Project 3

Parham Hassan & Banafshe Monfared

2022-07-15

Background The Canada Energy Regulator has a mandate to protect people and the environment during construction, operation, and abandonment of oil and gas pipelines and associated facilities. Despite its best efforts in prevention and mitigation, sometimes incidents that lead to adverse effects to people and the environment can happen. In the past 12 years there have been 723 incidents that involved release of substance.

Data Description The CER provides an open dataset of 723 incidents, from 2008 to 2020.

Main Question For this case study, we would like to understand what geographical and meteorological factors are associated with an incident that involves a release of substance. The dependent variable is probability of a geographical location having an incident and the independent variables are geographical (population density, type of land use, and other variables teams find relevant to include) and meteorological variables.

Introduction

We start by loading relevant packages and loading the Data :

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.1.3
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## Warning: package 'forcats' was built under R version 4.1.3
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data <- read_csv("C:\\Users\\Banafshe\\Desktop\\projectData1.csv")
```

```
## Rows: 1624 Columns: 16
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (13): Incident.Number, Reported.Date, Nearest.Populated.Centre, Province...
## dbl  (3): Latitude, Longitude, Year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data <- data%>%
  rename(SubstanceRelease = `Substance release`)
glimpse(data)
```

```
## Rows: 1,624
## Columns: 16
## $ Incident.Number                  <chr> "INC2007-097", "INC2008-001", "INC200~
## $ Reported.Date                    <chr> "1/2/2008", "1/2/2008", "1/23/2008", ~
## $ Nearest.Populated.Centre         <chr> "Grande Prairie", "Cromer", "Cromer",~
## $ Province                         <chr> "Alberta", "Manitoba", "Manitoba", "B~
## $ Company                          <chr> "Alliance Pipeline Ltd.", "Enbridge P~
## $ Status                           <chr> "Closed", "Closed", "Closed", "Closed~
## $ Latitude                         <dbl> 54.84000, 49.73135, 49.73135, 58.0120~
## $ Longitude                        <dbl> -118.65000, -101.23557, -101.23557, -~
## $ Approximate.Volume.Released..m3. <chr> "Not Provided", "8", "100", "Not Prov~
## $ Substance                        <chr> "Natural Gas - Sweet", "Crude Oil - S~
## $ Release.Type                     <chr> "Gas", "Liquid", "Liquid", "Gas", "Mi~
## $ Significant                      <chr> "No", "No", "No", "No", "Yes", "No", ~
## $ Year                             <dbl> 2008, 2008, 2008, 2008, 2008, 2008, 2~
## $ What.Happened                    <chr> "Corrosion and Cracking", "Corrosion ~
## $ Why.It.Happened                  <chr> "Maintenance", "Maintenance", "Mainte~
## $ SubstanceRelease                 <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Y~
```

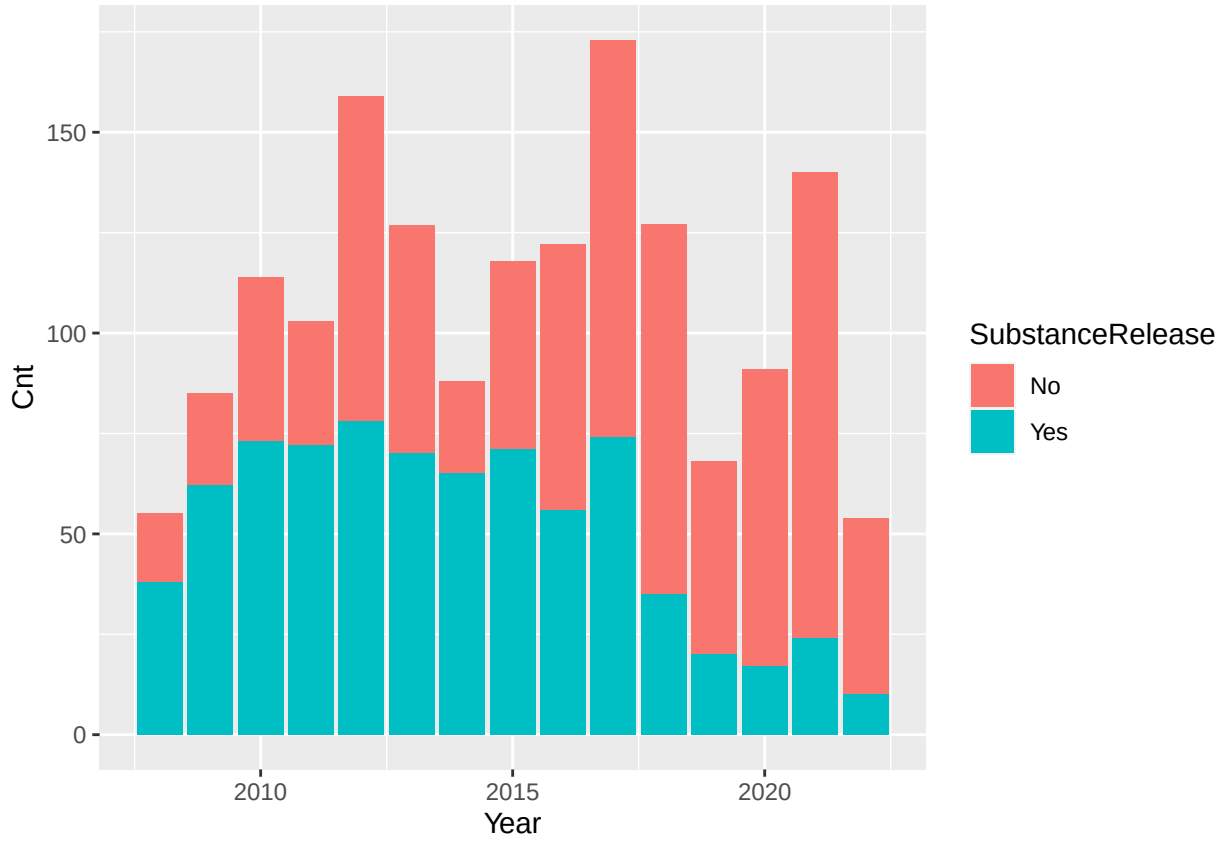
After that , we are going to see the number of yearly substance releases spliting them by yes or no .

```
t<-data%>%
  group_by(Year,SubstanceRelease)%>%
  summarize(Cnt = n())
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

for better comprehension ,we will plot our outcome :

```
t%>%
  ggplot(aes(x=Year, y=Cnt,fill=SubstanceRelease)) +
  geom_bar(stat="identity")
```




```
t<-pivot_wider(
  t,
  names_from = SubstanceRelease,
  values_from = `Cnt`,
)
```

this plot shows us the yearly substance releases from 2008 to 2020,on average we can see a gradual growth from 2008 until 2017 and then a noticeable reduction from 2017 until 2020 .

