Dealer Fraud Prediction

Summary:

One of the biggest problems in the Telco industry is to face possible frauds done to a company by dealer stores. Companies nowadays report seeing a subsequent increase in attempts by dealer to make fraudulent commissions for their own benefit.Managing these channels is a huge task and incurs lot of loss on the company side.

To prevent dealer fraud, Machine Learning algorithms help detect fraudulent dealers based on prior information of company purchases. In this demo we shall use the capabilities of HyperSense AI Studio to build a classifier model to help find fraudulent transactions.

The below model building demo is to illustrate HyperSense AI Studio capabilities of building the model using features such as

1. No-code – drag and drop ready-made operators.
2. Ease of data preparation with in-built mathematical functions
3. Quick Visualization to learn relationship between input features and their distributions.
4. Experimenting various models in no time
5. Deploying prediction models on production datasets.

This quick start follows a simple and most widely used workflow for an ML experiment:

1. Prepare the input data –
    i. Load the Data
    ii. Define additional features
2. Detect Outlier and see results

The overall ML model building effort using above features of HyperSense will take 15-20 mins.

Dealer Fraud Prediction Use Case Illustration:

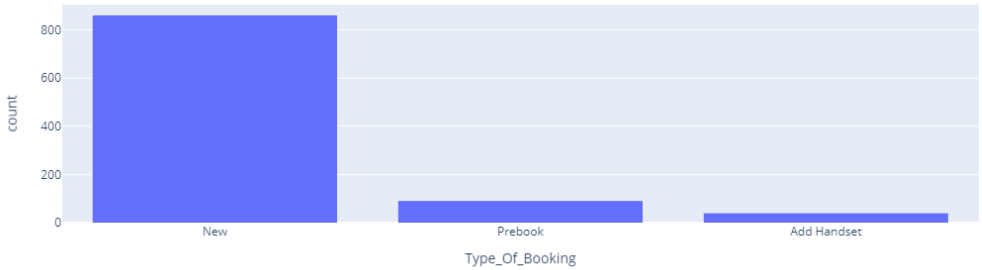
Detecting a fraud by just analyzing data is not a simple task. Using multiple factors such as, Dealer Category, Total Sales Count, Number of contracts; certain conclusion can be made to classify whether a customer makes a fraud purchase or not from the dealer. The pipeline we aim to build using HyperSense AI Studio can be deployed and run on a regular basis to help the company detect fraud transactions by feeding data containing multiple features.

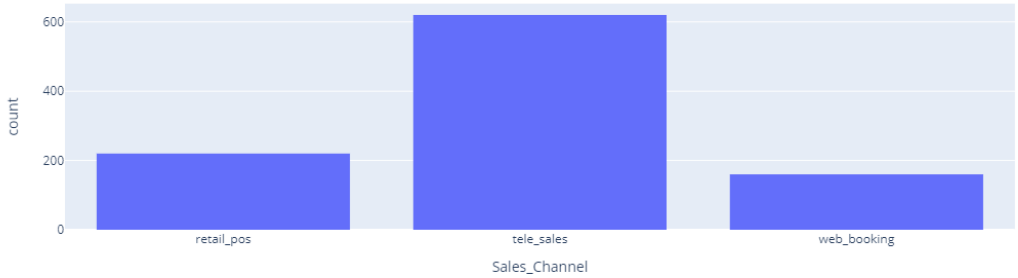
In the below implementation, we will see how to create a Dealer Fraud Prediction model.

Data source:link
Run Time: 3-4 mins
Efforts to build: Max. 30mins
Video Demo: link

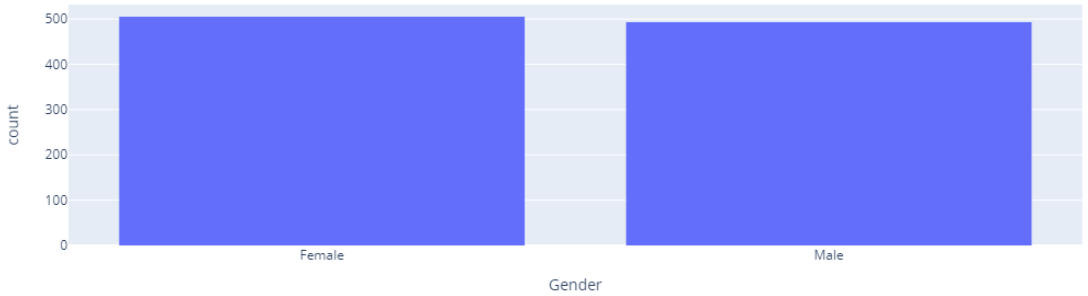
1. Inputs:

