' and a 75 percent chance of 'non-fraud.'

Similarly, if the amount is greater than ₹50,000 and the number of locations is greater than 1, then there is a 75 percent chance of 'fraud' and a 25 percent chance of 'non-fraud.'

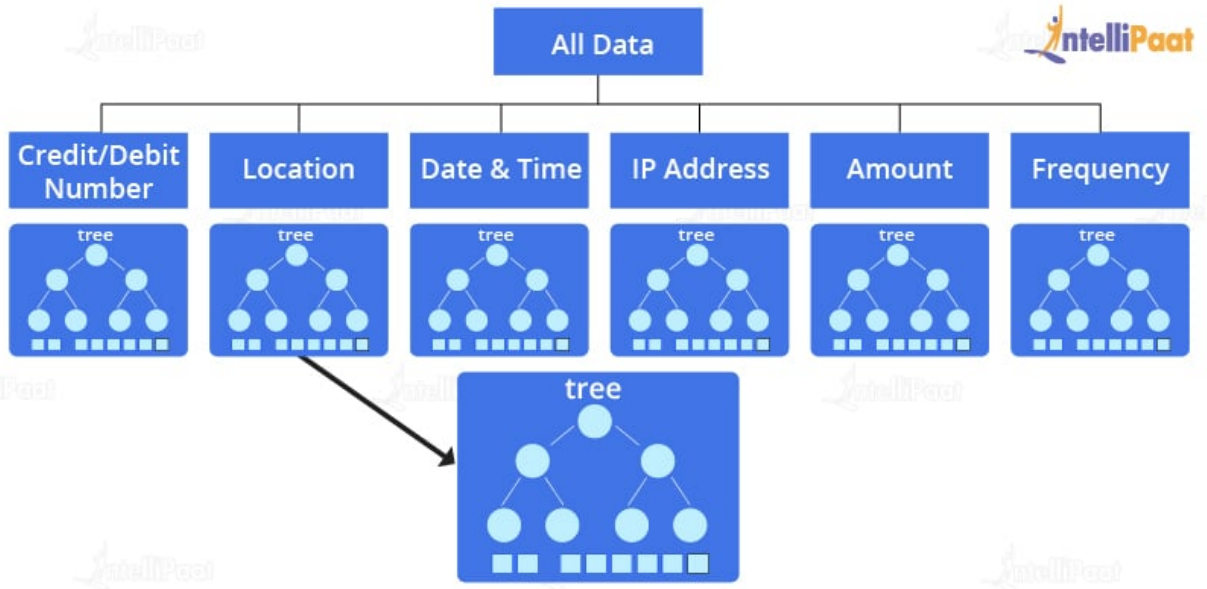
This is how a decision tree in Machine Learning helps in creating fraud detection algorithms.

Now, we will look at the random forest in Machine Learning used in fraud detection algorithms.

3. Fraud Detection Machine Learning Algorithms Using Random Forest: Random Forest uses a combination of decision trees to improve the results. Each decision

tree checks for different conditions. They are trained on random datasets and, based on the training of the decision trees, each tree gives the probability of the transaction being 'fraud' and 'non-fraud.' Then, the model predicts the result accordingly.

Use Case: Let's consider a scenario where a transaction is made. Now, we will see how the random forest in Machine Learning is used in fraud detection algorithms.



When a request for a transaction is given to the model, it checks for the information like the credit/debit card number, location, date, time, the IP address, the amount, and the frequency of the transaction. All this dataset is fed as an input into the fraud detection algorithm. Then this fraud detection algorithm selects variables from the given dataset that help in splitting up of the dataset. The below diagram shows the splitting up of the dataset into multiple decision trees.

So, the sub-trees consist of variables and the conditions to check those variables for an authorized transaction.

After checking all the conditions, all the sub-trees will give the probabilities for a transaction to be 'fraud' and 'non-fraud.' Based on the combined result, the model will mark the transaction as 'fraud' or 'genuine.'

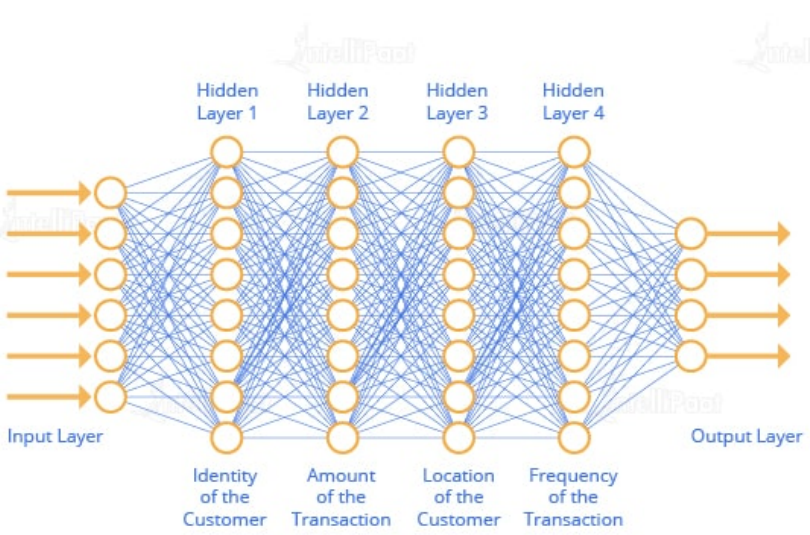
This is how a random forest in Machine Learning is used in fraud detection algorithms.

4. Fraud Detection Machine Learning Algorithms Using Neural Networks: Neural Networks is a concept inspired by the working of a human brain. Neural networks

in Deep Learning uses different layers for computation. It uses cognitive computing that helps in building machines capable of using self-learning algorithms that involve the use of data mining, pattern recognition, and natural language processing. It is trained on a dataset passing it through different layers several times.

It gives more accurate results than other models as it uses cognitive computing and it learns from the patterns of authorized behavior and thus distinguishes between 'fraud' and 'genuine' transactions.

Use Case: Now, we will look at an example where a neural network is used for fraud detection. There are different layers in a neural network that focus on different parameters to make a decision whether a transaction is 'fraud' or 'non-fraud.' In the below diagram it is shown how the layers of neural networks represent and work on different parameters.



First, the data is fed into the neural network. After that, the Hidden Layer 1 checks the amount of transaction, and similarly other layers check for the location, identity, IP address of the location, the frequency of transaction, and the mode of payment. There can be more business-specific parameters. These individual layers work on these parameters, and computation is done based on the models' self-learning and past experience to calculate the probabilities for detecting frauds.

Thus, neural networks work on data and learn from it, and it improves the model's performance over every iteration.

This is how neural networks are used for implementing fraud detection algorithms.

In this blog, we have seen how fraud detection algorithms work using Machine Learning techniques such as logistic regression, decision tree, random forest, and neural networks. This technology is improving day by day so that it provides us more accuracy and better results to prevent fraud.

8. APPLICATIONS

- The proposed fraud detection method takes very less time, which is also an important parameter of this real time application.
- Data mining can also be used to detect fraudulent credit card transactions, predict which customers are more likely to default their contractual obligations by going bankrupt as well as identify fraudulent credit applications.

1. The model used must be simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible.
2. Imbalance can be dealt with by properly using some methods which we will talk about in the next paragraph
3. For protecting the privacy of the user the dimensionality of the data can be reduced.
4. A more trustworthy source must be taken which double-check the data, at least for training the model.
5. We can make the model simple and interpretable so that when the scammer adapts to it with just some tweaks we can have a new model up and running to deploy.

- Ease of purchase Credit cards can make life easier.
- They allow customers to purchase on credit in arbitrary time, location and amount, without carrying the cash.
- Provide a convenient payment method for purchases made on the internet, over the telephone, through ATMs, etc.
- Keep customer credit history Having a good credit history is often important in detecting loyal customers.
- This history is valuable not only for credit cards, but also for other financial services like loans, rental applications, or even some jobs.
- Lenders and issuers of credit mortgage companies, credit card companies, retail

stores, and utility companies can review customer credit score and history to see how punctual and responsible customers are in paying back their debts.