In this section we dive a little bit deeper into our data , we want to see the substance release in each Province:

```
t2<-data%>%
  group_by(Province,SubstanceRelease)%>%
  summarize(Cnt = n())%>%
  arrange(desc(Cnt))
```

```
## `summarise()` has grouped output by 'Province'. You can override using the
## `.groups` argument.
```

```
t2
```

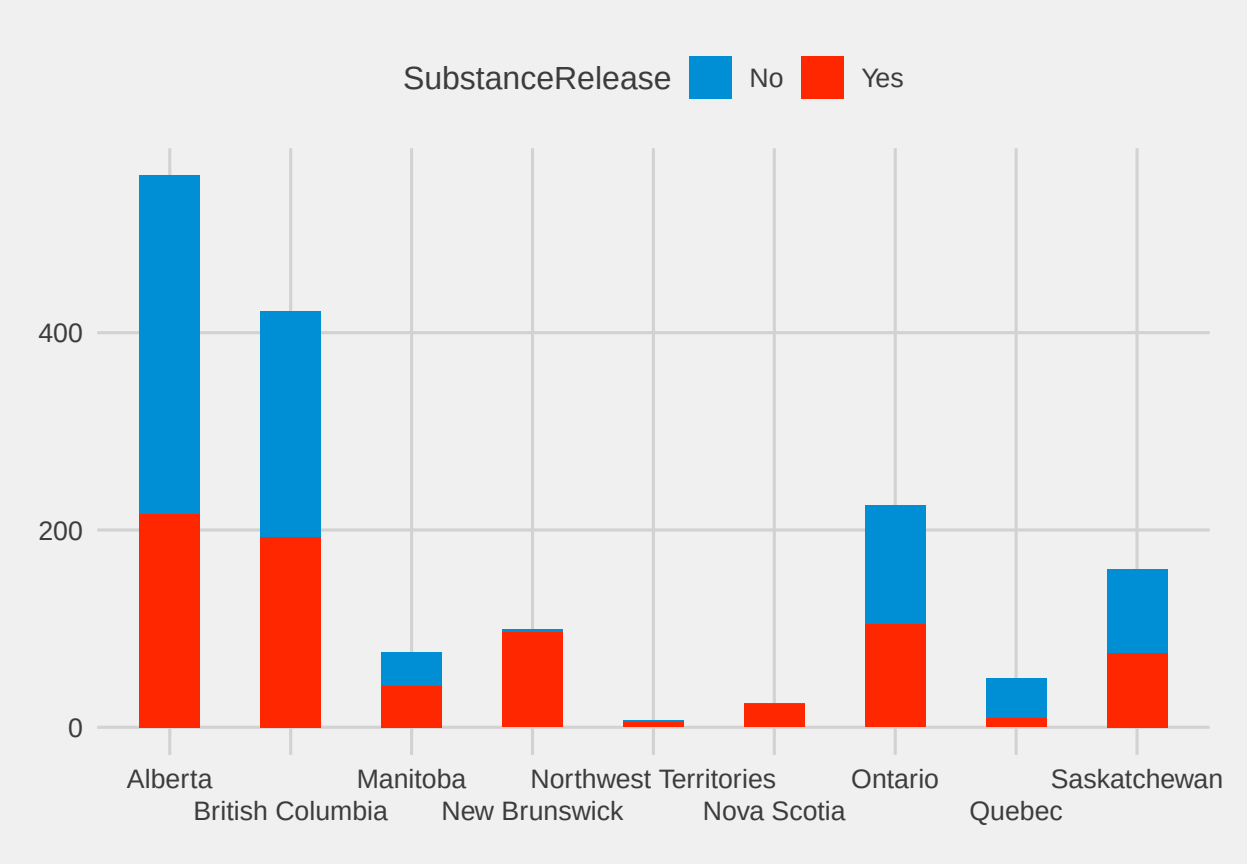
```
## # A tibble: 18 x 3
## # Groups:   Province [9]
##    Province              SubstanceRelease   Cnt
##    <chr>                 <chr>            <int>
##  1 Alberta               No                 343
##  2 British Columbia      No                 229
##  3 Alberta               Yes                216
##  4 British Columbia      Yes                193
##  5 Ontario               No                 121
##  6 Ontario               Yes                104
##  7 New Brunswick         Yes                 97
##  8 Saskatchewan          No                  85
##  9 Saskatchewan          Yes                 75
## 10 Manitoba              Yes                 42
## 11 Quebec                No                  41
## 12 Manitoba              No                  34
## 13 Nova Scotia           Yes                 24
## 14 Quebec                Yes                  9
## 15 Northwest Territories Yes                  5
## 16 New Brunswick         No                   3
## 17 Northwest Territories No                   2
## 18 Nova Scotia           No                   1
```

also for better comparison , we have sorted the number of releases in order to have a better prospective of the data .

for better comprehension ,we will plot our outcome :

```
library(ggplot2)

t2%>%
  ggplot(aes(x=Province, y=Cnt,fill=SubstanceRelease)) +
  geom_bar(stat="identity",width = 0.5)+
  scale_x_discrete(guide = guide_axis(n.dodge=2))+
   labs( y = "Number Of Releases", x = "Province")+
  theme_fivethirtyeight() +
  scale_fill_fivethirtyeight() +
  theme(legend.position = "top")
```



As illustrated above , we can see the probability of substance releases in some provinces vividly.for example in New Brunswick since the portion of substance release being ***Yes*** is much bigger than substance release being ***No***;that means we have a higher chance of predicting the substance release being ***Yes***.

On the contrary,prediction of this is more difficult in provinces such as Ontario since the portions for substance releases happening or not is approximately equal.

In conclusion , some of these provinces can help of with the prediction of substance releases.

for our next step,we will have to check the ***Companies*** and the number of releases:

```
t3<-data%>%
  group_by(Company,SubstanceRelease)%>%
  summarize(Cnt = n())
```

