**Training the model**

**Objective of the model:**

To reduce the false positive rate

**Details:**

* We have a logistic regression model in place, however we are currently getting very high false positive rate.
* We want to have another layer of model (preferably neural network – using tensorflow) so that the false positive rate could be reduced (by almost 80%)

**Quick note on false positive:**

|  |  |  |  |
| --- | --- | --- | --- |
| **False positive:** In actual the transaction was not fraud but the model identified it as fraud | | **Predicted** | |
| **No Fraud (0)** | **Fraud (1)** |
| **Actual** | **No Fraud (0)** | True Negative | False Positive |
| **Fraud (1)** | False Negative | True Positive |

**Artificial Neural Network (inputs):**

* **Input 1:** The output of the logistic regression (where Fraud = 1) is going as an input. Out of these (predicted frauds) we have to identify the True positive and False positive (frauds).
* **Input 2:** Some fields from the Data
* **Input 3:** New data sources / data fields that have been identified
* **Input 4:** Suppression rules
* **Input 5:** Weights – There is no way to control weights in the neural network, however, the weight of new frauds should be more so that the model learns faster. We can use radioactive decay function to solve this issue. Yweight = Y \* e-λt

**Predicted Y (Fraud):**

**However the false positive rate is very high**

**Artificial Neural Network**

**Predicted Y**

**Predicted Y (Fraud): False positive rate is low**

**New data sources / data fields**

**Weights**

**Suppression rules**

**Logistic Regression**

**Data**

**Few fields from here**

**Working of the model**

**Morning (Can be any cycle):**

* Once the model is implemented, the new data is passed through the model and predicted frauds are reported.
* The reported frauds are checked and the checker puts the correct flag (fraud / no fraud).

**Evening (Can be any cycle):**

* The new data with correct flags is used along with the historical data to re-train the model
* The model is validated and is ready to use for the next day

**Quick note on artificial neural network:**

|  |  |
| --- | --- |
| This “neuron” is a computational unit that takes as input x1,x2,x3 (and a +1 intercept term), and outputs **hW,b(x)=f(WTx)=f(∑Wixi+b)**, where f:R↦R is called the activation function.  In these notes, we will choose f(⋅) to be the sigmoid function:  **f(z)=1 / (1+exp(−z))** | http://ufldl.stanford.edu/tutorial/images/SingleNeuron.png |
| A neural network is put together by hooking together many of our simple “neurons,” so that the output of a neuron can be the input of another.  The circles labeled “+1” are called bias units, and correspond to the intercept term. The leftmost layer of the network is called the **input layer**, and the rightmost layer the **output layer**. The middle layer of nodes is called the **hidden laye**r, because its values are not observed in the training set. We also say that our example neural network has 3 input units (not counting the bias unit), 3 hidden units, and 1 output unit.  We will write a(l)i to denote the activation of unit i in layer l. For l=1, we also use a(1)i =xi to denote the i-th input. Given a fixed setting of the parameters W,b, our neural network defines a hypothesis hW,b(x) that outputs a real number. | http://ufldl.stanford.edu/tutorial/images/Network331.png |

**Adding more layers of models:**

* We can add another layer of model (like Gradient Boosting or SVM) between the logistic and artificial neural network.
* Such layers might help to further reduce the false positive rate.

**Feedback loop:**

**The predicted frauds are reported**

**The reported frauds are checked and the checker puts the correct flag (fraud / no fraud).**

**The new data with correct flags is used along with the historical data to re-train the model**

**Model**

**Predicted Y**

**Data**

**Reducing the false positive rate**

**The following options can be tried:**

* **More data:** Additional datasets / data fields will help to increase the accuracy of the model
* **More layers of models:** Adding another layer of model (like Gradient Boosting or SVM) between the logistic and artificial neural network. Such layers will help to further reduce the false positive rate.
* **More weights to the new data:** There is no way to control weights in the neural network, however, the weight of new frauds should be more so that the model learns faster. We can use radioactive decay function to solve this issue. Yweight = Y \* e-λt

**Monitoring of the model:**

The model needs to be monitored on regular intervals to access the accuracy of the model. We need to have certain threshold in place to act as trigger in case the model deteriorates.