Department of Mining Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BANARAS HINDU UNIVERSITY) VARANASI

EXPLORATORY PROJECT

MINING COMPONENTS FAILURE OPTIMZATION USING NAÏVE BAYES METHOD

Submitted By:

SHUBHAM SAMRAT
19155092

AE - 112

Mining – B. Tech

Supervised By:

Dr. Suprakash Gupta

Professor

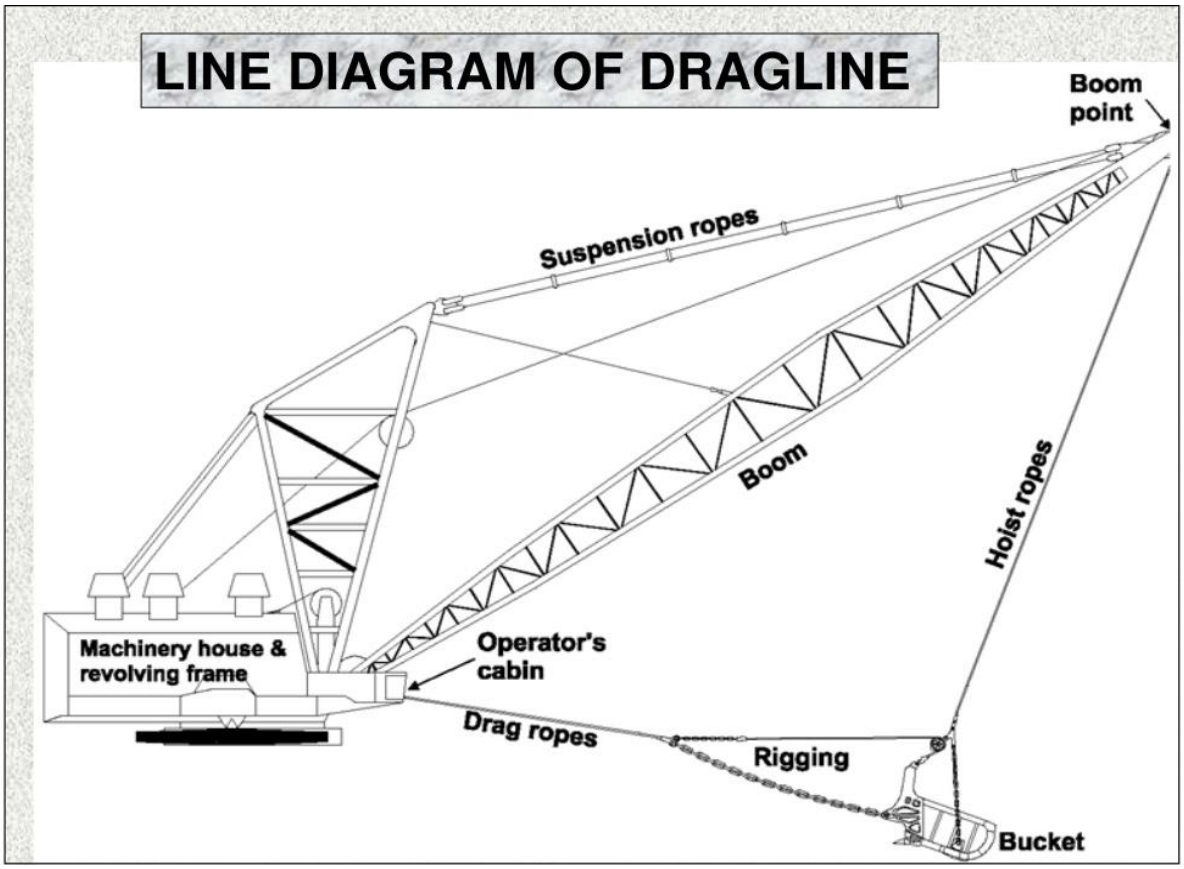
Department of Mining Engineering

INTRODUCTION

DRAGLINE: The dragline is a typical combined cyclic excavator and material carrier since it both excavates material and dumps it without the use of trucks or conveyor belts. The dragline sits above the waste or overburden block, usually 50 m or so wide, on the highwall side and excavates the material in front of itself, to dump it on the low wall or spoil side of the strip to uncover the coal seam below it.