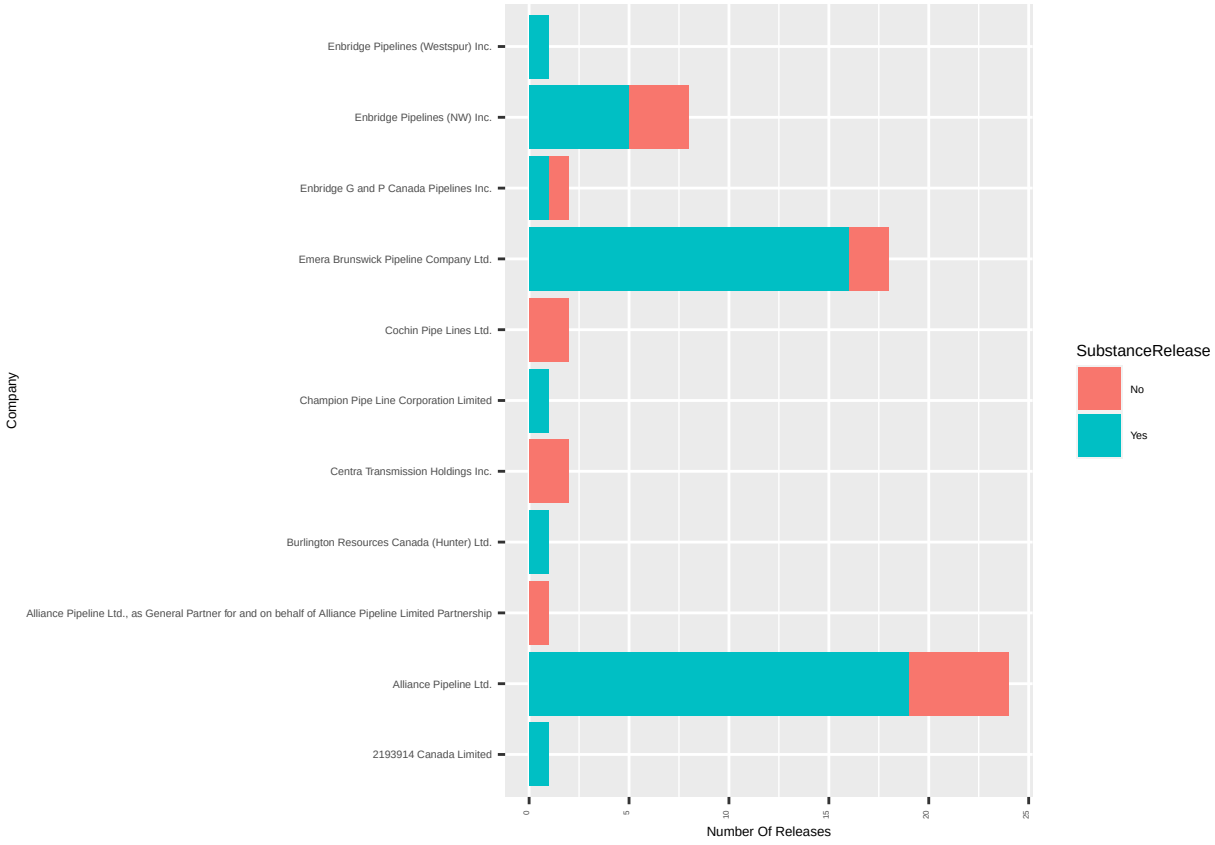
```
## `summarise()` has grouped output by 'Company'. You can override using the
## `.groups` argument.
```

```
t3
```

```
## # A tibble: 74 x 3
## # Groups:   Company [50]
##    Company                                SubstanceRelease   Cnt
##    <chr>                                  <chr>            <int>
##  1 2193914 Canada Limited                 Yes                  1
##  2 Alliance Pipeline Ltd.                 No                   5
##  3 Alliance Pipeline Ltd.                 Yes                 19
```

```
##  4 Alliance Pipeline Ltd., as General Partner for and on~ No                 1
##  5 Burlington Resources Canada (Hunter) Ltd.              Yes                1
##  6 Centra Transmission Holdings Inc.                      No                 2
##  7 Champion Pipe Line Corporation Limited                 Yes                1
##  8 Cochin Pipe Lines Ltd.                                 No                 2
##  9 Emera Brunswick Pipeline Company Ltd.                  No                 2
## 10 Emera Brunswick Pipeline Company Ltd.                  Yes               16
## # ... with 64 more rows
```

```
t3[1:15,]%>%
  ggplot(aes(x=Company, y=Cnt,fill=SubstanceRelease)) +
  geom_bar(stat="identity")+
#scale_x_discrete(guide = guide_axis(n.dodge=))
  labs( y = "Number Of Releases", x = "Company")+
    theme(axis.text.y = element_text(size = 4),
        axis.text.x = element_text(angle=90, hjust=1,size=2.98,vjust = 0),
        axis.title = element_text(size = 5),legend.title  = element_text(size = 6),legend.text = elemen
  coord_flip()
```



As it can be seen from the plot , just like `Province` , `Company` is a useful variable for the prediction of substance releases .

Next , we take a look at `Status` . this variable has three levels : ***Closed*** , ***Submitted*** and ***Initially Submitted***.

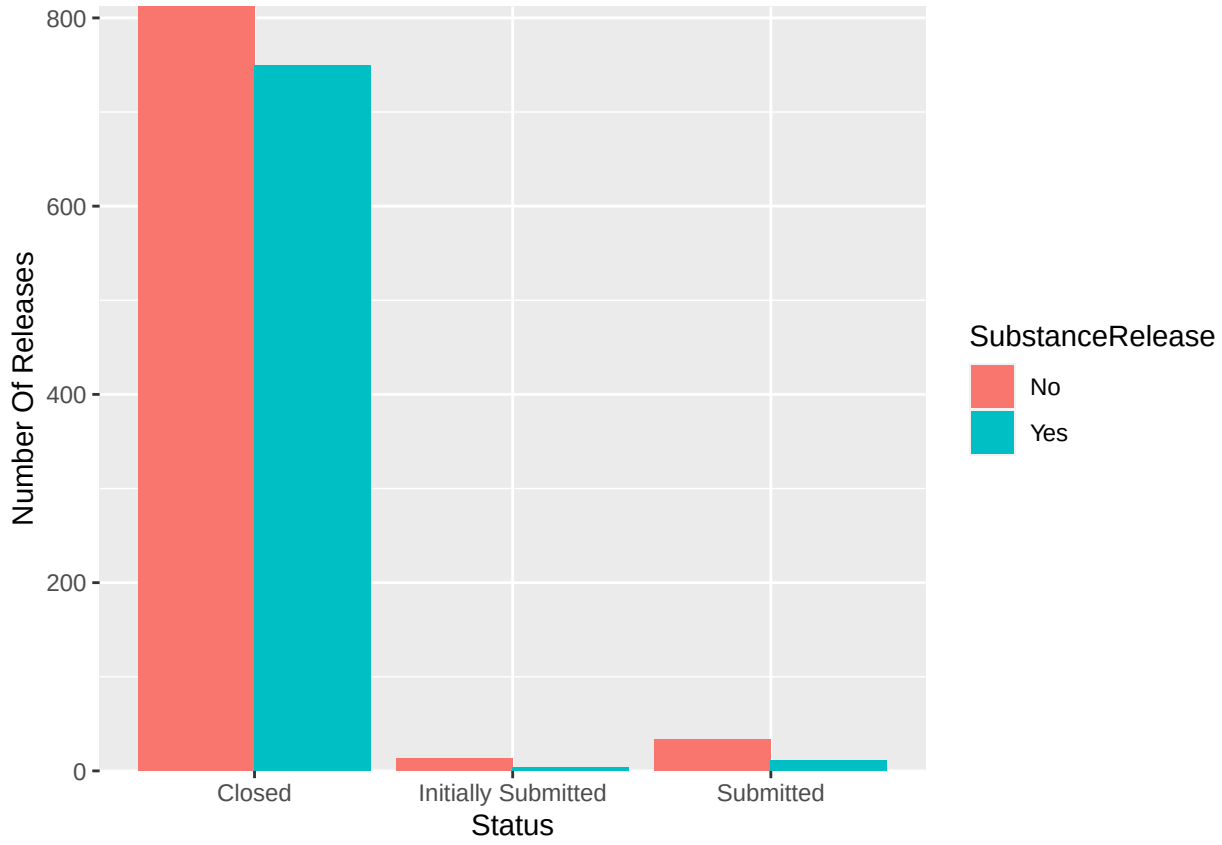
```
t4<-data%>%
  group_by(Status,SubstanceRelease)%>%
  summarize(Cnt = n())%>%
  arrange(desc(Cnt))
```

```
## `summarise()` has grouped output by 'Status'. You can override using the
## `.groups` argument.
```

```
t4
```

```
## # A tibble: 6 x 3
## # Groups:   Status [3]
##   Status              SubstanceRelease   Cnt
##   <chr>               <chr>            <int>
## 1 Closed              No                 812
## 2 Closed              Yes                750
## 3 Submitted           No                  34
## 4 Initially Submitted No                  13
## 5 Submitted           Yes                 11
## 6 Initially Submitted Yes                  4
```

```
t4%>%
  ggplot(aes(x=Status, y=Cnt,fill=SubstanceRelease)) +
  geom_bar(stat="identity",position = position_dodge())+
   scale_y_continuous(expand=c(0,0))+
  labs( y = "Number Of Releases", x = "Status")
```