- Protection of Purchases Credit cards may also offer customers, additional protection if the purchased merchandise becomes lost, damaged, or stolen.
- Both the buyer‟s credit card statement and company can confirm that the customer has bought if the original receipt is lost or stolen.
- In addition, some credit card companies provide insurance for large purchases.

9. CONCLUSION

The conclusions are hence similar to the ones we got from the Fraud Detection competition: **AutoAI provides a great starting point for analysts with little to no experience in Data Science and an even more powerful tool for data scientists with domain specific knowledge and skills**.

AutoAI is IBM's top-of-the-class solution in the emerging space of Automated Machine Learning, against the standard "manual" approach of Data Science teams in two real-world classification problems. In both cases, AutoAI quickly proves to be a powerful tool for an array of profiles with diverse skills:

- Analysts with little to no experience in Data Science can benefit from high-quality models ready to go in production with just one click of a button.

- Consolidated teams of data scientists can leverage their domain knowledge and expertise and start iterating over models based on the optimized results of AutoAI. This is where AutoAI can reach its best results.

This process is used to detect the credit card transaction, which are fraudulent or genuine. Data mining techniques of Predictive modeling, Decision trees and Logistic Regression are used to predict the fraudulent or genuine credit card transaction. In predictive modeling to detect and check output class distribution. The prediction model predicts continuous valued functions. We have to detect 148 may be fraud and other are genuine. In decision tree generate a tree with root node, decision node and leaf nodes. The leaf node may be 1 becomes fraud and 0 otherwise. Logistic Regression is same as linear regression but interpret curve is different. To generalize the linear regression model, when dependant variable is categorical and analyzes relationship between

multiple independent variables.

if the transaction is fraudulent, the system will record this transaction as a fraud in the database and will then reject it. Next, the acquiring bank sends a SMS alert to the real consumer that the transaction has not been processed, because the system suspects the transaction as fraudulent.

Nowadays, in the global computing environment, online payments are important, because online payments use only the credential information from the credit card to fulfill an application and then deduct money. Due to this reason, it is important to find the best solution to detect the maximum number of frauds in online systems.

Fraud detection is a complex issue that requires a substantial amount of planning before throwing machine learning algorithms at it. Nonetheless, it is also an application of data science and machine learning for the good, which makes sure that the customer's money is safe and not easily tampered with.

Detection of credit card fraud is an intent part of testing for the researchers over a long time and will be an interesting part of testing in the coming time. We are introducing a fraud detection system for credit-cards by applying three different algorithms and training our machine using these algorithms with the transaction records we have. The model that we built helps the authorities to get notified of the fraud in credit-cards and take the further necessary steps over the transaction and label the transaction as fraud or legitimate transaction. These algorithms show us that the given transaction tends to be a type of fraud or not, these algorithms were selected using experimentation, discussion and feature importance techniques as shown in methodology. It is a real-time transaction data from European credit-card holders which explains the skewness of data. Therefore, we can infer that there is a requirement of applying feature selection technique. We used a PCA algorithm to select the features from our transaction dataset which uses correlation and variance as parameters to select the features. We have set the summation of variance as 95% for selecting the features using the PCA algorithm. We also applied the PCA feature selection algorithm as there are no features which have high variance and correlation with the class column which determines the transaction as fraud or not. We have applied the three machine learning algorithms as stated in methodology and the models indicate a high accuracy score for each one of them. The scores of each model were 99.7%, 99.8%, 99.7% for the decision tree, support vector machine and random forest classifier algorithms respectively. As these models have high accuracy but the predicted values have a low precision, so with

the upcoming time we will be focusing on improvement in our model and get the best results with high precision in determining the fraud detections in credit card transaction records.

The result of the intended models in overall performance was superior. Overall results show that stacking classifier using Training Model, Meta classifier is most promising to predict fraud transaction in the dataset, followed by the classifier for data analysis. The paper focuses on this aspect of the Fraud Detection System and proposes a method for designing, learning, and upgrading the Datasets to boost the performance of fraud detection. Transactions are invariably related to feature vectors that have either received an outsized fraud score or a high chance of being a fraud generate alert. More number of alerted transactions are reported to the investigators, which represent the final layer of control. The assessment of the learning model is based on its accuracy recall, precision, specificity. The outcome of all the intended models in the overall performance was superior.

Although credit card frauds are found in large number in some major economies like the UK, Malaysia, Japan, Taiwan, Australia, and Hongkong as compared to India, because credit card industry in India is still in its nascent stage and cards have low credit limits. Of all the credit cards issued in India, 80-85% cards are active which is equal to 3-4 times smaller than Malaysia, which is on credit card fraud list.Although, the incidences of fraud in India are less, the RBI has advised the banks to establish the internal control system to check the credit card frauds within certain limits, and to strengthen their appraisal system.

This method proves accurate in finding out the fraudulent transactions and minimizing the number of false alert. Genetic Algorithm is appropriate in such kind of application areas. The use of this algorithm in credit card fraud detection system results in detecting or predicting the fraud probably in a very short span of time after the transactions has been made. This will eventually prevent the banks and customers from great losses and also will reduce risks.

Credit card fraud has become more and more rampant in recent years. To improve merchants' risk management level in an automatic and efficient way and building an accurate and easy handling credit card risk monitoring system is one of the key tasks for the merchant banks. One aim of this study is to identify the user model that best identifies fraud cases. There are many ways of detection of credit card fraud. If one of these or combination of algorithm is applied into bank credit card fraud detection

system, Then the probability of fraud transactions can be predicted soon after credit card transactions by the banks. This paper gives contribution towards the effective ways of credit card fraudulent detection. In our paper we survey on seven existing Techniques for credit card fraud detection with comparing their results hence we conclude that out of these method HMM model is one of the best model because in HMM model fraud detect using Card holders spending behavior, but we need to improvement HMM in future.

As card business transactions increase, so too do frauds. Clearly, global networking presents as many new opportunities for criminals as it does for businesses. While offering numerous advantages and opening up new channels for transaction business, the internet has also brought in increased probability of fraud in credit card transactions. The good news is that technology for preventing credit card frauds is also improving many folds with passage of time. Reducing cost of computing is helping in introducing complex systems, which can analyse a fraudulent transaction in a matter of fraction of a second. It is equally important to identify the right segment of transactions, which should be subject to review, as every transaction does not have the same amount of risk associated with it. Finding the optimally balanced ‘total cost of fraud’ and other measures outlined in this article can assist acquiring and issuing banks in combating frauds more efficiently.

10. FUTURE SCOPE

This paper has reviewed various machine learning algorithm detect fraud in credit card transaction. The performances of all this techniques are examined based on accuracy, precision and specificity metrics. We have selected supervised learning technique Random Forest to classify the alert as fraudulent or authorized. This classifier will be trained using feedback and delayed supervised sample. Next it will aggregate each probability to detect alerts. Further we proposed learning to rank approach where alert will be ranked based on priority. The suggested method will be able to solve the class imbalance and concept drift problem. Future work will include applying semi-supervised learning methods for classification of alert in FDS.

While we couldnt reach out goal of 100% accuracy in fraud detection, we did end up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here.

The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result.

This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project.

More room for improvement can be found in the dataset. As demonstrated before, the precision of the algorithms increases when the size of dataset is increased. Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves.

Future work will include a comprehensive tuning of the Random Forest algorithm I talked about earlier. Having a data set with non-anonymized features would make this particularly interesting as outputting the feature importance would enable one to see

what specific factors are most important for detecting fraudulent transactions.

We need to also make our systems learn from the past committed frauds and make them capable of adapting to future new methods of frauds.

There can be more accurate classification algorithms, data training and testing algorithms can be used to get better result.

In future, this technique will be standardized across all card associations and banks. Nonetheless, this approach is difficult because of customer's privacy concerns for customer data. Consequently, the challenge the credit companies must master is implementing such a system without spooking the customer. A second challenge is the timeliness of the detection. Customers want their transactions approved in seconds, not minutes. To address this issue, better machine learning algorithms are needed to raise flags about fraudulent transaction in real-time. Standardized techniques are desirable across industries, however they must account for user heterogeneity and security preferences, and models have to bed constantly update in order to detect and learn emerging fraudulent behaviors.