There are three types of Draglines:

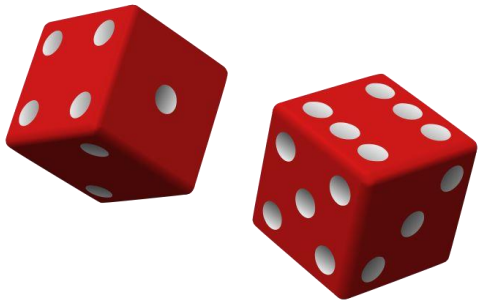
1- Truck mounted
2- Crawler mounted
3- Walking



APPLICATIONS OF DRAGLINE IN MINING: Dragline is a one of the surface mining equipment which is used to excavate materials and is so designed so that it can excavate below the level of the machine also. Dragline working can be divided into two parts: Digging and Walking. Among them walking is a steady process on which the mine design team has little control. Almost all walking draglines take a step of approximately 2 m within a time period of 0.75-1 min. Dragline used in surface mining is heavy equipment. They are mostly built on-site for strip mining operations to remove overburden and coal and they are largest mobile land machine ever built. It consists of large bucket which is suspended from a boom with wire ropes. The bucket is maneuvered by means of ropes and chains. The hoist rope is powered by electric motors and drag rope is used to draw bucket assembly horizontally. Various operations of the bucket for the desired purposes are controlled by maneuvering of the hoist and drag ropes. The dump operation of the dragline consists of positioning the bucket above the material to be excavated and then lowering it so as to drag it on the surface of the material. The bucket is then lifted by the hoist rope and a swing operation is then performed to 2 move the bucket to the place where the material is to be dumped, drag rope is then released causing the bucket to tilt and empty.

EFFECT OF FAILURE OF DRAGLINE: Overburden stripping in open cast coal mines is extensively carried out by walking draglines. Draglines' unavailability and unexpected failures result in delayed productions and increased maintenance and operating costs. Therefore, achieving high availability of draglines plays a crucial role for increasing economic feasibility of mining projects.

Naïve Bayes Classifier



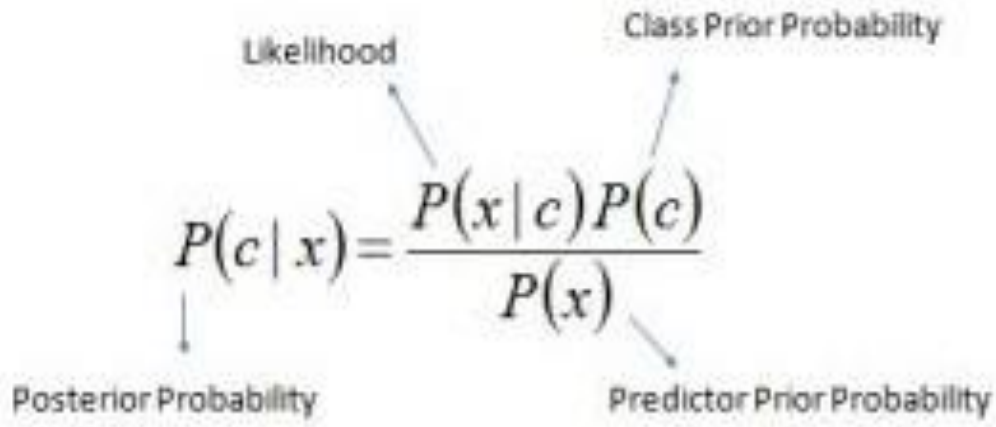
Naive Bayes – Introduction:

Naive Bayes - (Sometime aka Stupid Bayes :))

- Classification technique based on Bayes' Theorem
- With "naive" assumption of independence among predictors.
- Easy to build
- Particularly useful for very large data sets
- Known to outperform even highly sophisticated classification methods
- e.g. Earlier method for spam detection

Introduction - Bayes theorem:

- P(c|x) - the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) - the prior probability of class.
- P(x|c) - is the likelihood which is the probability of predictor given class.
- P(x) - is the prior probability of predictor.



