ATAL BIHARI VAJPAYEE INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT GWALIOR

BTECH. PROJECT REPORT

Collaborative Artificial Intelligence in a Public Blockchain Network

Submitted by:

Rathin R

(2019IMT-081)

Supervised by:

Dr Saumya Bhadauria

OUTLINE:

- Introduction
- Motivation
- Objective
- Methodology
- System Summary
- Benefits
- Results
- Conclusion
- Future Scope

Introduction:

It is a framework for sharing and improving a machine learning model. In this framework, anyone can freely access the model's predictions or provide data to help improve the model. An important challenge is that the system must be robust and incentivize participation, but discourage manipulation. Our framework is modular, and we propose and justify three example choices of "incentive mechanisms" with different advantages.

The goal of this work is to address the current centralization of artificial intelligence by sharing models freely. Such centralization includes machine learning expertise, siloed proprietary data, and access to machine learning model predictions (e.g. charged on a per-query basis).

Problem Statement/Motivation:

Difficult to set up AI/ML systems :

- Lack of Quality/Inaccessible Data: Data being one of the most valuable entity in this era is quite inaccessible to common people or those that we get our hands into are poor quality data.

- Inadequate Infrastructure : To train Large data sets a computer requires high performance GPU, which is apparently quite expensive.

Model Decay : A Model looses its accuracy if it is not updated regularly. For example considering a model which predicts the rent of houses in an area. If the model is not updated regularly it will still calculate rents based on previous values not considering the new rents as time goes by(rent prices increases in real time).

AI Skills are centralized : Data sets are largely owned by certain organizations (eg.Google) which then sell those data collected from the users on per query basis. Google generates a revenue of 92M$ every 24hrs.

Objectives

- Store Models in the Blockchain. Put Models in Smart Contracts so that anyone can update them, they can be used for free since they only read the contract’s state.

- Now that we have a public Model , we’ll let people update it & use those updates to build a dataset.

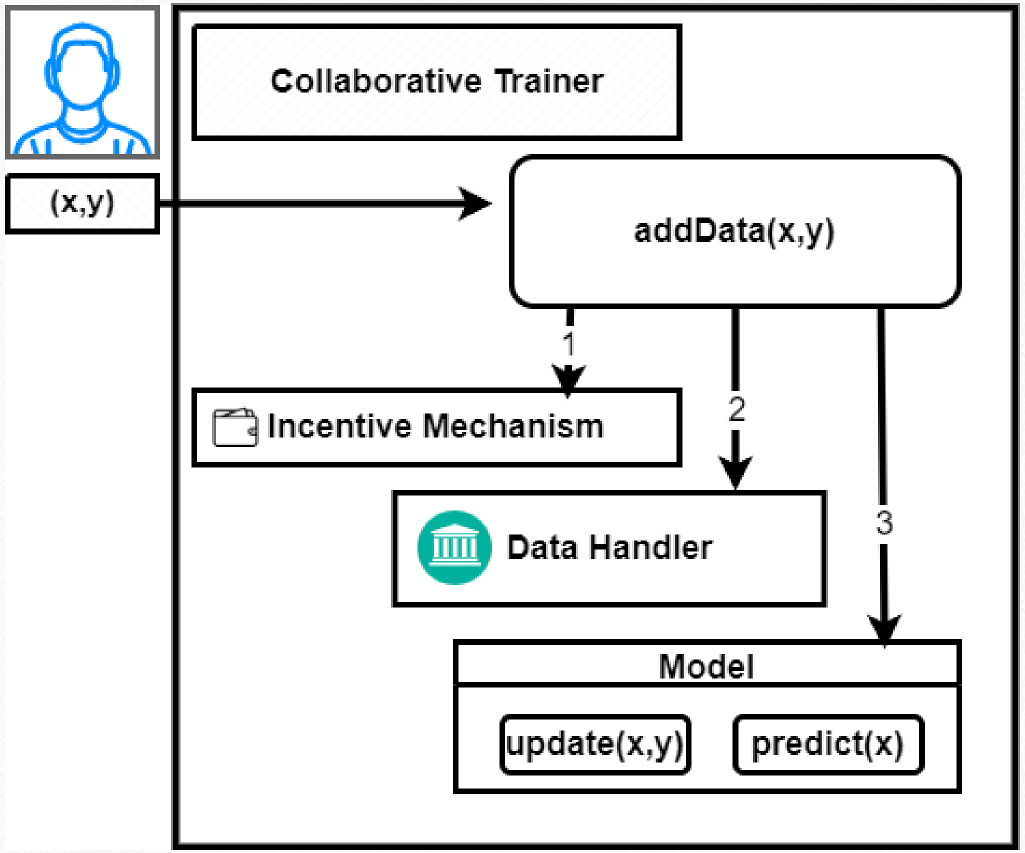
- To create an immutable warehouse of datasets that, once stored in the blockchain as "good data," cannot be manipulated—exploiting the hackproof property of blockchain to secure the datasets from manipulators.