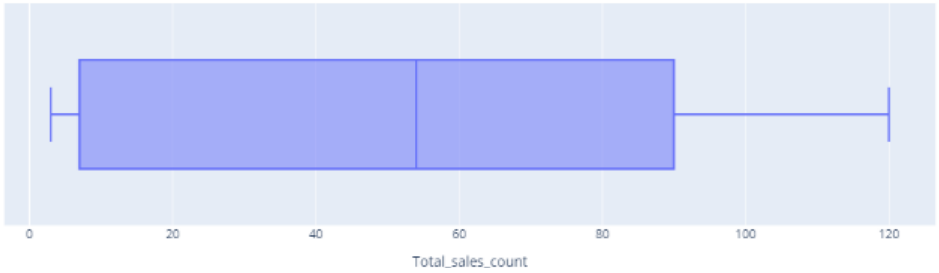
The first thing we need in machine learning is data. For this illustration, we use the Dealer Fraud Dataset, which is included in the data source section of this document. This dataset contains entries of various customersalong with respective dealer details. This dataset includes entries for individual customers, their age,Type of product, Total sales amount and count, Dealer information such as Dealer ID, Category and AIN Days.

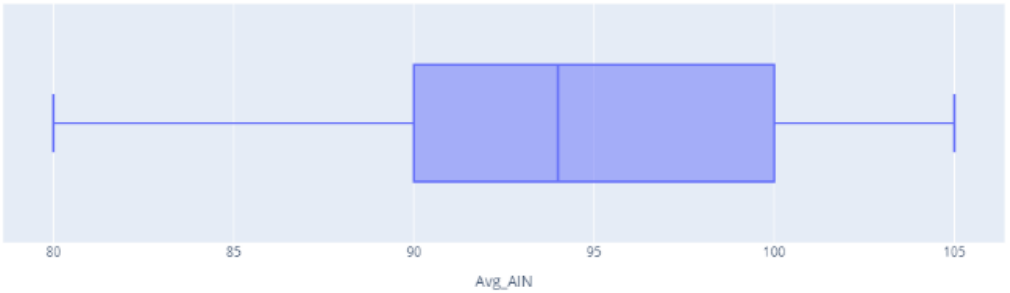


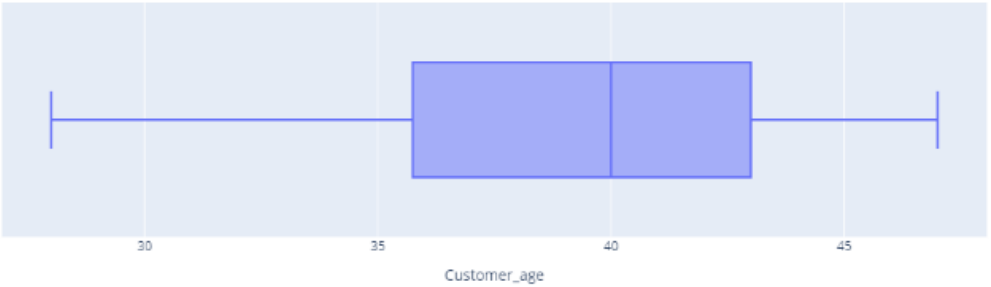


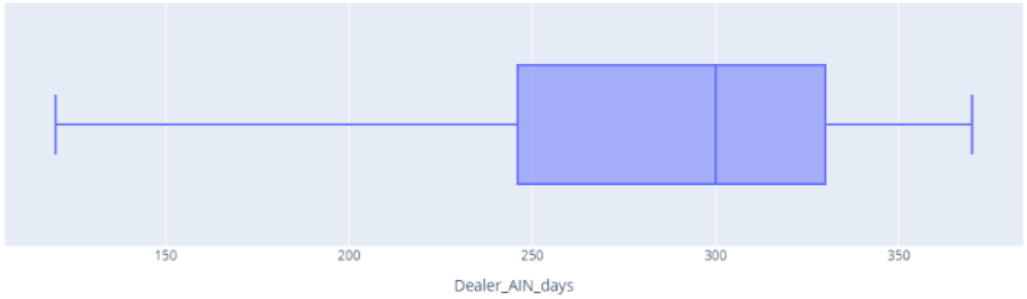

count
100
80
60
40
20
0
Oppo
Oneplus7
Samsung_note
Redmi
Samsung_galaxy
Realme
IPhoneX
Nokia
Pixel
Oneplus
Item_Model

