ESTIMATING THE PRESENCE OF IMPURITIES IN IRON ORE USING IBM WATSON MACHINE LEARNING

INTRODUCTION

Predicting-Iron-Ore-Impurity

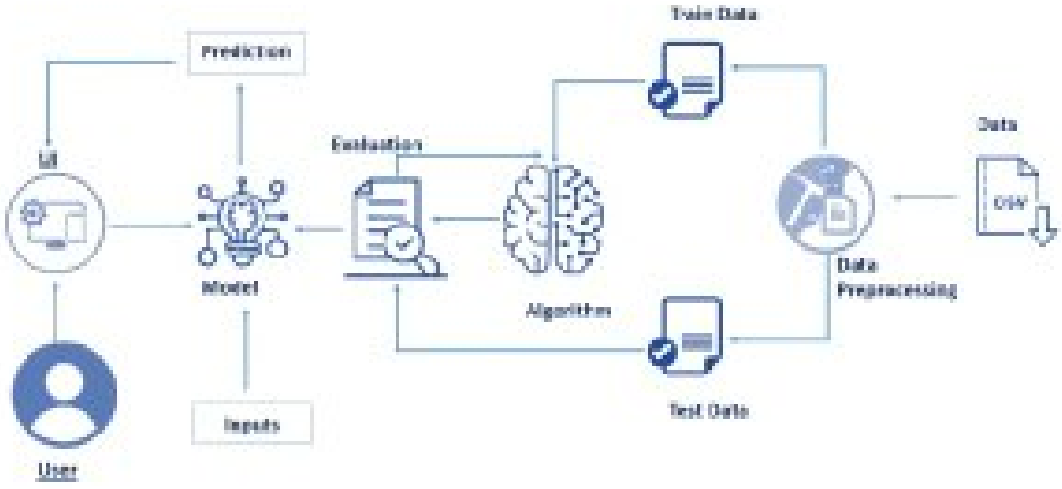
Ore is a naturally occurring solid material from which a metal or valuable mineral can be extracted profitably. The ore comes with naturally occurring impurities. One of such impurities is “Silica”. During the extraction of minerals from Ore, impurity is measured every hour. If we can predict how much silica (impurity) is in the ore concentrate, then we can provide such early information to the engineering team for corrective action. Basis the early information the engineering team will be able to reduce/control the impurity ensuring quality output through extraction from ore. It is also helpful for the environment as we reduce the amount of ore that goes into tailings (are the by-products left over from mining and extracting resources)

LITERATURE SURVEY

Random Forest. Random Forest (RF) is an ensemble machine learning proposed by Leo Breiman that combines predictors tree for predictive or classification task based on independent random samples of observations. In order to grow a tree, multiple random samples are drawn with replacement from the training dataset. This connotes that each tree would be grown with its version of the training dataset. Moreover, a subset of the explanatory variables is also randomly selected at each node during the learning process. After training, prediction scores are however determined by averaging the predictions from all individual

regression trees. This helps to attenuate overfitting and variance. Though, there are many methods that perhaps most likely to have impact on model performance. In this study, we focused on two main tuning hyperparameters that have significant effect on the prediction outcome of Random Forest.

THEORITICAL ANALYSIS



<u>Hardware Requirements</u>:

Processor: Intel® Core™ i3-2350M CPU @ 2.30GHz

Installed memory (RAM):4.00GB

System Type: 64-bit Operating System

EXPERIMENTAL INVESTIGATIONS

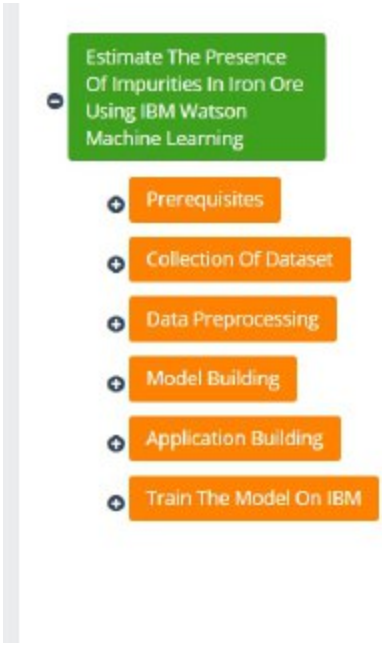
while working on the solution we investigated on the what is AL and what is ML and how to build models using them and how to do image processing. And mainly we had studied about the CNN because our solution mainly need this so we worked on these aspects.

Artificial Intelligence: Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems enabling it to even mimic human behaviour. Its applications lie in fields of Computer Vision, Natural Language Processing, Robotics, Speech Recognition, etc.