As you can see , this variable does not help that much in our model . Since the proportion in different levels of Substance releases is not that different in ***Closed*** ,and for the rest the difference is small .

After that , we check the variable `Significant`

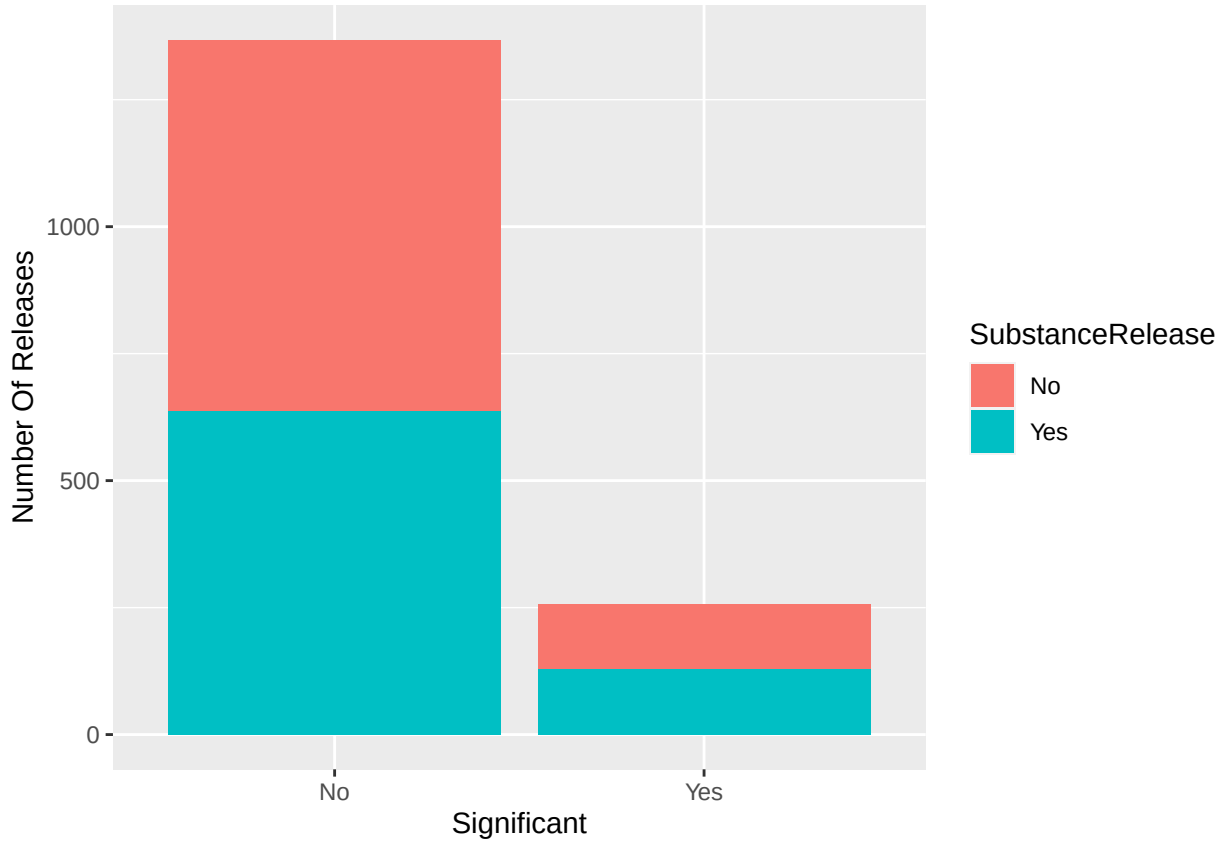
```
t5<-data%>%
  group_by(Significant,SubstanceRelease)%>%
  summarize(Cnt = n())%>%
  arrange(desc(Cnt))
```

```
## `summarise()` has grouped output by 'Significant'. You can override using the
## `.groups` argument.
```

```
t5
```

```
## # A tibble: 4 x 3
## # Groups:   Significant [2]
##   Significant SubstanceRelease   Cnt
##   <chr>       <chr>            <int>
## 1 No          No                 732
## 2 No          Yes                636
## 3 Yes         Yes                129
## 4 Yes         No                 127
```

```
t5%>%
  ggplot(aes(x=Significant, y=Cnt,fill=SubstanceRelease)) +
  geom_bar(stat="identity")+
  labs( y = "Number Of Releases", x = "Significant")
```



As you can observe from above , the proportions between substance release happening or not is equaly distributed in this variable .So this variable is not gonna be very useful.