The number of cashless transactions is at its peak point since the beginning of the digital era and it is most likely to increase in the future.

Future work can be performed to improve real-time scenarios combined with sufficient feature engineering and state-of-the-art machine learning methods.

For future work, the methods studied in this paper will be extended to online learning models. In addition, other online learning models will be investigated. The use of online learning will enable rapid detection of fraud cases, potentially in real-time. This in turn will help detect and prevent fraudulent transactions before they take place, which will reduce the number of losses incurred every day in the financial sector.

After evaluation, it is clearly shown the various methods which can detect the Fraud efficiently and provide accurate secuirity. Speed of the software can be enhanced by implementation of algorithms of less complexity.Inter mail server can be implemented using the same concept. Proper security provisions are made from malicious threats and hacking tools so that user account cannot be harmed intentionally or non-intentionally from frauds. Proper hierarchy of the users is maintained as per authority to access the data and use the services provided by the authority. Track all the necessary details during transaction process.

11. BIBILIOGRAPHY

- https://www.ibm.com/in-en/analytics/fraud-prediction
- https://github.com/IBM/predict-fraud-using-auto-ai
- https://www.ibmbigdatahub.com/tag/2369
- https://medium.com/ibm-garage/ibms-autoai-at-work-890359910a43
- https://iraangeles-ibm.github.io/SCDF-INNOVATION-CHALLENGE/Workshop_Watsion_Studio/AutoAI.html
- https://netrixllc.com/blog/credit-card-fraud-detection-using-machine-learning/
- https://intellipaat.com/blog/fraud-detection-machine-learning-algorithms/
- https://www.fico.com/blogs/5-keys-using-ai-and-machine-learning-fraud-detection
- https://towardsdatascience.com/detecting-credit-card-fraud-using-machine-learning-a3d83423d3b8
- https://developer.ibm.com/patterns/fraud-prediction-using-autoai/
- https://www.ijert.org/credit-card-fraud-detection-using-machine-learning-and-data-science
- https://www.3pillarglobal.com/insights/credit-card-fraud-detection
- https://datasciencecmu.wordpress.com/2014/04/18/the-future-of-fraud-detection-2/
- https://www.sciencedirect.com/science/article/pii/S1877050918309347/pdf?md5=875111b444c7a5907c9acc37161a1fc8&pid=1-s2.0-S1877050918309347-main.pdf
- https://www.researchgate.net/publication/332264296_A_comparative_analysis_of_various_credit_card_fraud_detection_techniques
- https://www.ijrte.org/wp-content/uploads/papers/v7i5s2/ES2073017519.pdf
- https://www.ukessays.com/essays/information-technology/credit-card-fraud-detection-using-hidden-markov-information-technology-essay.php
- http://dspace.lpu.in:8080/jspui/bitstream/123456789/2814/1/11604887_11_28_

2017%2012_50_04%20PM_Full%20report.pdf

- http://www.iosrjournals.org/iosr-jce/papers/Vol21-issue3/Series-5/H2103054552.pdf
- http://ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502246.pdf
- https://www.hindawi.com/journals/complexity/2018/5764370/
- https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/
- http://www.ijesrt.com/issues%20pdf%20file/Archive-2019/March-2019/26.pdf
- https://spd.group/machine-learning/credit-card-fraud-detection/
- http://www.questjournals.org/jrhss/papers/vol8-issue2/B08020411.pdf
- https://www.researchgate.net/publication/40227011_Credit_card_fraud_and_detection_techniques_A_review
- https://www.hindawi.com/journals/tswj/2014/252797/
- https://azslide.com/credit-card-frauds-and-measures-to-detect-and-prevent-them_59ff0f671723dd2bf6816dea.html
- https://link.springer.com/chapter/10.1007/978-3-030-49161-1_1
- https://docuri.com/download/review-paper-on-credit-card-fraud-detection_59b8e6a2f581717b5b87ae67_pdf
- https://www.slideshare.net/kalpesh1908/credit-card-fraud-detection-37156933
- https://ijarcce.com/wp-content/uploads/2016/02/IJARCCE-9.pdf
- http://www.ijmer.com/papers/Vol4_Issue9/Version-4/E0409_04-2431.pdf
- https://www.capgemini.com/wp-content/uploads/2017/07/Credit_Card_Transaction_Fraud_and_Mitigation_Trends.pdf
- http://ijirt.org/master/publishedpaper/IJIRT144240_PAPER.pdf
- https://www.sciencedirect.com/science/article/pii/S1877050915007103
- https://www.sciencedirect.com/science/article/pii/S187705092030065X
- https://emerj.com/ai-sector-overviews/machine-learning-for-credit-card-fraud/
- https://ieeexplore.ieee.org/document/8123782
- https://www.researchgate.net/publication/336800562_Credit_Card_Fraud_Detect

ion_using_Machine_Learning_and_Data_Science

- https://digitalscholarship.unlv.edu/cgi/viewcontent.cgi?article=4457&context=thesesdissertations
- https://www.datacamp.com/community/news/credit-card-fraud-detection-using-machine-learning-algorithm-8dyh3fefvrb
- https://www.ripublication.com/ijaer18/ijaerv13n24_18.pdf
- http://dx.doi.org/10.5121/csit.2020.101018
- https://thesai.org/Downloads/Volume9No1/Paper_3-Credit_Card_Fraud_Detection_Using_Deep_Learning.pdf
- https://www.hindawi.com/journals/scn/2018/5483472/
- https://acadpubl.eu/hub/2018-118-21/articles/21b/90.pdf
- https://liwaiwai.com/2020/02/06/autoai-set-to-make-it-easy-to-create-machine-learning-algorithms/
- http://csitgeu.in/wp/wp-content/uploads/2020/09/mp5_2020_ibmcs.html
- https://www.slideshare.net/karansachdeva20/icp-for-data-enterprise-platform-for-ai-ml-and-data-science
- https://data-flair.training/blogs/data-science-machine-learning-project-credit-card-fraud-detection/
- https://marutitech.com/machine-learning-fraud-detection/
- http://www.iosrjournals.org/iosr-jce/papers/Vol21-issue3/Series-5/H2103054552.pdf
- https://datasciencecmu.wordpress.com/2014/04/18/the-future-of-fraud-detection-2/
- https://dalpozz.github.io/static/pdf/Dalpozzolo2015PhD.pdf
- https://ourcodeworld.com/articles/read/1150/credit-card-fraud-detection-with-ai-and-machine-learning
- https://arxiv.org/pdf/1611.06439
- https://dl.acm.org/doi/10.1145/3368756.3369082
- https://researchbank.swinburne.edu.au/file/d47f8779-d6c2-420a-b662-658c1c3