- The main goal is to create a "Wikipedia" of datasets. Any ordinary man should have access to data for free at the ease of technology at our fingertips. Models can be trained in this Dapp by giving a small number of gas fees to the network. This can then be trained using blockchains computation power.

Methodology

Overview

3 steps for adding data



- Boxes are smart contracts
- Rounded boxes are methods

- Supervised machine learning: data with labels: (x, y)
- Minimize gas costs → efficient to train models
  - E.g. Naïve Bayes, Perceptron, or Nearest Centroid Classifier
- Encoding "off-chain" then fine-tuning "on-chain"
  - E.g. Image recognition, use VGG, MobileNetv2

Cost of using Ethereum for Training Text Classification

Action	Gas Cost	USD[1]
Deploy model contract of Perceptron with 100 weights	3,845,840	$4.71
Add data with 15 words (model agrees)[2]	177,693	$0.22
Add data with 15 words (model disagrees) [2]	249,037	$0.30

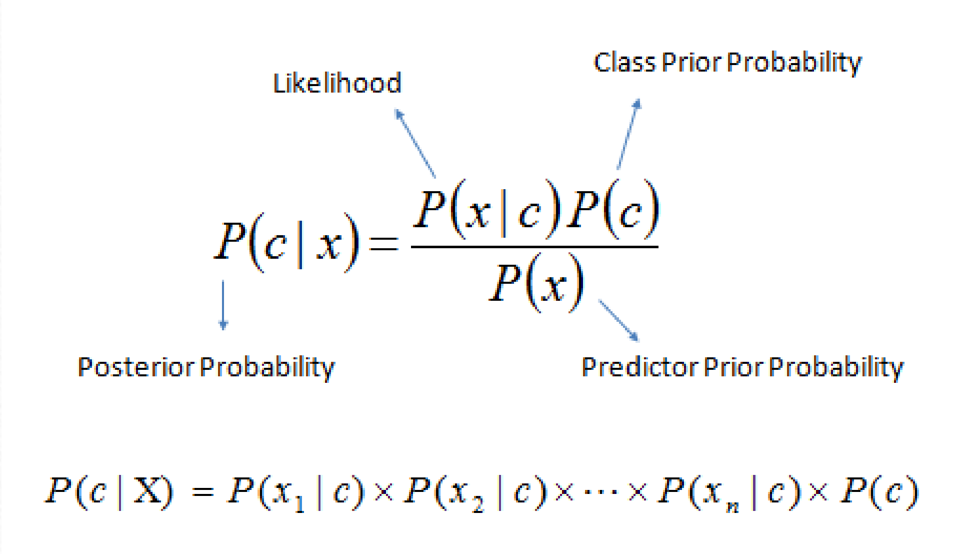
[1] Approximate costs in July 2019 with a modest gas price of 4gwei.

[2] Perceptron models are only updated when the model disagrees.

Models

Naïve Bayes

- Find the class with the most likely class for the data.
- x: list of features
- c: the classification/label
- Update: increment various counts



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

$$P(x_i|c)$$

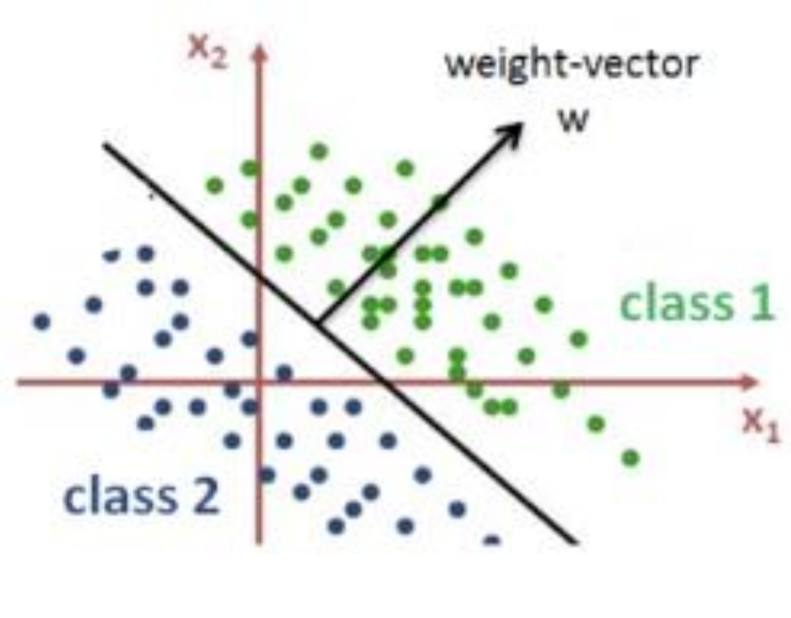
$$\text{probability of feature } i \text{ appearing in a sample in class } c$$

$$= \frac{\text{number of times feature } i \text{ appear in a sample of class } c}{\text{total count of all features in class } c}$$