Next, we have to tackle `Release.Type` .this variable has four levels which are ***Not Applicable***, ***Gas***, ***liquid*** and ***Miscellaneous*** .

```
t6<-data%>%
  group_by(Release.Type,SubstanceRelease)%>%
  summarize(Cnt = n())%>%
  arrange(desc(Cnt))
```

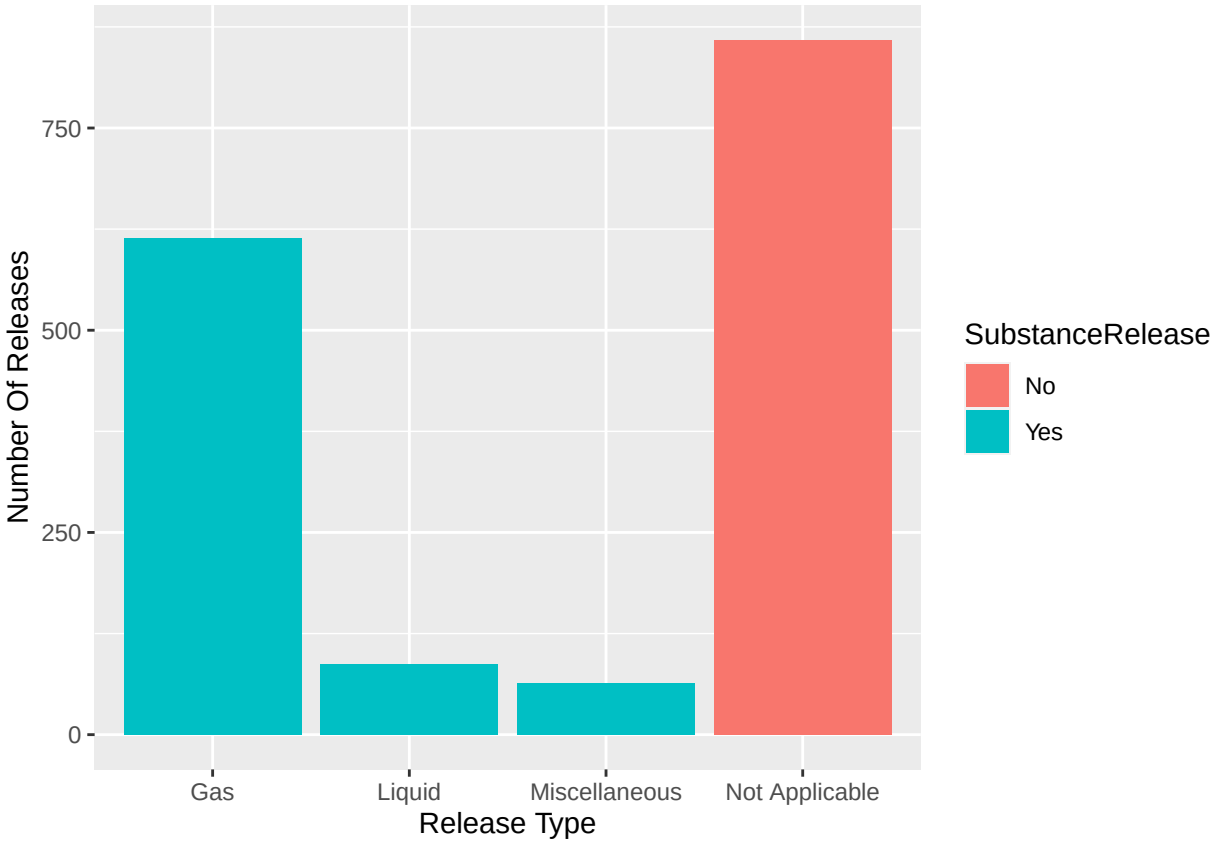
```
## `summarise()` has grouped output by 'Release.Type'. You can override using the
## `.groups` argument.
```

```
t6
```

```
## # A tibble: 4 x 3
## # Groups:   Release.Type [4]
##   Release.Type   SubstanceRelease   Cnt
##   <chr>          <chr>            <int>
```

```
## 1 Not Applicable No                859
## 2 Gas            Yes               614
## 3 Liquid         Yes                87
## 4 Miscellaneous  Yes                64
```

```
t6%>%
  ggplot(aes(x=Release.Type, y=Cnt,fill=SubstanceRelease)) +
  geom_bar(stat="identity")+
   labs( y = "Number Of Releases", x = "Release Type")
```



Other than the release type ***Not Applicable*** ,substance release always happen is rest of them .this give us the prediction of 100% which is not reasonable to use in our model.

This variable is for diffrent substances that release in our data

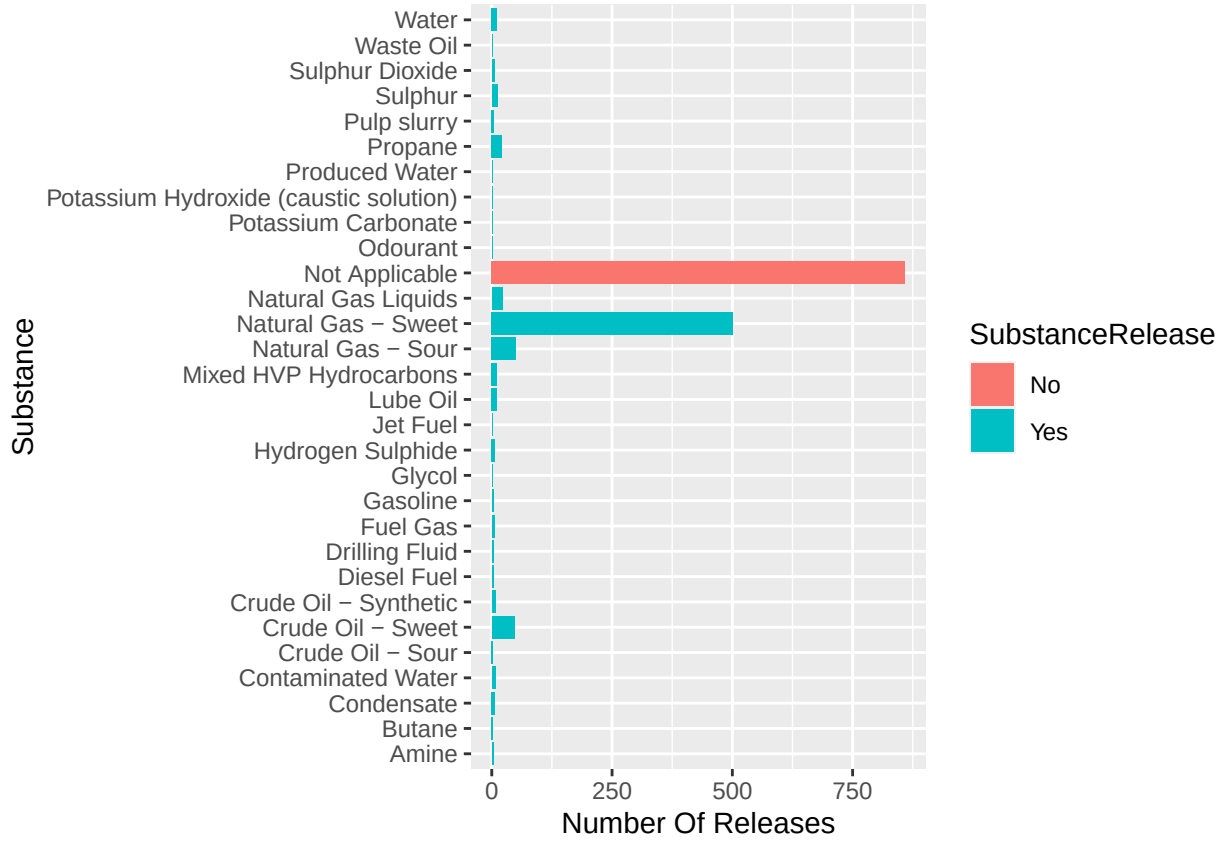
```
t7<-data%>%
  group_by(Substance,SubstanceRelease)%>%
  summarize(Cnt = n())%>%
  arrange(desc(Cnt))
```

```
## `summarise()` has grouped output by 'Substance'. You can override using the
## `.groups` argument.
```

```
t7
```

```
## # A tibble: 30 x 3
## # Groups:   Substance [30]
##    Substance              SubstanceRelease   Cnt
##    <chr>                  <chr>            <int>
##  1 Not Applicable         No                 859
##  2 Natural Gas - Sweet    Yes                501
##  3 Natural Gas - Sour     Yes                 50
##  4 Crude Oil - Sweet      Yes                 47
##  5 Natural Gas Liquids    Yes                 22
##  6 Propane                Yes                 21
##  7 Sulphur                Yes                 12
##  8 Lube Oil               Yes                 11
##  9 Mixed HVP Hydrocarbons Yes                 11
## 10 Water                  Yes                 11
## # ... with 20 more rows
```

```
t7%>%
  ggplot(aes(x=Substance, y=Cnt,fill=SubstanceRelease)) +
  geom_bar(stat="identity")+
   labs( y = "Number Of Releases", x = "Substance")+
  coord_flip()
```