70aaa/1/2018-randhawa-credit_card_fraud.pdf

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7256395/

12. APPENDIX

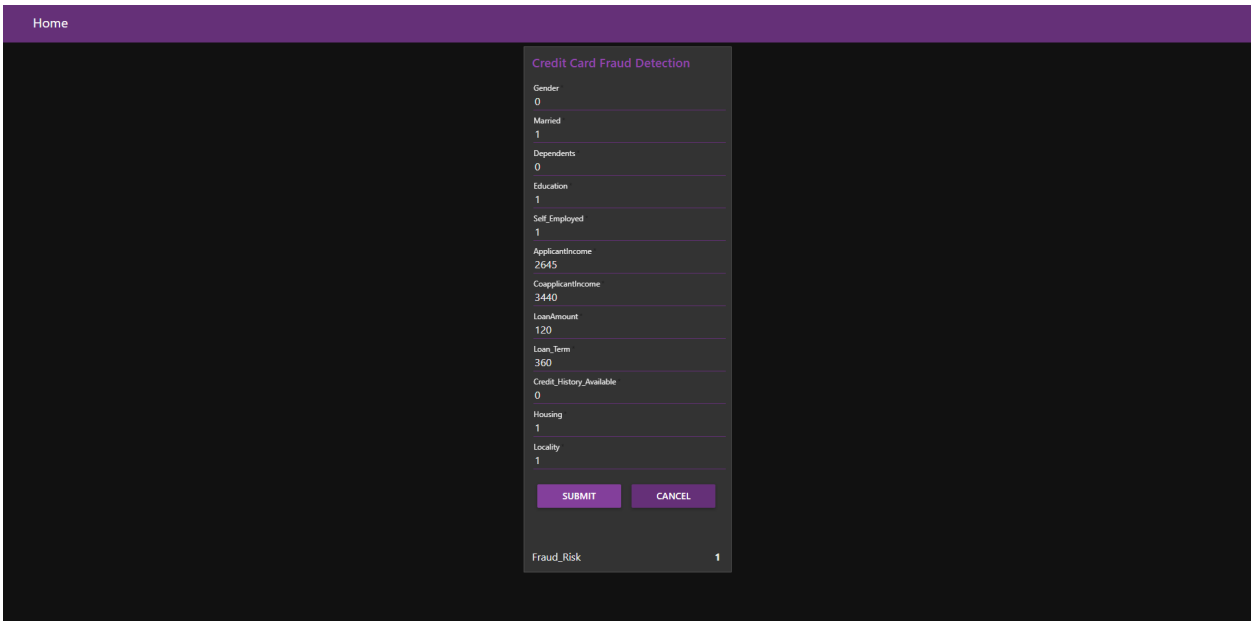
12.1. SOURCE CODE

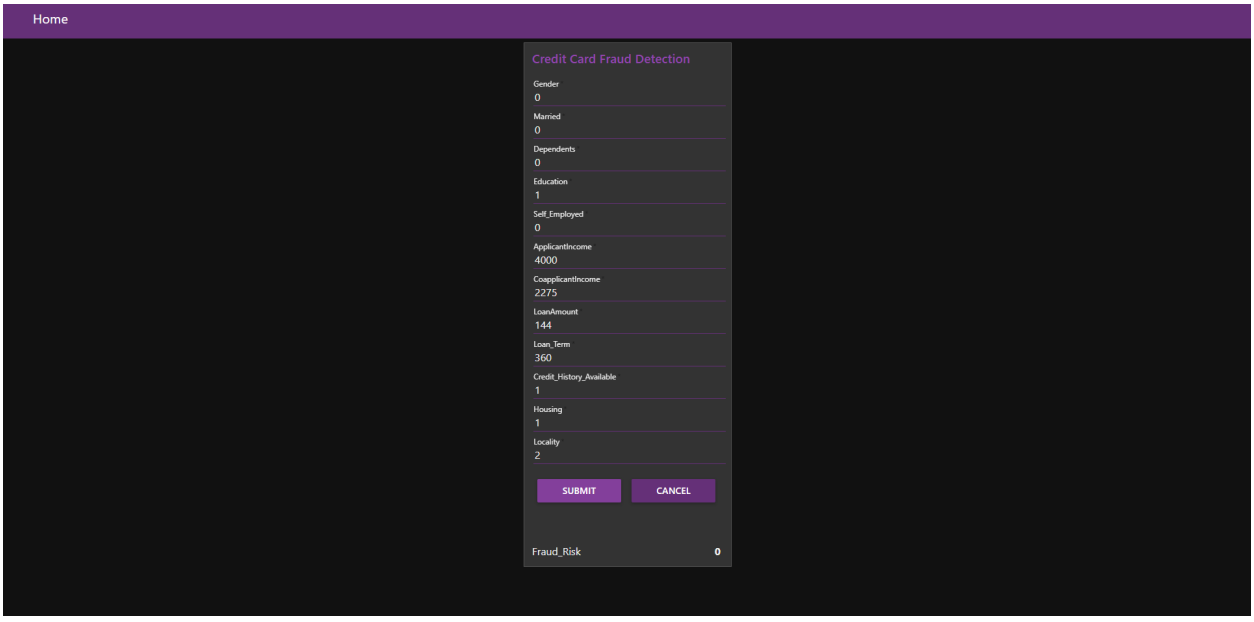
[{"id":"4df2b6d6.e88108","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"91827d30.61b85","type":"function","z":"4df2b6d6.e88108","name":"PreToken","func":"global.set(\"gr\",msg.payload.gr)\nglobal.set(\"md\",msg.payload.md)\nglobal.set(\"ds\",msg.payload.ds)\nglobal.set(\"en\",msg.payload.en)\nglobal.set(\"sd\",msg.payload.sd)\nglobal.set(\"ae\",msg.payload.ae)\nglobal.set(\"ce\",msg.payload.ce)\nglobal.set(\"lt\",msg.payload.lt)\nglobal.set(\"lm\",msg.payload.lm)\nglobal.set(\"cye\",msg.payload.cye)\nglobal.set(\"hg\",msg.payload.hg)\nglobal.set(\"ly\",msg.payload.ly)\nvar apikey=\"S9SFDtrvLzDOgbrVRRmTNWY-se63HSzkh1wkXinyY2to\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"}\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":520,"y":480,"wires":[["8e11718c.bea61"]]},{"id":"8e11718c.bea61","type":"http request","z":"4df2b6d6.e88108","name":"","method":"POST","ret":"obj","paytoqs":"ignore","url":"https://iam.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"","x":719,"y":468.9999952316284,"wires":[["42826774.9da4b8","2656e15c.17849e"]]},{"id":"c19879bc.555d68","type":"inject","z":"4df2b6d6.e88108","name":"","props":[{"p":"payload"},{"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":365.5,"y":393.99999809265137,"wires":[["91827d30.61b85"]]},{"id":"3aea70eb.8bd3e","type":"debug","z":"4df2b6d6.e88108","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":1158.0000114440918,"y":210.99999713897705,"wires":[]},{"id":"42826774.9da4b8","type":"function","z":"4df2b6d6.e88108","name":"Pre Prediction","func":"var gr = global.get('gr')\nvar md = global.get('md')\nvar ds = global.get('ds')\nvar en = global.get('en')\nvar sd = global.get('sd')\nvar ae = global.get('ae')\nvar ce = global.get('ce')\nvar lt = global.get('lt')\nvar lm = global.get('lm')\nvar cye = global.get('cye')\nvar hg = global.get('hg')\nvar ly = global.get('ly')\nvar token=msg.payload.access_token\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token}\nmsg.payload={\"input_data\": [{\"fields\": [['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',\n 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Term',\n