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

How Naive Bayes algorithm works

- We are training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing).
- Now, we need to classify whether players will play or not based on weather condition.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

- Convert the data set into a frequency table
- Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.
- Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement being correct?

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

- P (Yes | Sunny) = P (Sunny | Yes) * P(Yes) / P (Sunny)

  Here we have:

- P (Sunny |Yes) = 3/9 = 0.33,
- P(Sunny) = 5/14 = 0.36,
- P(Yes)= 9/14 = 0.64
- Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, (**high probability**)

- P (No | Sunny) = P (Sunny | No) * P(No) / P (Sunny)

  Here we have:

- P (Sunny |No) = 2/5 = 0.4,
- P(Sunny) = 5/14 = 0.36,
- P(No)= 5/14 = 0.36
- Now, P (No | Sunny) = 0.4 * 0.36 / 0.36 = 0.40, (**low probability**)

Hence this was the working of the Naïve Bayes Algorithm.

Tips to improve the Naive Bayes Model:

If continuous features do not have normal distribution,

- we should use transformation or different methods to convert

- If test data set has zero frequency issue,
  - apply smoothing techniques “Laplace smoothing”

- Remove correlated features,
  - as the highly correlated features are voted twice in the model
  - and it can lead to over inflating importance.

- Naive Bayes classifier has limited options for parameter tuning

- Can’t be ensembled - because there is no variance to reduce

Variants

Gaussian:

- It is used in classification and it assumes that features follow a normal distribution.

Multinomial:

- It is used for discrete counts.
- Implements the naive Bayes algorithm for multinomially distributed data
- It is one of the two classic naive Bayes variants used in text classification

Bernoulli:

- The binomial model is useful if your feature vectors are binary (i.e. zeros and ones).
- One application would be text classification with 'bag of words' model
- where the 1s & 0s are "word occurs in the document"
- and "word does not occur in the document" respectively.

http://scikit-learn.org/stable/modules/naive_bayes.html

Applications:

1. Real time Prediction:

- Naive Bayes is an eager learning classifier and it is sure fast.

  Thus, it could be used for making predictions in real time.

2. Multi class Prediction:

- Well known for multi class prediction feature.

1. **Text classification/ Spam Filtering/ Sentiment Analysis:**

- Mostly used in text classification
- Have higher success rate as compared to other algorithms.
- Widely used in Spam filtering (identify spam e-mail)
- and Sentiment Analysis

2. Recommendation System:

- Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning

and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

Further references

- http://scikit-learn.org/stable/modules/naive_bayes.html
- https://en.wikipedia.org/wiki/Bayes%27_theorem
- https://en.wikipedia.org/wiki/Naive_Bayes_classifier

NEED FOR NAÏVE BAYES CLASSIFICATION FOR A DRAGLINE:

Since dragline is the biggest mining machine ever being used in mines. Its failure puts a stop in production which is highly uneconomic for the mining projects. That's why we need to do naïve bayes classification of its properties to get a summary of its working and failure stats by observing that stats one can be ready with proper maintenance to prevent the hindrance in production that would be caused due to failure of Dragline.

METHODOLOGY:

HMR (Hour meter reading) and remarks were provided for the subsystems of the dragline specifically dragging system, hoist and rigging. Thus, the classification has to be done using the Naïve Bayes Classifier using the inbuilt software package of sklearn of Python. This is just the prediction method which is based on the previous HMR and remarks of the previous breakdown of the machine, we were predicting

whether in the future time the breakdown can be due to which remark so that particular Maintenance Team can be prepared in advance to tackle the scenario and ensure the smooth functioning of the Dragline.

Now here we have gathered data of an operating dragline for the span of two years and we have divided it's working in 3 different sub parts.

They are:

1- Rigging
2- Hoist
3- Dragging System

So now we will do the Naïve bayes classification of these sub parts separately using python software for the obtained data set.