as you can see by the plot above , `Substance` has many categories but yet it can come handy while perdicting the model.

```
table(data$SubstanceRelease,data$Release.Type)
```

```
##
##       Gas Liquid Miscellaneous Not Applicable
##   No    0      0             0            859
##   Yes 614     87            64              0
```

```
table(data$SubstanceRelease,data$Significant)
```

```
##
##        No Yes
##   No  732 127
##   Yes 636 129
```

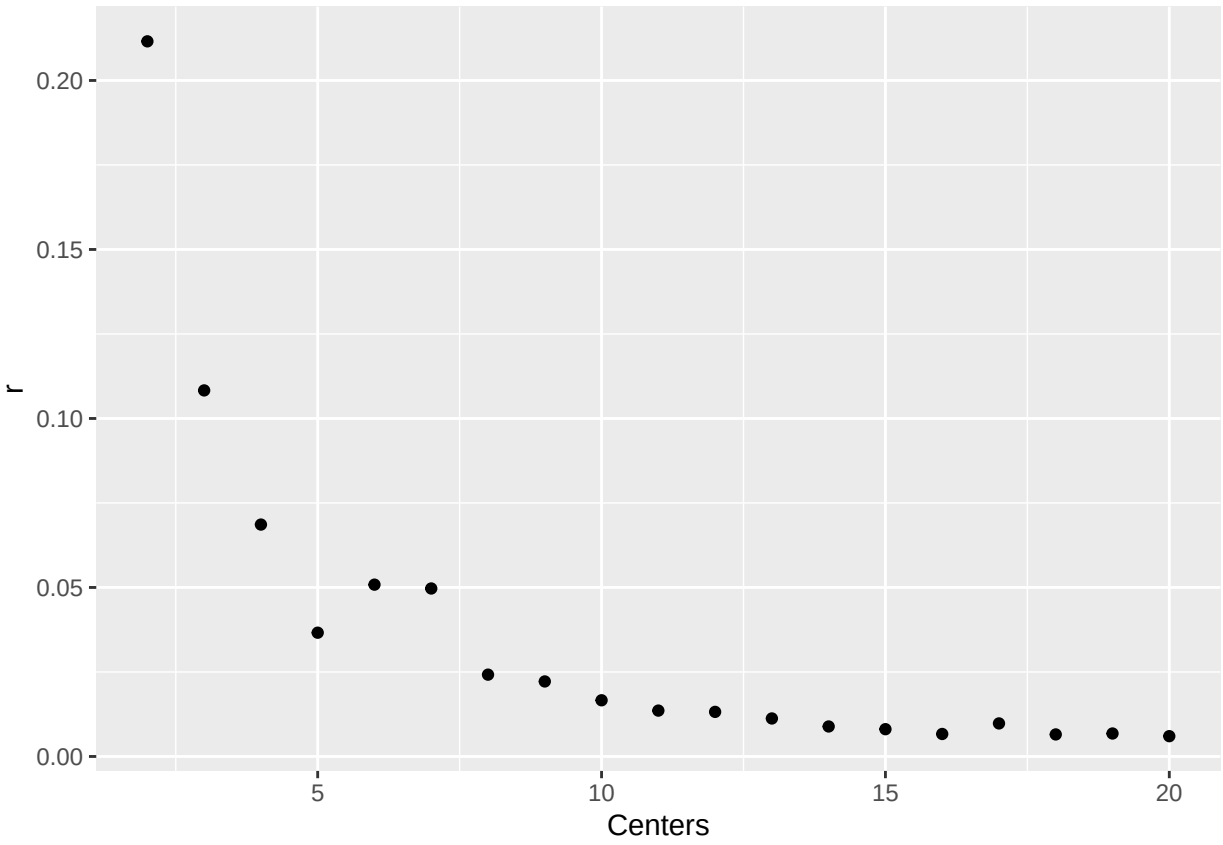
```
chisq.test(table(data$SubstanceRelease,data$Significant))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(data$SubstanceRelease, data$Significant)
## X-squared = 1.1641, df = 1, p-value = 0.2806
```

```
n.why.It.Happend<-c()
n.What.Happened<-c()
for(i in 1:1624)
{
  n.why.It.Happend <- c(n.why.It.Happend,length(strsplit(data$Why.It.Happened[i],",")[[1]]))
  n.What.Happened <- c(n.What.Happened,length(strsplit(data$What.Happened[i],",")[[1]]))
}
data<-data%>%
  mutate(n.why.It.Happend = n.why.It.Happend,
         n.What.Happened = n.What.Happened)
```

K-means Algorithm For Latitude & Longitude

```
r = c()
for(i in 2:20)
{
  k = kmeans(cbind(data$Latitude,data$Longitude) , centers = i)
  r = c(r,k$tot.withinss / k$betweenss)
}
d<-as.data.frame(cbind(2:20,r))
colnames(d)<-c("Centers" , "r")
ggplot(d,aes(x=Centers,y=r))+
  geom_point()
```




```
k = kmeans(cbind(data$Latitude,data$Longitude) , centers = 10)
k_m = as.factor(k$cluster)
data<-data%>%
  mutate(k = k_m)
```