Models

Perceptron

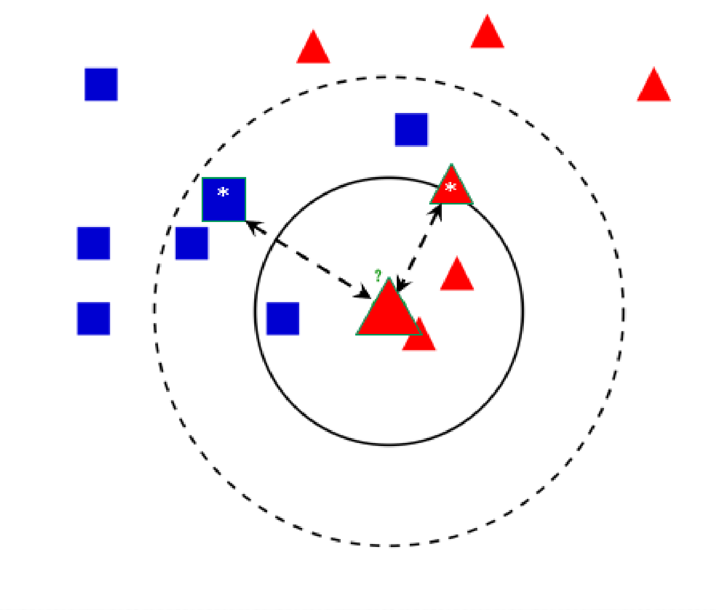
- Only update when expected class ≠ predicted class
- Predicted classification: $\hat{y} = w(t) \cdot x + b$
- Easy to update: $w(t+1) = w(t) + r \cdot (y - \hat{y}) \cdot x$
  - x : data (vector/list of numbers)
  - y : expected classification/label (number)
  - $w(t)$: weights at time t (vector/list of numbers)
  - r : learning rate (number)
  - b : bias/offset (number)



Models

Nearest Centroid Classifier

- Find the most similar point to your data
- Easy to update moving average: $\mathrm{avg}(t+1) = \frac{x + n \cdot \mathrm{avg}(t)}{t+1}$
- Enforce normalized: no one can move a centroid too much
- Encode "off-chain" using a known encoder: tested with 512 dimensions



Incentivizing quality data

There are many ways to encourage contributors to submit good quality data.

We analyze several examples in our paper:

1. **Gamified** (non-financial, points + badges like Stack overflow)
2. Based on established theory in **Prediction Markets**
3. Deposit, Refund, and Take: **Self-Assessment**

Incentivizing quality data

BASED ON PREDICTION MARKETS

Here an outside party, such as an academic institution or a company, provides a pool of reward funds and a test dataset. Participants are rewarded according to how well they improve the model's performance as measured by the test data. When this provider is available, we will be able to give very robust incentives for participation. The mechanism is also resilient against manipulative or malicious providers and participants

Phases

1) Commitment
2) Participation
3) Reward

1) Commitment Phase

- A generous **provider** stakes a **bounty** to be split and rewarded to contributors.
- Now the provider must prove they have a valid **test set** but without revealing all of it yet.[1]
    - Provider shares hashes for portions of their test set: $h_1, h_2, \dots, h_N$
    - Provider reveals a portion of the test set randomly chosen by a smart contract: $H = \{h_i : 1 \leq i \leq N\}, |H| < N$

[1] Similar to the DanKu Protocol: https://algorithmia.com/research/ml-models-on-blockchain

2) Participation Phase

- **Participants** submit one training **data** sample at time along with a small **deposit** of funds.
- The shared **model** is **updated** using the provided data sample.

3) Reward Phase

- The **provider reveals** the rest of the **test set** and the smart contract validates that it matches the hashes they originally gave in the Commitment Phase.
- Participants are **rewarded** based on how much their data contribution helped the model **improve** its **accuracy** on the test set:

  change in loss (error rate): $L(h_{t-1}, D) - L(h_t, D)$

  change in accuracy: $A(h_t, D) - A(h_{t-1}, D)$

Incentivizing quality data

DEPOSIT, REFUND, AND TAKE: **SELF-ASSESSMENT:**

Here are the highlights of the proposal:

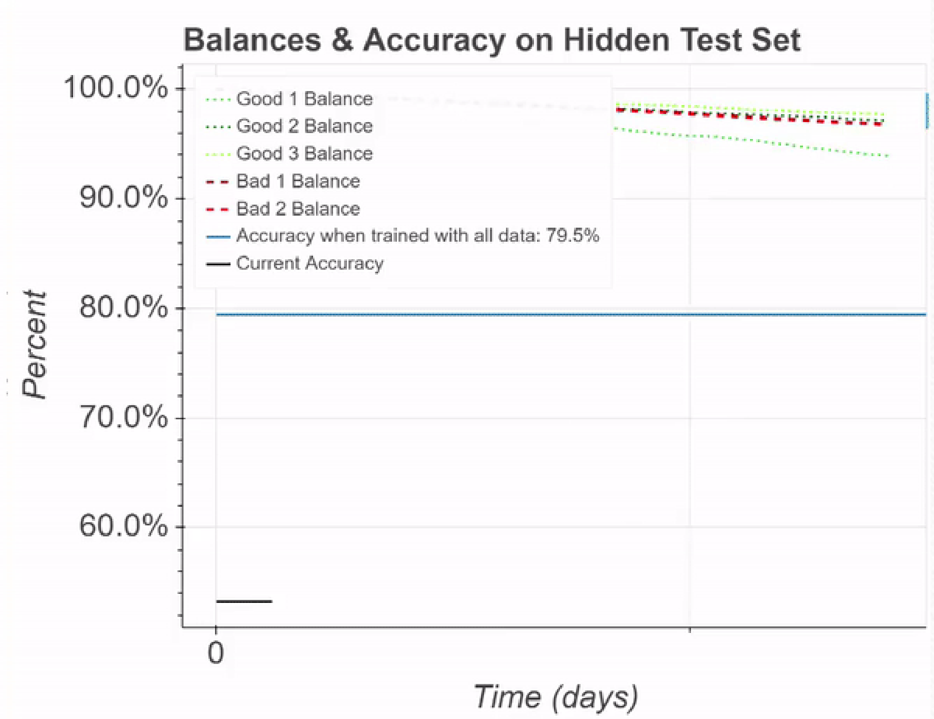
- Deploy a model, h, already trained with some data.

- Deposit: Each data contribution with data x and label y also requires a deposit, d. Data and meta-data for each contribution is stored in the data handler.

- Refund: To claim a refund on their deposit, after a time t has passed and if the current model, h, still agrees with the originally submitted classification, i.e. if h(x) == y, then the contributor can have their entire deposit d returned. – We now assume that (x, y) is "good" data. – The successful return of the deposit should be recorded in a tally of points for the wallet address.

- Take: A contributor that has already had data validated in the Refund stage can locate a data point (x, y) for which h(x) != y and request to take a portion of the deposit, d, originally given when (x, y) was submitted.

Rewarded based on accuracy improvement with respect to a test set.



Participants submit data and deposits.

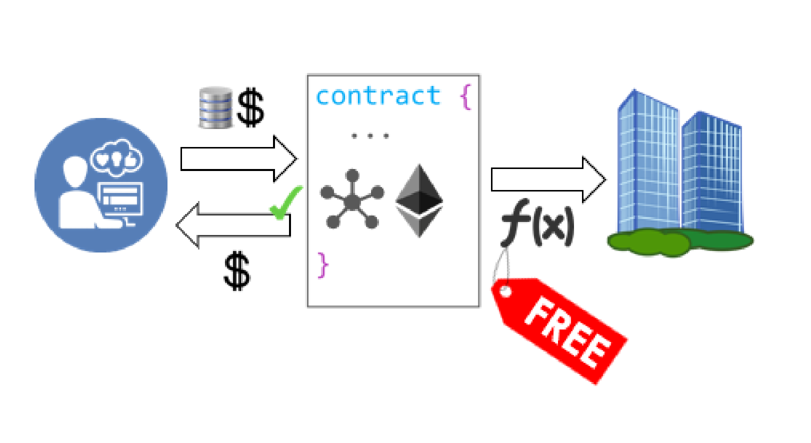
System Summary

Goals:

- Free to use models in smart contracts
- Build high quality datasets

Method:

- Deploy an initial model
- Contributors submit data + deposit
- Contributors can get a reward after submitting good data
- The model remains free to use for inference