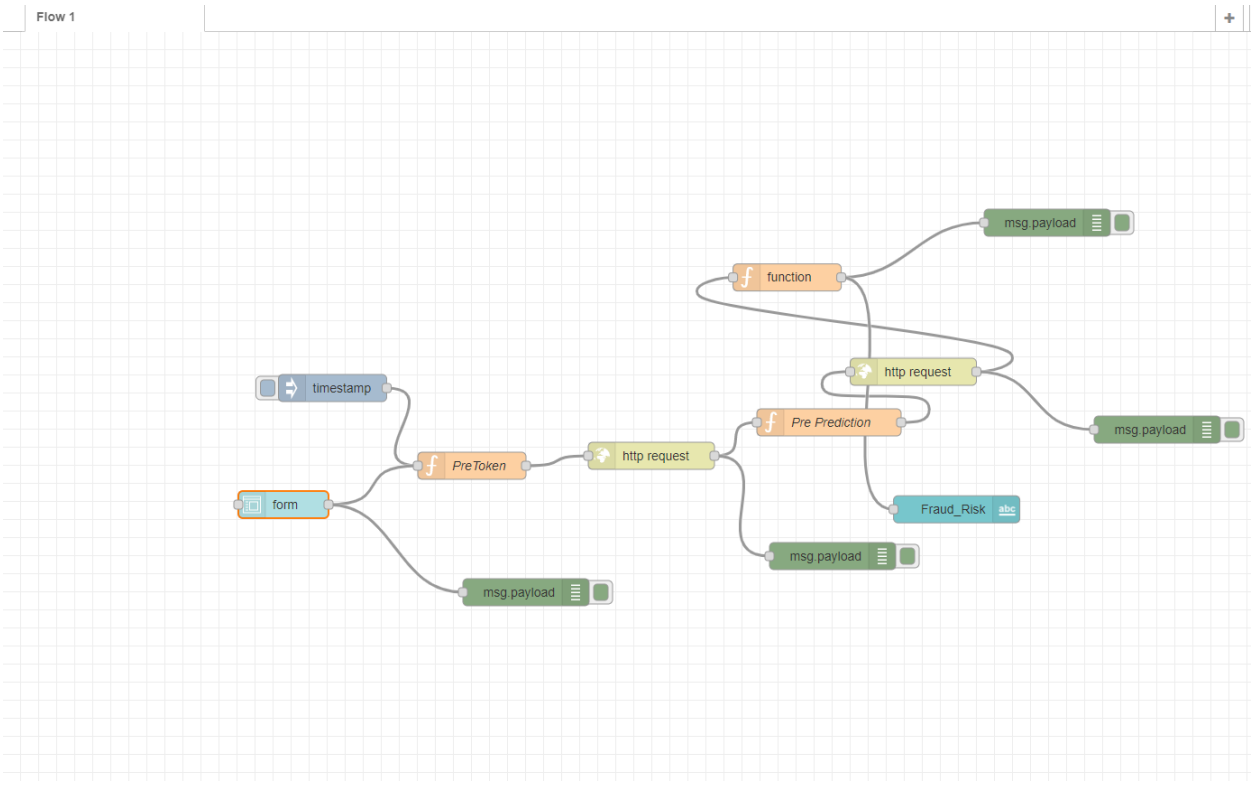
'Credit_History_Available', 'Housing', 'Locality']], \"values\":
[[gr,md,ds,en,sd,ae,ce,lt,lm,cye,hg,ly]]}]}\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":916.0000076293945,"y":431.999999
0463257,"wires":[["3d8ac48c.88539c"]]},{"id":"3d8ac48c.88539c","type":"http
request","z":"4df2b6d6.e88108","name":"","method":"POST","ret":"obj","paytoqs":"ignore","url
":"https://us-south.ml.cloud.ibm.com/ml/v4/deployments/08e5e0ae-e146-457b-a725-c
80f5cd49c46/predictions?version=2020-09-01","tls":"","persist":false,"proxy":"","authType":
"","x":1009.5000076293945,"y":375.9999990463257,"wires":[["bbdd88c7.b54178","e727f6
ea.0ed4d8"]]},{"id":"3d6d2485.8e04bc","type":"ui_form","z":"4df2b6d6.e88108","name":"","la
bel":"","group":"dfce9012.bca2","order":1,"width":0,"height":0,"options":[{"label":"Gender","val
ue":"gr","type":"number","required":true,"rows":null},{"label":"Married","value":"md","type":"nu
mber","required":true,"rows":null},{"label":"Dependents","value":"ds","type":"number","requir
ed":true,"rows":null},{"label":"Education","value":"en","type":"number","required":true,"rows":
null},{"label":"Self_Employed","value":"sd","type":"number","required":true,"rows":null},{"labe
l":"ApplicantIncome","value":"ae","type":"number","required":true,"rows":null},{"label":"Coap
plicantIncome","value":"ce","type":"number","required":true,"rows":null},{"label":"LoanAmou
nt","value":"lt","type":"number","required":true,"rows":null},{"label":"Loan_Term","value":"lm","
type":"number","required":true,"rows":null},{"label":"Credit_History_Available","value":"cye","
type":"number","required":true,"rows":null},{"label":"Housing","value":"hg","type":"number","r
equired":true,"rows":null},{"label":"Locality","value":"ly","type":"number","required":true,"rows
":null}],"formValue":{"gr":"","md":"","ds":"","en":"","sd":"","ae":"","ce":"","lt":"","lm":"","cye":"","hg":"","ly
":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":311,"y":522.999995231628
4,"wires":[["91827d30.61b85","d4aa15dd.c5e708"]]},{"id":"e5084fcb.fa572","type":"ui_text",
"z":"4df2b6d6.e88108","group":"dfce9012.bca2","order":2,"width":0,"height":0,"name":"","lab
el":"Fraud_Risk","format":"{{msg.payload}}","layout":"row-spread","x":1057.600036621093
8,"y":528.2000074386597,"wires":[]},{"id":"bbdd88c7.b54178","type":"function","z":"4df2b6
d6.e88108","name":"","func":"msg.payload=msg.payload.predictions[0].values[0][0]\nretu
rn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":869.6000366210938,"y":271.600001
33514404,"wires":[["3aea70eb.8bd3e","e5084fcb.fa572"]]},{"id":"d4aa15dd.c5e708","type":
"debug","z":"4df2b6d6.e88108","name":"","active":true,"tosidebar":true,"console":false,"tost
atus":false,"complete":"false","statusVal":"","statusType":"auto","x":580,"y":620,"wires":[]},{"id
":"2656e15c.17849e","type":"debug","z":"4df2b6d6.e88108","name":"","active":true,"tosideb
ar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto",
"x":920,"y":580,"wires":[]},{"id":"e727f6ea.0ed4d8","type":"debug","z":"4df2b6d6.e88108","na
me":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statu
sVal":"","statusType":"auto","x":1280,"y":440,"wires":[]},{"id":"dfce9012.bca2","type":"ui_grou

p","z":"","name":"Credict Card Fraud Detection","tab":"d9b5735b.0d01f","order":1,"disp":true,"width":"6","collapse":false},{"id":"d9b5735b.0d01f","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]

12.2. UI OUTPUT SCREENSHOTS







Projects / Sample

Launch IDE | Add to project +

Announcement Cloud Pak for Data as a Service / IBM Satellite Beta starts November 30th. Learn more.

Data assets

0 assets selected.

Name	Type	Created by	Last modified
CSV fraud_dataset.csv	Data Asset	SIONA PANDA	Oct 07, 2020, 03:32 PM

Load | Files | Catalog

Drop files here or browse for files to upload.

AutoAI experiments

New AutoAI experiment +

Name	Status	Model type	Last modified
Classification	Completed	Binary Classification	Oct 07, 2020, 03:40 PM

Deep learning experiments

New deep learning experiment +

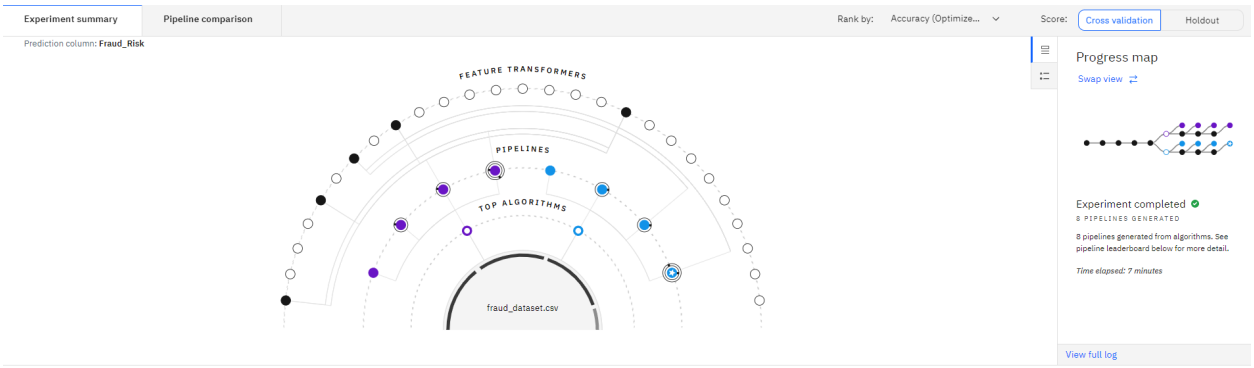
Name	Status	Last modified
You don't have any Deep learning experiments yet		