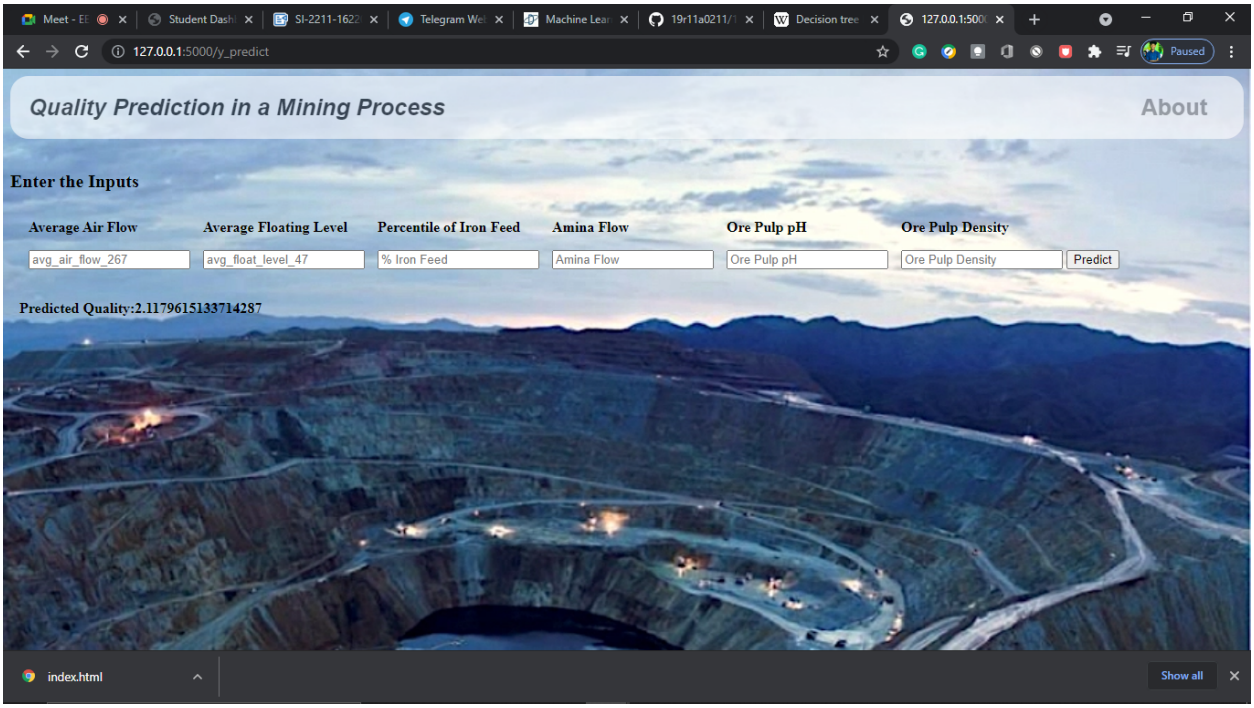
Transfer Learning: A major assumption in many machine learning and data mining algorithms is that the training and future data must be in the same feature space and have the same distribution. However, in many real-world applications, this assumption may not hold. For example, we sometimes have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest, where the latter data may be in a different feature space or follow a different data distribution. In such cases, knowledge transfer, if done successfully, would greatly improve the performance of learning by avoiding much expensive data labelling efforts. In recent years, transfer learning has emerged as a new learning framework to address this problem

Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called **classification trees**; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**. Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.

FLOWCHART



RESULT



ADVANTAGES:

1. We got the result with high accuracy.

2. API allows content to be embedded from any site or application more easily.

3. Using machine learning model that we developed, the impurity from the ore can be predicted with utmost accuracy.

4. Through API any user or company can customize the content and services that they use the most.

DISADVANTAGES:

1. Implementing and providing API can be costly in terms of development times

2. Ongoing maintenance requirements

3. Providing support

APPLICATIONS

. The main application of this model is to predict the impurity from an ore. It is well trained so that it will predict the correct data.

CONCLUSION

Without pre-trained Keras model, the train accuracy is 100 % and validation accuracy is 93.75.0%. The validation result had a best figure of 91.09% as accuracy. It is observed that without using pre-trained Keras model, although the training accuracy is >90%, the overall accuracy is low unlike where pre-trained model is used.

In the model that we have developed

FUTURE SCOPE

Future study for further development of the method will be in two-fold. On the one hand, a validation study should be conducted in order to evaluate the predictive power of the proposed model using dataset of more parameters.On the other hand, plans are in far advance to extend the application of the
methodology for different ores like Sulfide ores, Bauxite ores etc while considering different related parameters

BIBILOGRAPHY

1.https://en.wikipedia.org/wiki/Iron_ore
2.https://www.diva-portal.org/smash/get/diva2:1386720/FULLTEXT01.pdf
3.https://en.wikipedia.org/wiki/Decision_tree_learning

APPENDIX

```
import numpy as np

from flask import Flask, request, jsonify, render_template

import pickle

app = Flask(__name__)

model= pickle.load(open('mining.pkl','rb'))

@app.route('/')

def home():

    return render_template('index.html')

@app.route('/about')

def about():

    return  render_template("about.html")
```

```
@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[x for x in request.form.values()]]
    prediction = model.predict(x_test)
    pred=prediction[0]
    print(prediction)
    return render_template('index.html', prediction_text='Predicted Quality:{}'.format(pred))
if name == "__main__":
    app.run(debug=True)
```