Benefits

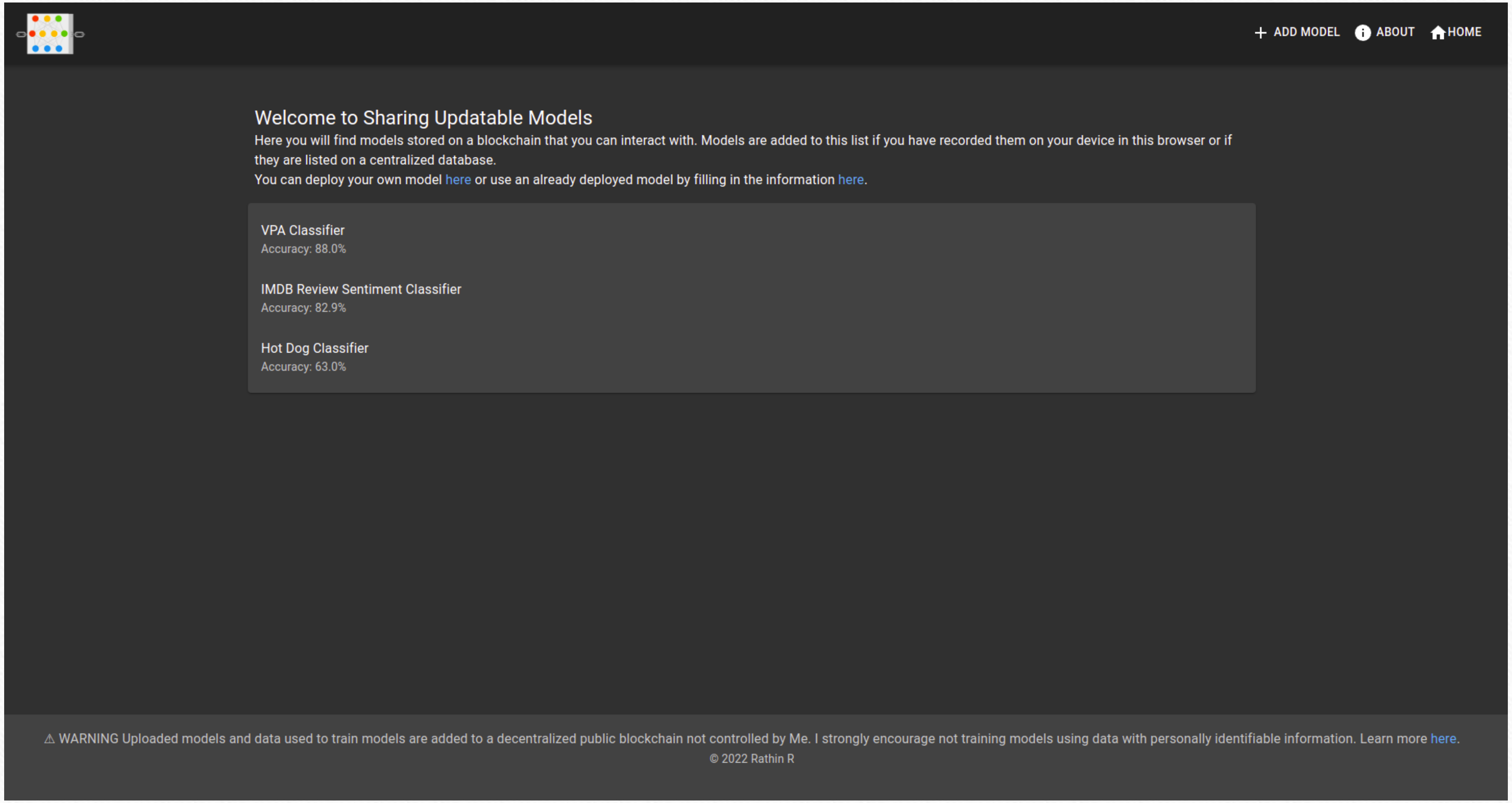
Consumers

- Improve products and services you use
- Instead of paying more for services, you pay **if you want** to update a service
- Chance for rewards: achievements or digital assets

Developers

- Easier to maintain deployed models
- Analytics for their models
- Services to direct to the best version of a choice of models

Results:



Welcome to Sharing Updatable Models

Here you will find models stored on a blockchain that you can interact with. Models are added to this list if you have recorded them on your device in this browser or if they are listed on a centralized database.

You can deploy your own model here or use an already deployed model by filling in the information here.