####Modeling With Train and Test

```
data<-data%>%
  mutate(SubstanceRelease = ifelse(SubstanceRelease == "Yes",1,0),
         Significant = ifelse(Significant == "Yes",1,0))
data<-data%>%
  mutate(Province = ifelse(Province %in% c("British Columbia","Northwest Territories"
,"Ontario","Saskatchewan","Quebec"),"Other",Province))
n<-nrow(data)
n.train = trunc(0.7*n)
n.test = n - n.train
train = sample(1:n,n.train)
train.x = data[train,-16]
train.y = data[train,16]
test.x = data[-train,-16]
test.y = data[-train,16]

fit<-glm(SubstanceRelease ~ k,family = binomial(link="logit"),data=cbind(train.x,train.y))
fit
```

```
##
```

```
## Call:  glm(formula = SubstanceRelease ~ k, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Coefficients:
## (Intercept)           k2           k3           k4           k5           k6
##     -0.8473       1.6094       0.1072       0.5336       3.9608       0.8858
##          k7           k8           k9          k10
##      0.3895      -0.7183       1.2412       0.6961
##
## Degrees of Freedom: 1135 Total (i.e. Null);  1126 Residual
## Null Deviance:       1571
## Residual Deviance: 1372  AIC: 1392
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ k, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5128  -0.9902  -0.6160   1.1611   1.8737
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8473     0.1725  -4.911 9.04e-07 ***
## k2            1.6094     0.3668   4.388 1.14e-05 ***
## k3            0.1072     0.2412   0.444 0.656711
## k4            0.5336     0.2744   1.945 0.051825 .
## k5            3.9608     0.5392   7.345 2.06e-13 ***
## k6            0.8858     0.2612   3.391 0.000697 ***
## k7            0.3895     0.2871   1.357 0.174930
## k8           -0.7183     0.3408  -2.108 0.035026 *
## k9            1.2412     0.2214   5.607 2.06e-08 ***
## k10           0.6961     0.2602   2.675 0.007475 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1371.8  on 1126  degrees of freedom
## AIC: 1391.8
##
## Number of Fisher Scoring iterations: 5
```

```
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.5983607
```

```r
fit<-glm(SubstanceRelease ~ Significant ,family = binomial(link="logit"),data=cbind(train.x,train.y))
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Significant, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.211  -1.111  -1.111   1.245   1.245
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.15706    0.06485  -2.422   0.0154 *
## Significant  0.23531    0.16305   1.443   0.1490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1568.7  on 1134  degrees of freedom
## AIC: 1572.7
##
## Number of Fisher Scoring iterations: 3
```

```r
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.5163934
```

```r
fit1<-glm(SubstanceRelease ~ Latitude + Longitude ,family = binomial(link="logit"),data=cbind(train.x,t
summary(fit1)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Latitude + Longitude, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8753  -1.0761  -0.5664   1.1829   2.0187
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.184948   0.986751  -6.268 3.66e-10 ***
## Latitude     0.252264   0.030636   8.234  < 2e-16 ***
## Longitude    0.066228   0.007202   9.196  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1470.3  on 1133  degrees of freedom
## AIC: 1476.3
##
## Number of Fisher Scoring iterations: 4
```

```
yhat<-round(predict.glm(fit1,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.557377
```

```
fit2<-glm(SubstanceRelease ~ Province,family = binomial(link="logit"),data=cbind(train.x,train.y))
summary(fit2)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Province, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6923  -1.0957  -0.9622   1.2614   1.4091
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -0.5298     0.1043  -5.079 3.79e-07 ***
## ProvinceManitoba        0.6782     0.2922   2.321   0.0203 *
## ProvinceNew Brunswick   4.1271     0.7243   5.698 1.21e-08 ***
## ProvinceNova Scotia    16.0959   352.9858   0.046   0.9636
## ProvinceOther           0.3346     0.1329   2.518   0.0118 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1433.1  on 1131  degrees of freedom
## AIC: 1443.1
##
## Number of Fisher Scoring iterations: 14
```

```
yhat<-round(predict.glm(fit2,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.5942623
```

```
fit<-glm(SubstanceRelease ~ Year ,family = binomial(link="logit"),data=cbind(train.x,train.y))
fit
```

```
##
## Call:  glm(formula = SubstanceRelease ~ Year, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Coefficients:
## (Intercept)         Year
##    354.8974      -0.1762
##
## Degrees of Freedom: 1135 Total (i.e. Null);  1134 Residual
## Null Deviance:       1571
## Residual Deviance: 1449  AIC: 1453
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Year, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6818  -0.9932  -0.7402   1.0713   1.7694
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 354.89743   33.93650   10.46   <2e-16 ***
## Year         -0.17618    0.01684  -10.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1449.0  on 1134  degrees of freedom
## AIC: 1453
##
## Number of Fisher Scoring iterations: 4
```

```
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.6721311
```

```
fit3<-glm(SubstanceRelease ~ Release.Type,family = binomial(link="logit"),data=cbind(train.x,train.y))
```

```
## Warning: glm.fit: algorithm did not converge
```

```
summary(fit3)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Release.Type, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.409e-06  -2.409e-06  -2.409e-06   2.409e-06   2.409e-06
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  2.657e+01  1.725e+04   0.002    0.999
## Release.TypeLiquid           7.251e-10  4.841e+04   0.000    1.000
## Release.TypeMiscellaneous   -4.827e-06  5.527e+04   0.000    1.000
## Release.TypeNot Applicable -5.313e+01  2.255e+04  -0.002    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.5708e+03  on 1135  degrees of freedom
## Residual deviance: 6.5906e-09  on 1132  degrees of freedom
## AIC: 8
##
## Number of Fisher Scoring iterations: 25
```

```
yhat<-round(predict.glm(fit3,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 1
```

```
fit4<-glm(SubstanceRelease ~ Status,family = binomial(link="logit"),data=cbind(train.x,train.y))
summary(fit4)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Status, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.137  -1.137  -1.137   1.219   1.665
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -0.09688    0.06051  -1.601    0.109
## StatusInitially Submitted -1.00173    0.66941  -1.496    0.135
## StatusSubmitted           -0.54497    0.39534  -1.378    0.168
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1566.3  on 1133  degrees of freedom
## AIC: 1572.3
##
## Number of Fisher Scoring iterations: 4
```

```
yhat<-round(predict.glm(fit4,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.5266393
```

```
fit<-glm(SubstanceRelease ~ Year + Province,family = binomial(link="logit"),data=cbind(train.x,train.y))
fit
```

```
##
## Call:  glm(formula = SubstanceRelease ~ Year + Province, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Coefficients:
##           (Intercept)                   Year       ProvinceManitoba
##              347.8365               -0.1729                 0.5028
## ProvinceNew Brunswick    ProvinceNova Scotia          ProvinceOther
##                4.1038               15.9977                 0.3994
##
## Degrees of Freedom: 1135 Total (i.e. Null);  1130 Residual
## Null Deviance:       1571
## Residual Deviance: 1326  AIC: 1338
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Year + Province, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0945  -0.9605  -0.6144   1.0892   1.9522
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           347.83648   35.37901   9.832  < 2e-16 ***
## Year                   -0.17289    0.01756  -9.845  < 2e-16 ***
## ProvinceManitoba        0.50281    0.31018   1.621  0.10502
## ProvinceNew Brunswick   4.10375    0.73031   5.619 1.92e-08 ***
## ProvinceNova Scotia    15.99766  344.29598   0.046  0.96294
## ProvinceOther           0.39941    0.14030   2.847  0.00442 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1325.7  on 1130  degrees of freedom
## AIC: 1337.7
##
## Number of Fisher Scoring iterations: 14
```

```
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.682377
```

```
#fit5<-glm(SubstanceRelease ~ Substance,family = #binomial(link="logit"),data=cbind(train.x,train.y))
#yhat<-round(predict.glm(fit5,newdata = test.x,type = "response"))
#tb<-table(yhat,as.data.frame(test.y)[,1])
#sum(diag(tb))/sum(tb)

#fit6<-glm(SubstanceRelease ~ Nearest.Populated.Centre,family = #binomial(link="logit"),data=cbind(trai
#yhat<-round(predict.glm(fit6,newdata = test.x,type = "response"))
#tb<-table(yhat,as.data.frame(test.y)[,1])
#sum(diag(tb))/sum(tb)

#fit6<-glm(SubstanceRelease ~ Company,family = #binomial(link="logit"),data=cbind(train.x,train.y))
#yhat<-round(predict.glm(fit6,newdata = test.x,type = "response"))
#tb<-table(yhat,as.data.frame(test.y)[,1])
#sum(diag(tb))/sum(tb)
```