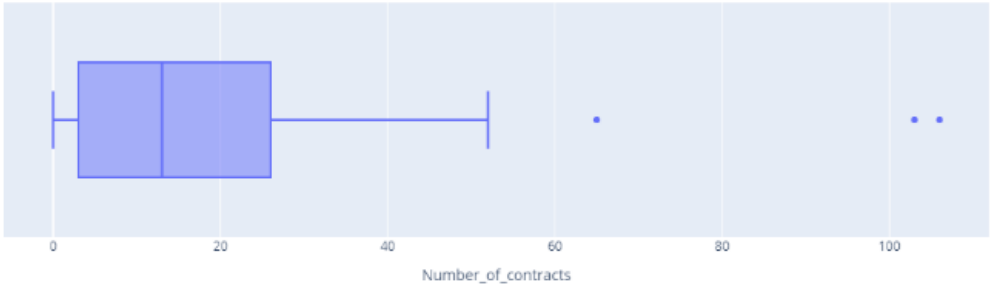

count
500
400
300
200
100
0
Female
Male
Gender


0
20
40
60
80
100
120
Total_sales_count


80
85
90
95
100
105
Avg_AIN


30
35
40
45
Customer_age


150
200
250
300
350
Dealer_AIN_days


0
20
40
60
80
100
Number_of_contracts

The entire Data Preparation process is performed by 3 operators provided by HyperSense AI Studio.

- CSV Reader Operator
- Missing Value Treatment Operator
- Encoder Operator

Features are individual attributes which contribute during the model training process. In the Dealer fraud dataset, each row represents a customer, and each column is a feature of that customer. Using these operators of HyperSense AI Studio, we get clean data in the right format which any classification model would accept as input.

The process is summarized below –

1.1. CSV Reader –

CSV Reader allows the user to load CSV files (Input) and converts the data into a readable format and can be used to feed the Machine Learning model.

1.2. Missing Value Treatment –

This operator allows the user to fill the missing entities in the dataset as per the user's decision. Missing values in each feature present in the dataset can be treated by various approaches. Users can treat missing values in continuous features with Mean, Median, Mode, KNN and User Imputation whereas categorical features can be treated with Mode and User imputation.

1.3. Encoder –

Not all machine learning algorithms can handle categorical variables as they are without any transformations. Encoding becomes a vital step where the categorical variables whose values exist as labels need to be converted into numeric form which can be well understood by the model. Dealer Category is encoded in our Dealer Fraud dataset.

This concludes the Data Preparation process required for the Dealer Fraud dataset.