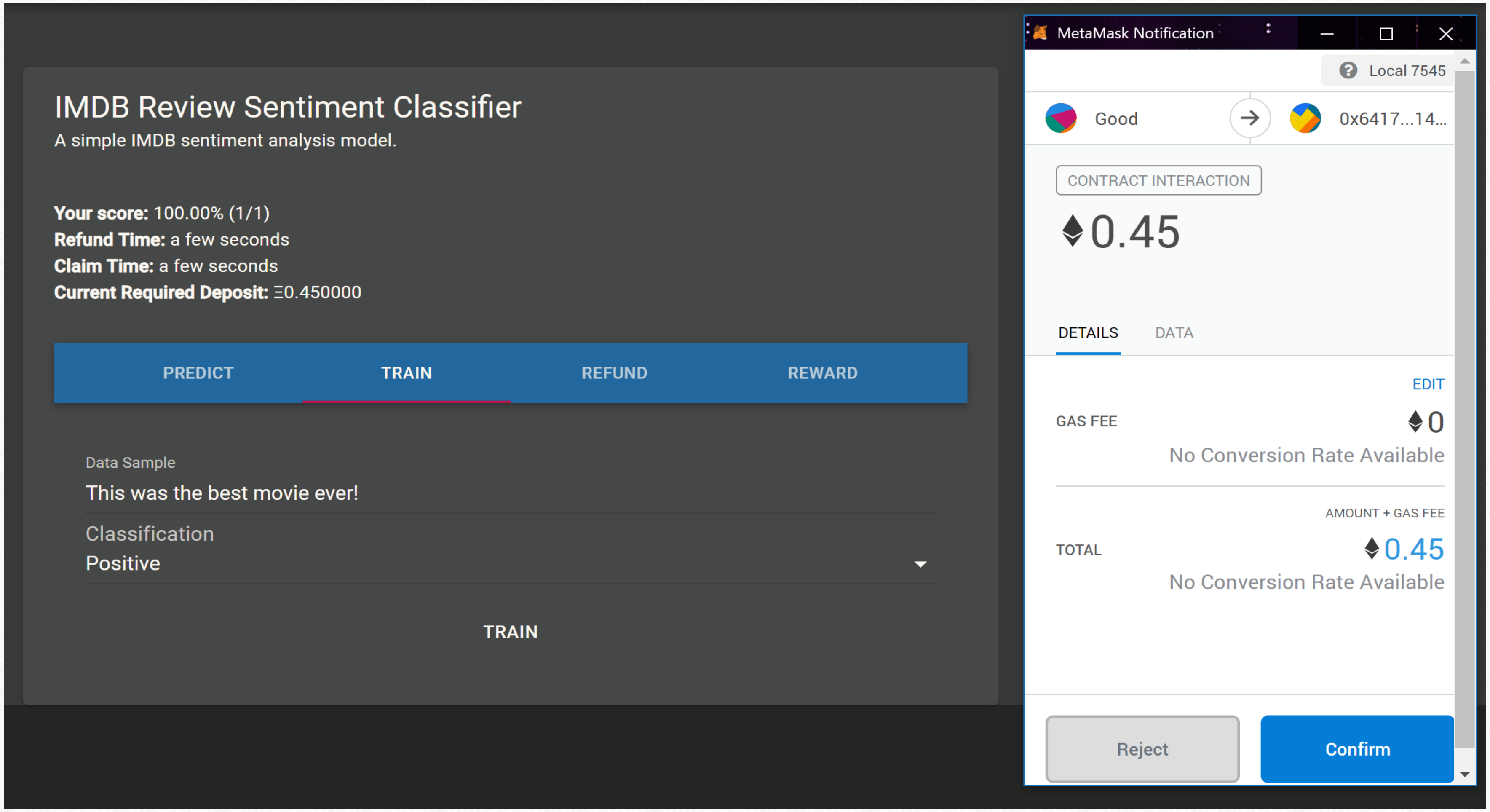
VPA Classifier
Accuracy: 88.0%

IMDB Review Sentiment Classifier
Accuracy: 82.9%

Hot Dog Classifier
Accuracy: 63.0%

⚠ WARNING Uploaded models and data used to train models are added to a decentralized public blockchain not controlled by Me. I strongly encourage not training models using data with personally identifiable information. Learn more here.

© 2022 Rathin R



IMDB Review Sentiment Classifier

A simple IMDB sentiment analysis model.

Your score: 100.00% (1/1)
Refund Time: a few seconds
Claim Time: a few seconds
Current Required Deposit: Ξ0.450000

Data Sample
This was the best movie ever!