DRAGGING SYSTEM

Below you can see the working of the model:

In [1]:
```
import pandas as pd
df = pd.read_csv("dragging system.csv")
df.head()
```

Out[1]:

	Sr. No.	HMR	Remarks	Unnamed: 3
0	1	99051.0	drag_chain_slip	NaN
1	2	99526.0	drag_chain_slip	NaN
2	3	100123.5	drag_rope	NaN
3	4	100141.0	drag_rope	NaN
4	5	100183.0	drag_rope	NaN

In [2]:
```
df.drop(['Sr. No.','Unnamed: 3'],axis='columns', inplace=True)
df.head()
```

Out[2]:

	HMR	Remarks
0	99051.0	drag_chain_slip
1	99526.0	drag_chain_slip
2	100123.5	drag_rope
3	100141.0	drag_rope
4	100183.0	drag_rope

In [3]:
```
dummies= pd.get_dummies(df.Remarks)
dummies
```

Out[3]:

	drag chain	drag chain link broken	drag chain problem	drag drum drive break	drag rope out from pulley	drag socket pin lock broken	drag walk problem	drag_chain	drag_chain_slip	drag_rope	drag_rope_out
0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
265	0	0	0	0	0	0	0	0	0	0	1
266	0	0	0	0	0	0	0	0	0	0	1
267	0	0	0	0	0	0	0	0	0	0	1
268	0	0	0	0	0	0	0	0	0	0	1
269	0	0	0	0	0	0	0	0	0	0	1

270 rows × 11 columns

270 rows × 11 columns

In [4]:
```
merged = pd.concat([df,dummies],axis='columns')
merged
```

Out[4]:

	HMR	Remarks	drag chain	drag chain link broken	drag chain problem	drag drum drive break	drag rope out from pulley	drag socket pin lock broken	drag walk problem	drag_chain	drag_chain_slip	drag_rope	drag_rope_out
0	99051.0	drag_chain_slip	0	0	0	0	0	0	0	0	1	0	0
1	99526.0	drag_chain_slip	0	0	0	0	0	0	0	0	1	0	0
2	100123.5	drag_rope	0	0	0	0	0	0	0	0	0	1	0
3	100141.0	drag_rope	0	0	0	0	0	0	0	0	0	1	0
4	100183.0	drag_rope	0	0	0	0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	173720.5	drag_rope_out	0	0	0	0	0	0	0	0	0	0	1
266	173880.5	drag_rope_out	0	0	0	0	0	0	0	0	0	0	1
267	173990.0	drag_rope_out	0	0	0	0	0	0	0	0	0	0	1
268	174073.5	drag_rope_out	0	0	0	0	0	0	0	0	0	0	1
269	174407.0	drag_rope_out	0	0	0	0	0	0	0	0	0	0	1

270 rows × 13 columns

270 rows × 13 columns

In [10]:
```
final = merged.drop(['Remarks','drag chain','drag chain link broken','drag chain problem','drag drum drive break'],axis='columns')
final
```

Out[10]:

	HMR	drag rope out from pulley	drag socket pin lock broken	drag walk problem	drag_chain	drag_chain_slip	drag_rope	drag_rope_out
0	99051.0	0	0	0	0	1	0	0
1	99526.0	0	0	0	0	1	0	0
2	100123.5	0	0	0	0	0	1	0
3	100141.0	0	0	0	0	0	1	0
4	100183.0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...
265	173720.5	0	0	0	0	0	0	1
266	173880.5	0	0	0	0	0	0	1
267	173990.0	0	0	0	0	0	0	1
268	174073.5	0	0	0	0	0	0	1
269	174407.0	0	0	0	0	0	0	1

270 rows × 8 columns

In [23]:
```
inputs = final.drop('drag_rope',axis='columns')
target= final.drag_rope
```

In [12]:
```
inputs.columns[inputs.isna().any()]
```

Out[12]:
```
Index([], dtype='object')
```

270 rows × 8 columns

In [23]:
```
inputs = final.drop('drag_rope',axis='columns')
target= final.drag_rope
```