2. Outlier Detection and Results:

2.1. Outlier Detection –

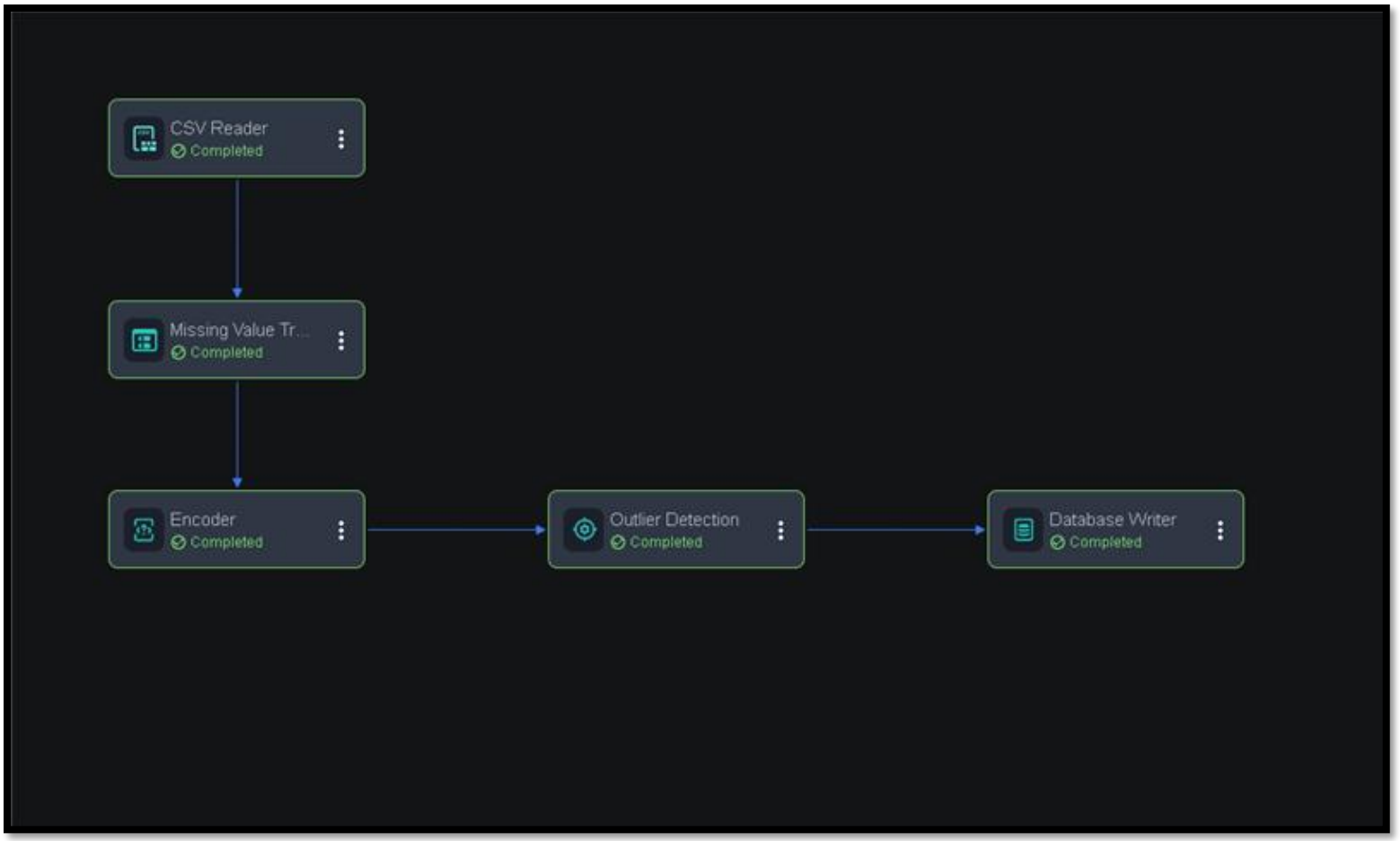
Removing outlier prior to training results in good prediction. Using the Outlier Detection Operator, users can view or remove the outliers by checking IQR or using methods such as DBSCAN and Isolation Forest.

Isolation Forest is used for the Multivariate outlier detection on our Dealer Fraud Dataset and based on four features –Number of contracts, Dealer Category, Dealer AIN Days and Total Sales count. The dealers which fall on the

outlier region are labelled as an Outlier. Isolation Forest tries to separate out scattered data points from those which appear closer together.

Based on the number of estimator (Number of trees that will be formed) and contamination (percentage of anomalous points we require), the anomaly data points are found and labelled as outliers.

This completes the entire process, and our pipeline is now ready to run and show the results.



Final ML Pipeline for Dealer Fraud prediction Model

3. Data Insights

3.1Predictions:

The final output is the outlier detection predictions itself,which contains the data points labelled byIsolation Forest.

DEALER_ID	TYPE_OF_PRODUCT	TOTAL_SALES_AMOUNT	TOTAL_SALES_COUNT	NUMBER_OF_CONTRACTS	DEALER_CATEGORY	DEALER_AIN_DAYS	CUSTOMER_AGE	AVG_AIN	OUTLIERS
A008	Iphone	60000.0	60.0	26.0	0.0	287.0	41.0	100.0	0
A003	Iphone	60000.0	60.0	26.0	0.0	287.0	31.0	100.0	0
A002	Samsung_note	40000.0	40.0	0.0	1.0	331.0	42.0	100.0	0
A002	Redmi	42000.0	84.0	13.0	1.0	332.0	37.0	94.0	0
A009	IPhoneX	6000.0	6.0	4.0	2.0	122.0	31.0	100.0	0
A009	Realme	60500.0	110.0	106.0	2.0	122.0	28.0	105.0	1
A003	OnePlus	30000.0	60.0	13.0	0.0	287.0	34.0	105.0	0
A004	Iphone	35000.0	60.0	26.0	0.0	301.0	47.0	90.0	0
A002	Samsung_galaxy	66000.0	105.0	13.0	1.0	333.0	47.0	90.0	0
A006	Oppo	5000.0	10.0	3.0	2.0	232.0	28.0	90.0	0

4. Results

4.1 Data Write S3:

Model results can be saved to different databases using Database writer operator, currently we can see that Outlier Detector operator results are being written to AWS S3 cloud storage.