Models

Watson Machine Learning models

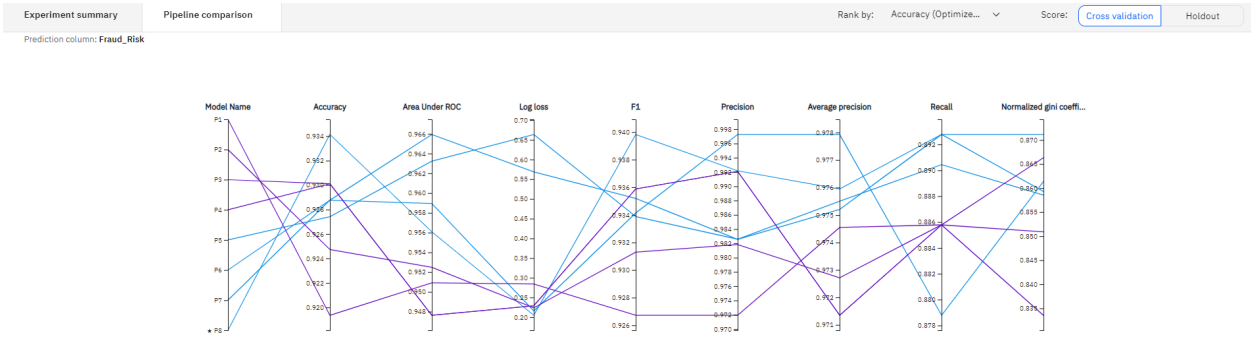
New model from file +

Name	Type	Software specification	Last modified
Classification - P8 RandomForestClassifierEstimator	wml-hybrid_0.1	hybrid_0.1	Oct 07, 2020



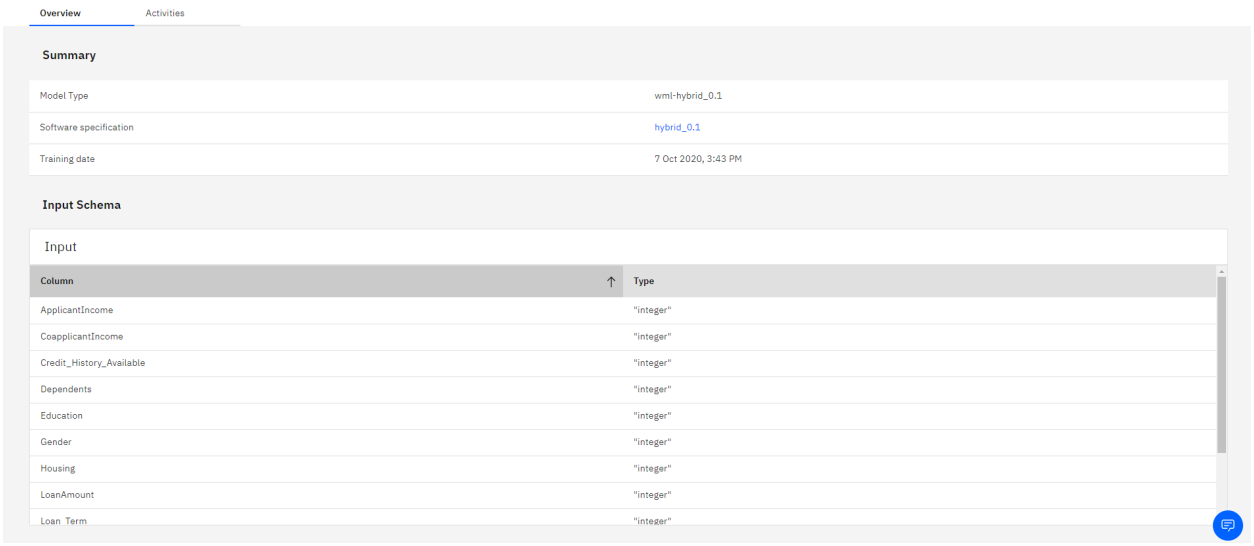
Pipeline leaderboard

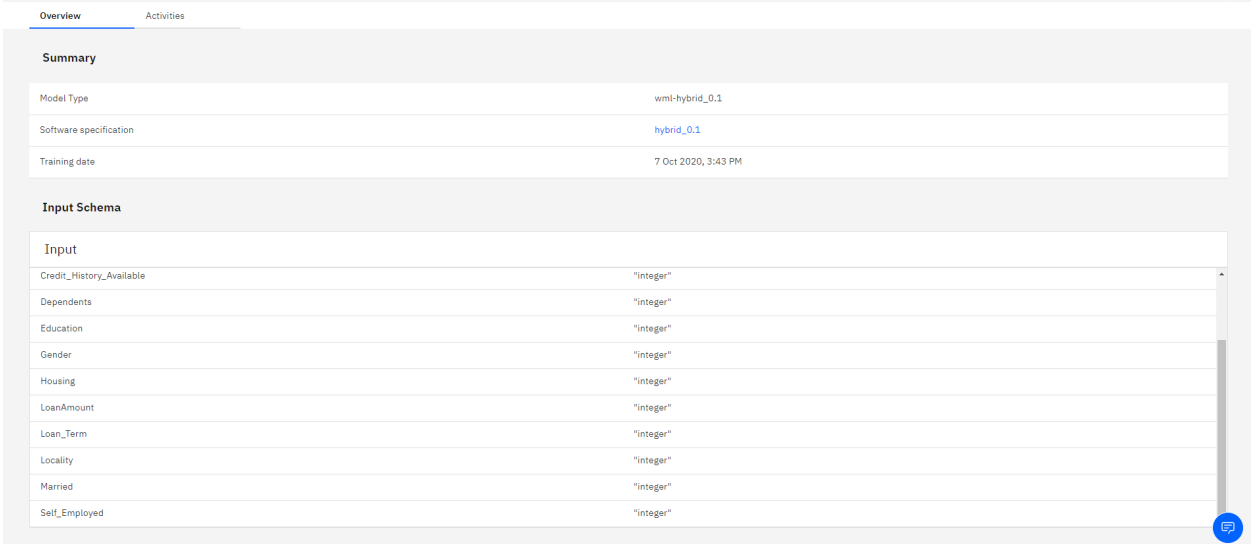
Rank	Name	Algorithm	Accuracy (Optimized)	Enhancements	Build time
★ 1	Pipeline 8	Random Forest Classifier	0.934	HPO-1 FE HPO-2	00:00:39
2	Pipeline 3	LGBM Classifier	0.930	HPO-1 FE	00:00:46
3	Pipeline 4	LGBM Classifier	0.930	HPO-1 FE HPO-2	00:00:57
4	Pipeline 6	Random Forest Classifier	0.929	HPO-1	00:00:14
5	Pipeline 7	Random Forest Classifier	0.929	HPO-1 FE	00:01:27
6	Pipeline 5	Random Forest Classifier	0.927	*None*	00:00:03
7	Pipeline 2	LGBM Classifier	0.925	HPO-1	00:00:17
8	Pipeline 1	LGBM Classifier	0.919	*None*	00:00:01