Classification
Positive

MetaMask Notification

Local 7545

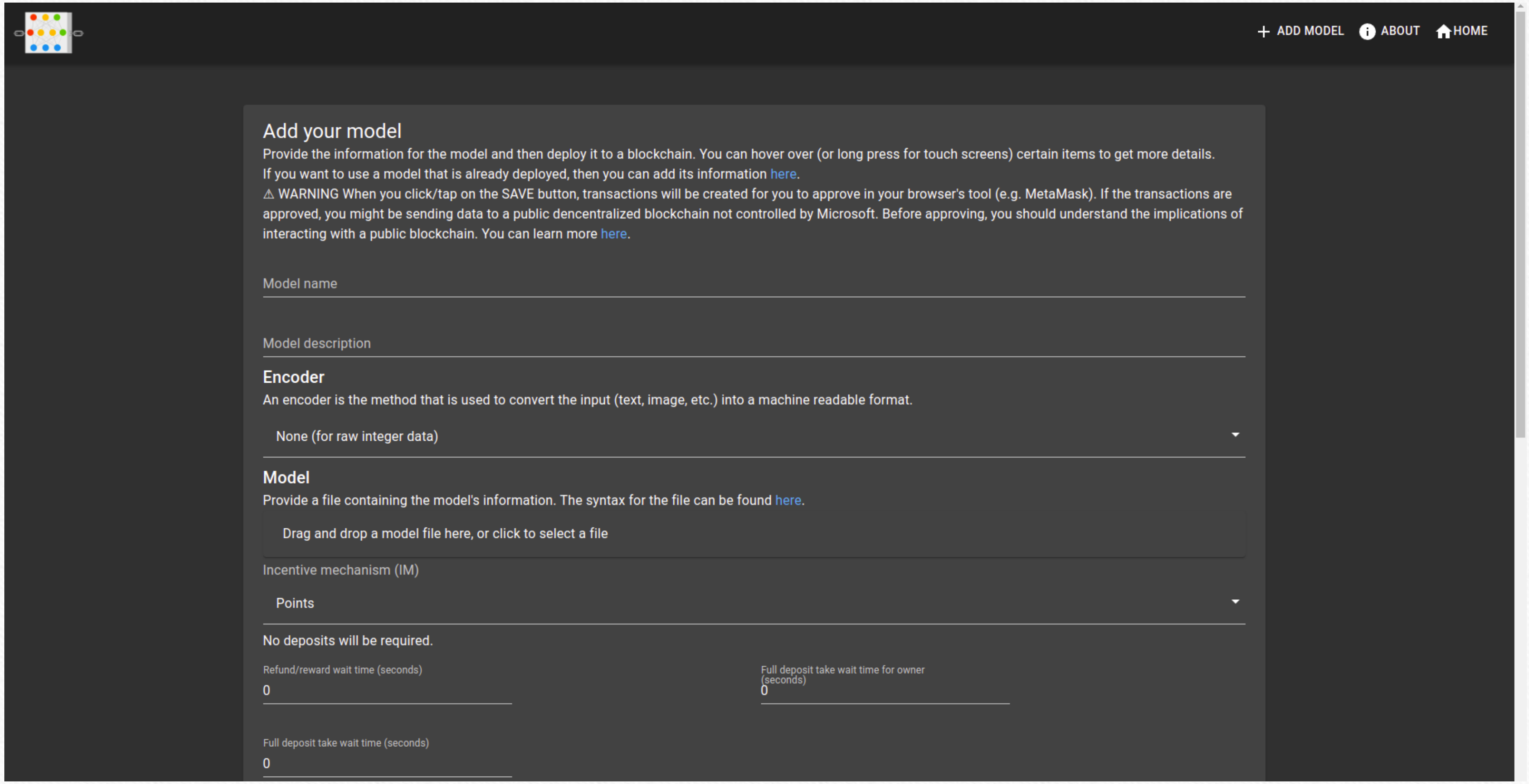
Good → 0x6417...14...

CONTRACT INTERACTION

0.45

GAS FEE 0
No Conversion Rate Available

AMOUNT + GAS FEE
TOTAL 0.45
No Conversion Rate Available



Add your model

Provide the information for the model and then deploy it to a blockchain. You can hover over (or long press for touch screens) certain items to get more details.

If you want to use a model that is already deployed, then you can add its information here.

⚠ WARNING When you click/tap on the SAVE button, transactions will be created for you to approve in your browser's tool (e.g. MetaMask). If the transactions are approved, you might be sending data to a public dencentralized blockchain not controlled by Microsoft. Before approving, you should understand the implications of interacting with a public blockchain. You can learn more here.

Model name

Model description

Encoder

An encoder is the method that is used to convert the input (text, image, etc.) into a machine readable format.

None (for raw integer data)

Model

Provide a file containing the model's information. The syntax for the file can be found here.

Drag and drop a model file here, or click to select a file

Incentive mechanism (IM)