In [12]:
```
inputs.columns[inputs.isna().any()]
```

Out[12]:
```
Index([], dtype='object')
```

In [13]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.2)
```

In [14]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

In [15]:
```
model.fit(X_train,y_train)
```

Out[15]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [16]:
```
model.score(X_test,y_test)
```

Out[16]:
```
0.9814814814814815
```

In [17]:
```
X_test[0:10]
```

Out[17]:

	HMR	drag rope out from pulley	drag socket pin lock broken	drag walk problem	drag_chain	drag_chain_slip	drag_rope_out
140	145462.0	0	0	0	0	1	0
217	159096.0	0	0	0	1	0	0
56	116291.0	0	0	0	0	1	0
156	148400.0	0	0	0	0	1	0
252	169294.5	0	0	0	0	0	0
219	159814.5	0	0	0	1	0	0
237	165060.0	0	0	0	0	0	1
88	127146.0	0	0	0	0	1	0
267	173990.0	0	0	0	0	0	1
269	174407.0	0	0	0	0	0	1

In [18]:
```
y_test[0:10]
```

Out[18]:
```
140    0
217    0
56     0
156    0
252    0
219    0
237    0
88     0
267    0
269    0
Name: drag_rope, dtype: uint8
```

In [19]:
```
model.predict(X_test[0:10])
```

Out[19]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=uint8)
```

RESULT:

I have made the model for the dragging system of the dragline. So, that based on the previous data my model can classify the future failure of this subsystem with the remark so that only that particular section of this subsystem of the dragline can be repaired and ensure the smooth functioning of the dragline.

Here the features for my model is only **HMR** and labels are **drag_rope**, **drag_chain_slip**, **drag_chain** and **drag_rope_out** based on that by using the Gaussian Naïve bayes Classifier of python I have predicted the label which is going to be failed in the future time.

Below in the given code I have predicted for the Label: **drag_rope**. Based on features (HMR) my model has predicted when again in the future the same problem of the drag_rope is going to occur.

The Model worked very well with a score of **88.14%** and predicted whether in the future the problem will be caused due to the drag_rope.

Hoist

Below you can see the working of the model for the hoist of the dragline.

In [1]:
```
import pandas as pd
df = pd.read_csv("hoist.csv")
df.head()
```

Out[1]:

	Sr. No.	HMR	Remarks
0	1	103678.5	hoist_drum
1	2	105157.5	hoist_drum
2	3	110937.0	hoist_drum
3	4	111304.0	hoist_drum
4	5	111862.0	hoist_problem

In [2]:
```
df.drop(['Sr. No.'],axis='columns', inplace=True)
df.head()
```

Out[2]:

	HMR	Remarks
0	103678.5	hoist_drum
1	105157.5	hoist_drum
2	110937.0	hoist_drum
3	111304.0	hoist_drum
4	111862.0	hoist_problem

In [3]:
```
dummies= pd.get_dummies(df.Remarks)
dummies
```

Out[3]:

	hoist_chain	hoist_drum	hoist_problem	hoist_rope_broken	hoist_tripping
0	0	1	0	0	0
1	0	1	0	0	0
2	0	1	0	0	0
3	0	1	0	0	0
4	0	0	1	0	0
...	...	...	...	...	...
58	1	0	0	0	0
59	1	0	0	0	0
60	1	0	0	0	0
61	1	0	0	0	0
62	1	0	0	0	0

63 rows × 5 columns

In [4]:
```
merged = pd.concat([df,dummies],axis='columns')
merged
```

Out[4]:

	HMR	Remarks	hoist_chain	hoist_drum	hoist_problem	hoist_rope_broken	hoist_tripping
0	103678.5	hoist_drum	0	1	0	0	0
1	105157.5	hoist_drum	0	1	0	0	0
2	110937.0	hoist_drum	0	1	0	0	0
3	111304.0	hoist_drum	0	1	0	0	0
4	111862.0	hoist_problem	0	0	1	0	0
...	...	...	...	...	...	...	...
58	166498.0	hoist_chain	1	0	0	0	0
59	172186.0	hoist_chain	1	0	0	0	0
60	173685.5	hoist_chain	1	0	0	0	0
61	174230.0	hoist_chain	1	0	0	0	0
62	174545.5	hoist_chain	1	0	0	0	0

63 rows × 7 columns

63 rows × 7 columns

In [5]:
```
final = merged.drop(['Remarks'],axis='columns')
final
```

Out[5]:

	HMR	hoist_chain	hoist_drum	hoist_problem	hoist_rope_broken	hoist_tripping
0	103678.5	0	1	0	0	0
1	105157.5	0	1	0	0	0
2	110937.0	0	1	0	0	0
3	111304.0	0	1	0	0	0
4	111862.0	0	0	1	0	0
...	...	...	...	...	...	...
58	166498.0	1	0	0	0	0
59	172186.0	1	0	0	0	0
60	173685.5	1	0	0	0	0
61	174230.0	1	0	0	0	0
62	174545.5	1	0	0	0	0

63 rows × 6 columns

In [6]:
```
inputs = final.drop('hoist_rope_broken',axis='columns')
target= final.hoist_rope_broken
```

In [7]:
```
inputs.columns[inputs.isna().any()]
```

Out[7]:
```
Index([], dtype='object')
```

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.2)
```

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.2)
```

In [9]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

In [10]:
```
model.fit(X_train,y_train)
```

Out[10]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [11]:
```
model.score(X_test,y_test)
```

Out[11]:
```
0.8461538461538461
```

In [12]:
```
X_test[0:10]
```

Out[12]:

	HMR	hoist_chain	hoist_drum	hoist_problem	hoist_tripping
28	140404.5	0	0	0	1
0	103678.5	0	1	0	0
19	138051.0	0	0	0	0
32	141618.0	0	0	0	1
35	142431.5	0	0	0	1
2	110937.0	0	1	0	0
31	141234.5	0	0	0	1
44	155214.0	0	0	0	1
12	127797.5	0	0	0	0
45	155268.0	0	0	0	1

In [13]:
```
y_test[0:10]
```

Out[13]:
```
28    0
0     0
19    1
32    0
35    0
2     0
31    0
44    0
12    1
45    0
Name: hoist_rope_broken, dtype: uint8
```

In [14]:
```
model.predict(X_test[0:10])
```

Out[14]:
```
array([0, 0, 0, 0, 0, 1, 0, 0, 1, 0], dtype=uint8)
```

RESULT:

Above model is for the hoist of the dragline. So, based on the previous data my model can classify the future failure of this subsystem with the remark so that only that particular section of this subsystem of the dragline can be repaired and ensure the smooth functioning of the dragline.

Here the features for my model is only **HMR** and labels are **hoist_chain**, **hoist_drum**, **hoist_problem** and **hoist_tripping** based on that by using the Gaussian Naïve bayes Classifier of python I have predicted the label which is going to be failed in the future time.

Below in the given code I have predicted for the Label: **hoist_rope_broken**. Based on features (HMR) my model has predicted when again in the future the same problem of the drag_rope is going to occur.

The Model worked very well with a score of **84.61%** and predicted whether in the future the problem will be caused due to the hoist_rope_broken.

RIGGING:

Below you can see the working of the model for rigging subsystem of the dragline.

In [1]:
```
import pandas as pd
df = pd.read_csv("Rigging.csv")
df.head()
```

Out[1]:

	Sr. No.	HMR	Remarks
0	1	98678.5	drum_rope_broke
1	2	98731.0	drum_rope_broke
2	3	99039.5	drum_rope_broke
3	4	99703.0	drum_rope_broke
4	5	99923.5	drum_rope_broke

In [2]:
```
df.drop(['Sr. No.'],axis='columns', inplace=True)
df.head()
```

Out[2]:

	HMR	Remarks
0	98678.5	drum_rope_broke
1	98731.0	drum_rope_broke
2	99039.5	drum_rope_broke
3	99703.0	drum_rope_broke
4	99923.5	drum_rope_broke

In [3]:
```
dummies= pd.get_dummies(df.Remarks)
dummies
```

Out[3]:

	drum_pulley_pinout	drum_rope_broke	drum_rope_slip	dump_socket_pinout
0	0	1	0	0
1	0	1	0	0
2	0	1	0	0
3	0	1	0	0
4	0	1	0	0
...	...	...	...	...
264	0	0	1	0
265	0	0	1	0
266	0	0	1	0
267	0	0	1	0
268	0	0	1	0

269 rows × 4 columns

In [4]:
```
merged = pd.concat([df,dummies],axis='columns')
merged
```

Out[4]:

	HMR	Remarks	drum_pulley_pinout	drum_rope_broke	drum_rope_slip	dump_socket_pinout
0	98678.5	drum_rope_broke	0	1	0	0
1	98731.0	drum_rope_broke	0	1	0	0
2	99039.5	drum_rope_broke	0	1	0	0
3	99703.0	drum_rope_broke	0	1	0	0
4	99923.5	drum_rope_broke	0	1	0	0
...	...	...	...	...	...	...
264	172028.0	drum_rope_slip	0	0	1	0
265	173073.5	drum_rope_slip	0	0	1	0
266	173482.0	drum_rope_slip	0	0	1	0
267	173880.5	drum_rope_slip	0	0	1	0
268	174194.0	drum_rope_slip	0	0	1	0

269 rows × 6 columns

269 rows × 6 columns

In [5]:
```
final = merged.drop(['Remarks'],axis='columns')
final
```

Out[5]:

	HMR	drum_pulley_pinout	drum_rope_broke	drum_rope_slip	dump_socket_pinout
0	98678.5	0	1	0	0
1	98731.0	0	1	0	0
2	99039.5	0	1	0	0
3	99703.0	0	1	0	0
4	99923.5	0	1	0	0
...	...	...	...	...	...
264	172028.0	0	0	1	0
265	173073.5	0	0	1	0
266	173482.0	0	0	1	0
267	173880.5	0	0	1	0
268	174194.0	0	0	1	0

269 rows × 5 columns

In [6]:
```
inputs = final.drop('drum_rope_broke',axis='columns')
target= final.drum_rope_broke
```

In [7]:
```
inputs.columns[inputs.isna().any()]
```

Out[7]:
```
Index([], dtype='object')
```

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.2)
```

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.2)
```

In [9]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

In [10]:
```
model.fit(X_train,y_train)
```

Out[10]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [11]:
```
model.score(X_test,y_test)
```

Out[11]:
```
0.9074074074074074
```

In [12]:
```
X_test[0:10]
```

Out[12]:

	HMR	drum_pulley_pinout	drum_rope_slip	dump_socket_pinout
251	167288.5	0	1	0
183	150959.0	1	0	0
37	112523.0	0	0	0
80	120950.0	0	0	1
215	158201.5	0	1	0
182	150662.0	1	0	0
67	117954.0	0	0	1
242	165412.0	0	1	0
52	115902.5	0	0	0
25	107629.0	0	0	0

In [13]:
```
y_test[0:10]
```

Out[13]:
```
251    0
183    0
37     1
80     0
215    0
182    0
67     0
242    0
52     1
25     1
Name: drum_rope_broke, dtype: uint8
```

In [14]:
```
model.predict(X_test[0:10])
```

Out[14]:
```
array([0, 0, 1, 0, 0, 0, 1, 0, 1, 1], dtype=uint8)
```

RESULT:

Above model is for the Rigging of the dragline. So, based on the previous data, my model can classify the future failure of this subsystem with the remark so that only that particular section of this subsystem of the dragline can be repaired and ensure the smooth functioning of the dragline.

Here the features for my model is only **HMR** and labels are **drum_rope_broke**, **drum_socket_pin_out**, **drum_pulley_pin_out** and **drum_rope_slip** based on that by using the Gaussian Naïve bayes Classifier of python I have predicted the label which is going to be failed in the future time.

Below in the given code I have predicted for the Label: **drum_rope_broke**. Based on features (HMR) my model has predicted when again in the future the same problem of the drag_rope is going to occur.

The Model worked very well with a score of **90.71%** and predicted whether in the future the problem will be caused due to the drum_rope_broke.

Conclusion:

The naïve Bayes algorithm is a popular algorithm that handles categorical or non-continuous data but has challenge in its methodology which attribute is attribute independent assumption and it usually being criticized for that methodology, this work tends to

address it in order to improve its performance efficiency and accuracy. So, this classifier is very useful to classify the situations like the above and prepare us for the future. This analysis should be the essential part of the dragline system for increasing the availability and proper maintenance to improve the utilization of the dragline system in opencast mining projects. To improve this optimization of the dragline, it is necessary to remove the failure causes at every steps of the life cycle such as design plan, construction of machine, operation and maintenance.

CREDITS

SUPRAKASH Sir (‘Supervising’)

DEEPAK Sir (‘Assisting’)

Python (‘Programming Language’)

Jupyter Notebook

Thanks