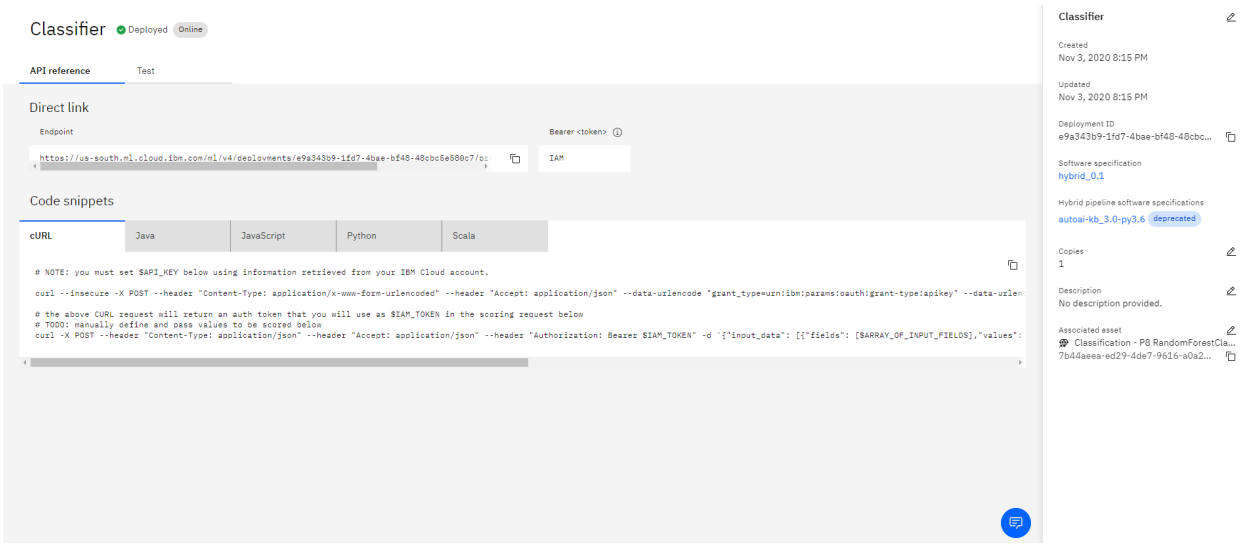


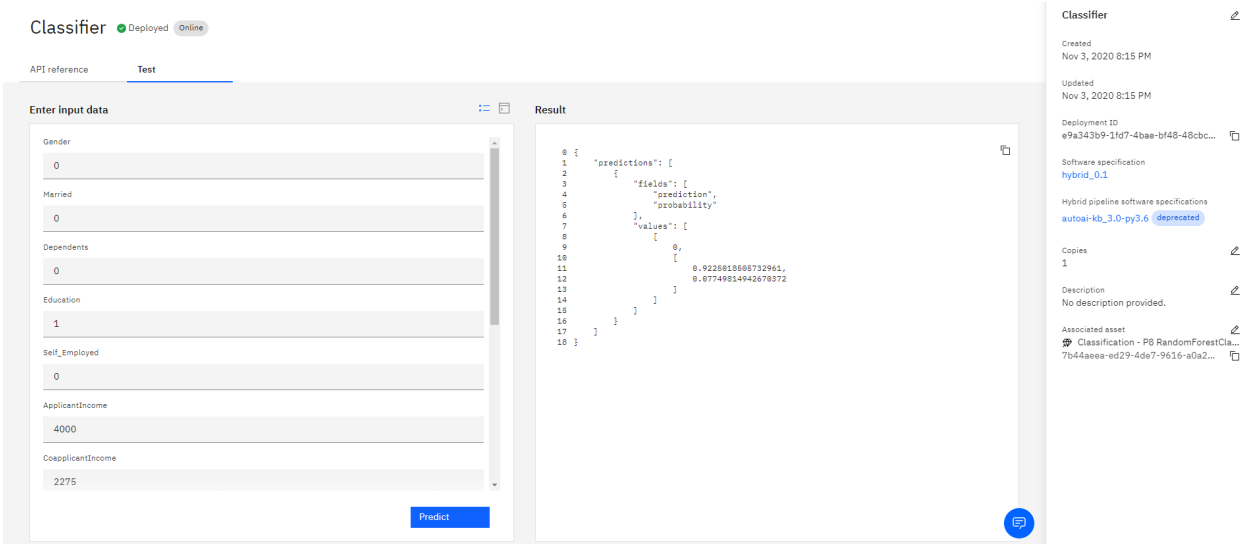
Pipeline leaderboard

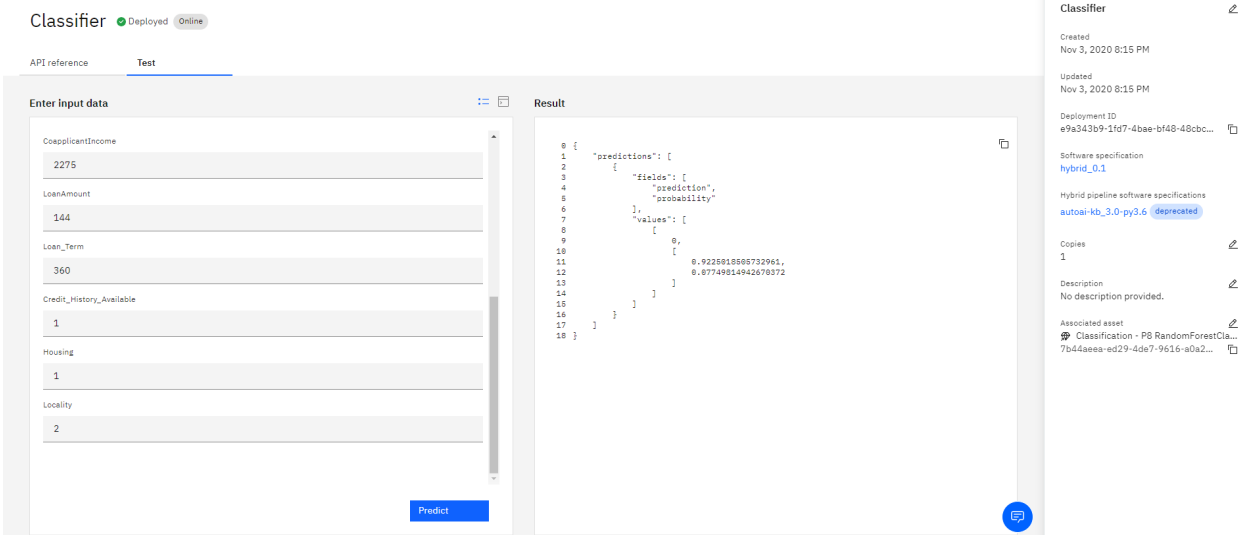
Rank	Name	Algorithm	Accuracy (Optimized)	Average precisi...	F_1	Log loss	Normalized Gin...	Precision	Recall	ROC AUC
★ 1	Pipeline 8	Random Forest Classifier	0.934	0.976	0.940	0.205	0.871	0.992	0.893	0.956
2	Pipeline 3	LGBM Classifier	0.930	0.971	0.936	0.229	0.866	0.992	0.886	0.948
3	Pipeline 4	LGBM Classifier	0.930	0.971	0.936	0.229	0.866	0.992	0.886	0.948
4	Pipeline 6	Random Forest Classifier	0.929	0.978	0.934	0.216	0.861	0.997	0.879	0.959
5	Pipeline 7	Random Forest Classifier	0.929	0.975	0.935	0.569	0.859	0.983	0.893	0.966
6	Pipeline 5	Random Forest Classifier	0.927	0.975	0.934	0.664	0.859	0.983	0.890	0.963
7	Pipeline 2	LGBM Classifier	0.925	0.973	0.931	0.224	0.851	0.982	0.886	0.952
8	Pipeline 1	LGBM Classifier	0.919	0.975	0.927	0.284	0.834	0.972	0.886	0.951

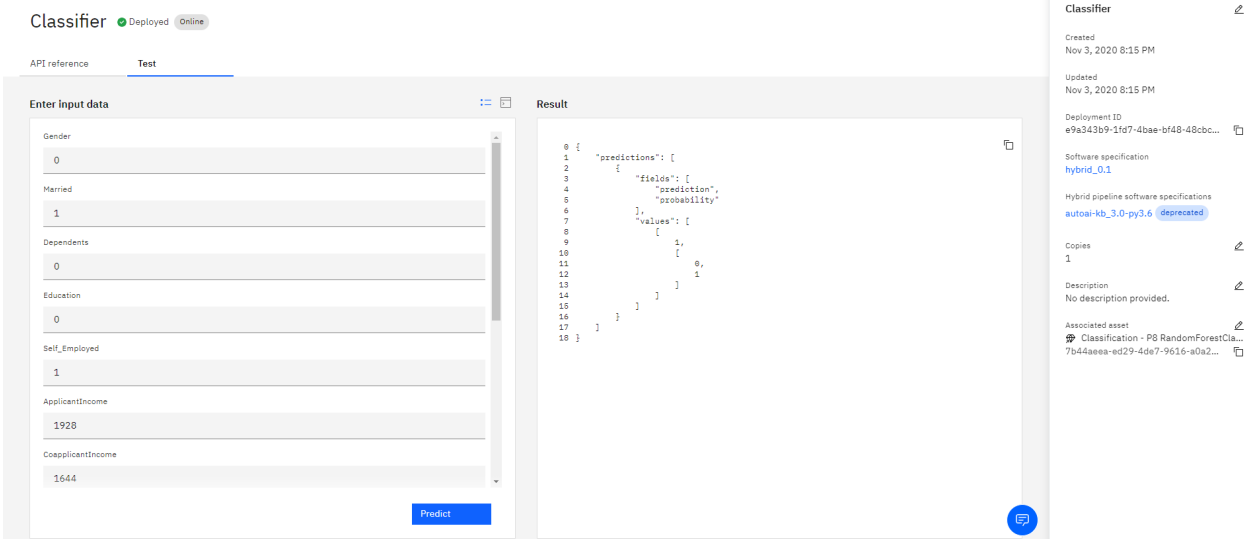

Overview
Activities
Summary
Model Type
wml-hybrid_0.1
Software specification
hybrid_0.1
Training date
7 Oct 2020, 3:43 PM
Input Schema
Input
Column
Type
ApplicantIncome
"integer"
CoapplicantIncome
"integer"
Credit_History_Available
"integer"
Dependents
"integer"
Education
"integer"
Gender
"integer"
Housing
"integer"
LoanAmount
"integer"
Loan_Term
"integer"