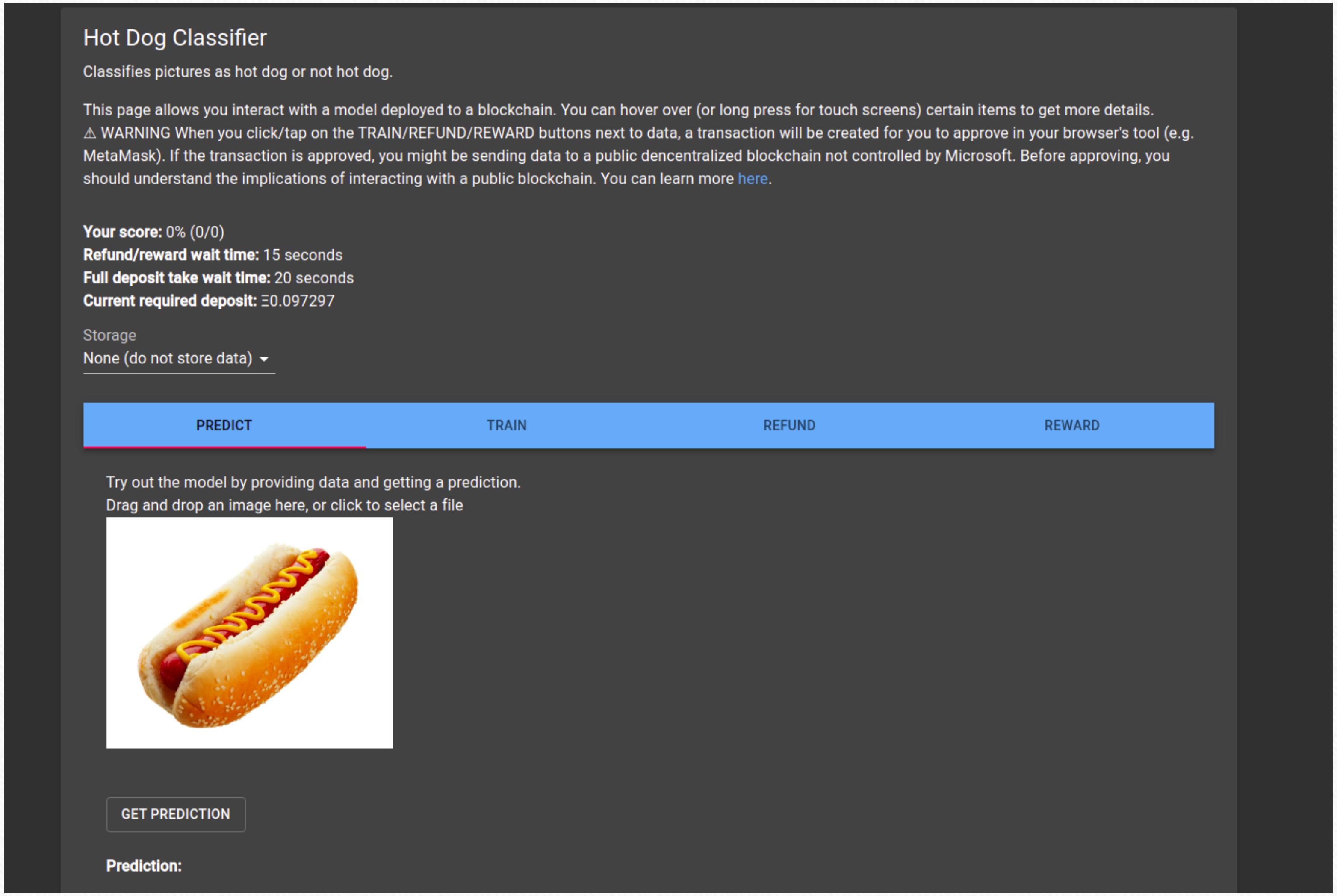
Points

No deposits will be required.

Refund/reward wait time (seconds)
0

Full deposit take wait time for owner (seconds)
0

Full deposit take wait time (seconds)
0



Classifies pictures as hot dog or not hot dog.

This page allows you interact with a model deployed to a blockchain. You can hover over (or long press for touch screens) certain items to get more details.
⚠ WARNING When you click/tap on the TRAIN/REFUND/REWARD buttons next to data, a transaction will be created for you to approve in your browser's tool (e.g. MetaMask). If the transaction is approved, you might be sending data to a public dencentralized blockchain not controlled by Microsoft. Before approving, you should understand the implications of interacting with a public blockchain. You can learn more here.

Your score: 0% (0/0)
Refund/reward wait time: 15 seconds
Full deposit take wait time: 20 seconds
Current required deposit: Ξ0.097297

Storage
None (do not store data)

Try out the model by providing data and getting a prediction.
Drag and drop an image here, or click to select a file

Prediction:

Future Scope:

There are a few areas where the presented framework can be configured and built upon.

A. Models

More research needs to be done on the types of models that will work well within this framework:

1) Unsupervised Models
2) Complex Models
3) Recovering Corrupted Models

B. Incentive Mechanisms

More exploration, analysis, and experiments with incentive mechanisms in this space needs to be done with emphasis on the type of model each incentive mechanism works well with. The incentive mechanisms imposed by the smart contract could be hidden to end users by 3rd party services that build services around this proposed framework. These services could validate data contribution themselves offering their own rewards to users of their platforms that do not wish to interact with these smart contracts.

C. Privacy

Contributors may not want to publish their data to a public blockchain. Initially we propose to only use this for framework for data that is safe to become public. E.g. certain queries to a personal assistant such as, "What will the weather be like tomorrow?", which contains no personal data. Future work can be done to not submit data directly to the smart contract and instead just submit model updates

Conclusion:

I have presented a configurable framework for training a model and collecting data on a blockchain by leveraging several baseline incentive mechanisms and existing types of machine learning models for incremental learning. Ideal scenarios have varying data with generally agreed upon labels.

Currently, this framework is mainly designed for models that can be efficiently updated but we hoping to improve the framework with compatibility with more complex models.