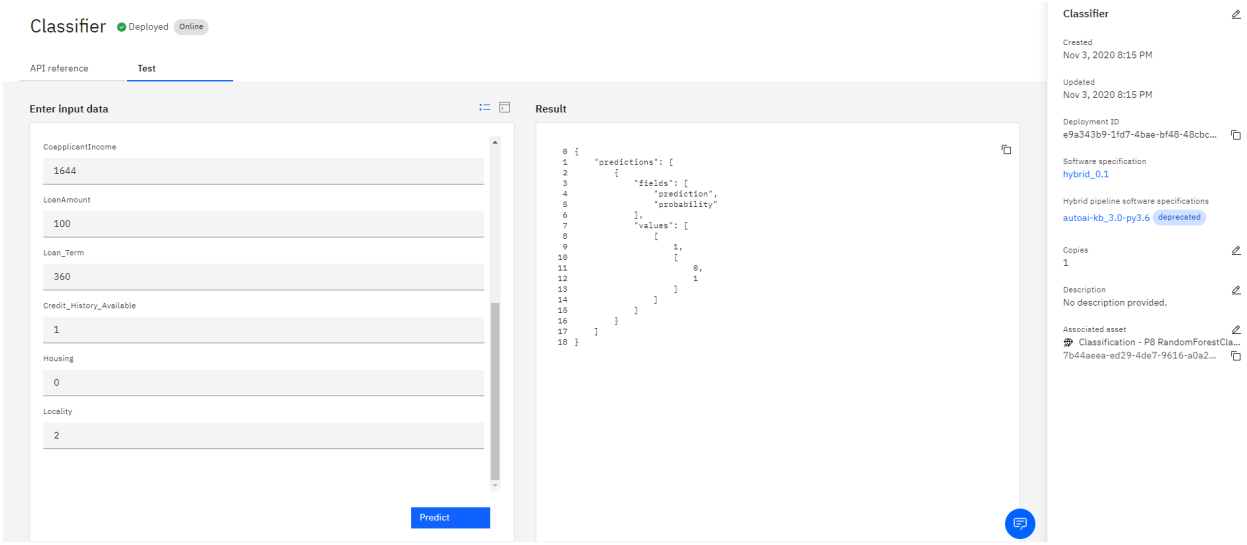

Overview
Activities
Summary
Model Type
wml-hybrid_0.1
Software specification
hybrid_0.1
Training date
7 Oct 2020, 3:43 PM
Input Schema
Input
Credit_History_Available
"integer"
Dependents
"integer"
Education
"integer"
Gender
"integer"
Housing
"integer"
LoanAmount
"integer"
Loan_Term
"integer"
Locality
"integer"
Married
"integer"
Self_Employed
"integer"


Classifier
Deployed
Online
API reference
Test
Direct link
Endpoint
Bearer <token>
IAM
Code snippets
cURL
Java
JavaScript
Python
Scala
NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account.
curl --insecure -X POST --header "Content-Type: application/x-www-form-urlencoded" --header "Accept: application/json" --data-urlencode "grant_type=urn:ibm:params:oauth:grant-type:apikey" --data-urlen
the above CURL request will return an auth token that you will use as $IAM_TOKEN in the scoring request below
TODO: manually define and pass values to be scored below
curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" --header "Authorization: Bearer $IAM_TOKEN" -d '{"input_data": [{"fields": [$ARRAY_OF_INPUT_FIELDS],"values":
Classifier
Created
Nov 3, 2020 8:15 PM
Updated
Nov 3, 2020 8:15 PM
Deployment ID
e9a343b9-1fd7-4bae-bf48-48cbc...
Software specification
hybrid_0.1
Hybrid pipeline software specifications
autoai-kb_3.0-py3.6
deprecated
Copies
1
Description
No description provided.
Associated asset
Classification - P8 RandomForestCla...
7b44aeea-ed29-4de7-9616-a0a2...


Classifier
Deployed
Online
API reference
Test
Enter input data
Gender
0
Married
0
Dependents
0
Education
1
Self_Employed
0
ApplicantIncome
4000
CoapplicantIncome
2275
Predict
Result
{
"predictions": [
{
"fields": [
"prediction",
"probability"
],
"values": [
[
0,
[
0.9226818508732961,
0.07749814942670372
]
]
]
}
]
}
Classifier
Created
Nov 3, 2020 8:15 PM
Updated
Nov 3, 2020 8:15 PM
Deployment ID
e9a343b9-1fd7-4bae-bf48-48cbc...
Software specification
hybrid_0.1
Hybrid pipeline software specifications
autoai-kb_3.0-py3.6
deprecated
Copies
1
Description
No description provided.
Associated asset
Classification - P8 RandomForestCla...
7b44aeea-ed29-4de7-9616-a0a2...


Classifier
Deployed
Online
API reference
Test
Enter input data
CoapplicantIncome
2275
LoanAmount
144
Loan_Term
360
Credit_History_Available
1
Housing
1
Locality
2
Predict
Result
{
"predictions": [
{
"fields": [
"prediction",
"probability"
],
"values": [
[
0,
[
0.9226818508732961,
0.07749814942670372
]
]
]
}
]
}
Classifier
Created
Nov 3, 2020 8:15 PM
Updated
Nov 3, 2020 8:15 PM
Deployment ID
e9a343b9-1fd7-4bae-bf48-48cbc...
Software specification
hybrid_0.1
Hybrid pipeline software specifications
autoai-kb_3.0-py3.6
deprecated
Copies
1
Description
No description provided.
Associated asset
Classification - P8 RandomForestCla...
7b44aeea-ed29-4de7-9616-a0a2...


Classifier
Deployed
Online
API reference
Test
Enter input data
Gender
0
Married
1
Dependents
0
Education
0
Self_Employed
1
ApplicantIncome
1928
CoapplicantIncome
1644
Predict
Result
{
"predictions": [
{
"fields": [
"prediction",
"probability"
],
"values": [
[
1,
[
0,
1
]
]
]
}
]
}
Classifier
Created
Nov 3, 2020 8:15 PM
Updated
Nov 3, 2020 8:15 PM
Deployment ID
e9a343b9-1fd7-4bae-bf48-48cbc...
Software specification
hybrid_0.1
Hybrid pipeline software specifications
autoai-kb_3.0-py3.6
deprecated
Copies
1
Description
No description provided.
Associated asset
Classification - P8 RandomForestCla...
7b44aeea-ed29-4de7-9616-a0a2...


Classifier
Deployed
Online
API reference
Test
Enter input data
CoapplicantIncome
1644
LoanAmount
100
Loan_Term
360
Credit_History_Available
1
Housing
0
Locality
2
Predict
Result
{
"predictions": [
{
"fields": [
"prediction",
"probability"
],
"values": [
[
1,
[
0,
1
]
]
]
}
]
}
Classifier
Created
Nov 3, 2020 8:15 PM
Updated
Nov 3, 2020 8:15 PM
Deployment ID
e9a343b9-1fd7-4bae-bf48-48cbc...
Software specification
hybrid_0.1
Hybrid pipeline software specifications
autoai-kb_3.0-py3.6
deprecated
Copies
1
Description
No description provided.
Associated asset
Classification - P8 RandomForestCla...
7b44aeea-ed29-4de7-9616-a0a2...