As you can see above , we did not include these variables in the model since they would cause errors in test and training process because of the many classes they each have .(and for some reason in this process some of the classes disappear!).also fixing this issue is tremendously time consuming and there is no reasonable way to merge these classes together in order to get decent results.

```
fit<-glm(SubstanceRelease ~ n.What.Happened + n.why.It.Happend + Latitude + Longitude + Year + Province
fit
```

```
##
## Call:  glm(formula = SubstanceRelease ~ n.What.Happened + n.why.It.Happend +
##     Latitude + Longitude + Year + Province + Status + Significant,
##     family = binomial(link = "logit"), data = cbind(train.x,
##         train.y))
##
## Coefficients:
##               (Intercept)            n.What.Happened
##                 365.16103                    0.01108
##          n.why.It.Happend                   Latitude
##                   0.07954                    0.14476
##                 Longitude                       Year
##                   0.01260                   -0.18474
##          ProvinceManitoba      ProvinceNew Brunswick
##                   0.86351                    4.66809
```

```
##       ProvinceNova Scotia              ProvinceOther
##                  16.72851                    0.67491
## StatusInitially Submitted            StatusSubmitted
##                   0.57325                    0.58156
##               Significant
##                   0.37173
##
## Degrees of Freedom: 1135 Total (i.e. Null);  1123 Residual
## Null Deviance:       1571
## Residual Deviance: 1276  AIC: 1302
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ n.What.Happened + n.why.It.Happend +
##     Latitude + Longitude + Year + Province + Status + Significant,
##     family = binomial(link = "logit"), data = cbind(train.x,
##         train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1765  -0.9417  -0.5540   1.0370   2.1117
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                365.161034  40.361936   9.047  < 2e-16 ***
## n.What.Happened              0.011085   0.142192   0.078   0.9379
## n.why.It.Happend             0.079541   0.100130   0.794   0.4270
## Latitude                     0.144760   0.033983   4.260 2.05e-05 ***
## Longitude                    0.012599   0.009106   1.384   0.1665
## Year                        -0.184739   0.020051  -9.213  < 2e-16 ***
## ProvinceManitoba             0.863512   0.321329   2.687   0.0072 **
## ProvinceNew Brunswick        4.668088   0.775185   6.022 1.72e-09 ***
## ProvinceNova Scotia         16.728508 342.817105   0.049   0.9611
## ProvinceOther                0.674909   0.154493   4.369 1.25e-05 ***
## StatusInitially Submitted    0.573254   0.693635   0.826   0.4085
## StatusSubmitted              0.581558   0.459603   1.265   0.2057
## Significant                  0.371729   0.193389   1.922   0.0546 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1275.8  on 1123  degrees of freedom
## AIC: 1301.8
##
## Number of Fisher Scoring iterations: 14
```

```
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.7254098
```

```
fit<-glm(SubstanceRelease ~ Latitude + Longitude + Year + Province,family = binomial(link="logit"),data=
fit
```

```
##
## Call:  glm(formula = SubstanceRelease ~ Latitude + Longitude + Year +
##     Province, family = binomial(link = "logit"), data = cbind(train.x,
##     train.y))
##
## Coefficients:
##           (Intercept)               Latitude              Longitude
##             335.03276                0.14683                0.01104
##                  Year       ProvinceManitoba  ProvinceNew Brunswick
##              -0.16984                0.94990                4.71142
##   ProvinceNova Scotia          ProvinceOther
##              16.74205                0.70865
##
## Degrees of Freedom: 1135 Total (i.e. Null);  1128 Residual
## Null Deviance:       1571
## Residual Deviance: 1283  AIC: 1299
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ Latitude + Longitude + Year +
##     Province, family = binomial(link = "logit"), data = cbind(train.x,
##     train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0411  -0.9443  -0.5781   1.0390   2.0745
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           335.032759  37.038940   9.045  < 2e-16 ***
## Latitude                0.146828   0.033436   4.391 1.13e-05 ***
## Longitude               0.011042   0.008974   1.230    0.219
## Year                   -0.169836   0.018363  -9.249  < 2e-16 ***
## ProvinceManitoba        0.949897   0.320029   2.968    0.003 **
## ProvinceNew Brunswick   4.711415   0.773202   6.093 1.11e-09 ***
## ProvinceNova Scotia    16.742046 343.858157   0.049    0.961
## ProvinceOther           0.708654   0.152233   4.655 3.24e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1283.5  on 1128  degrees of freedom
## AIC: 1299.5
##
## Number of Fisher Scoring iterations: 14
```

```
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.670082
```

```
fit<-glm(SubstanceRelease ~ k + Year + Province,family = binomial(link="logit"),data=cbind(train.x,trai
fit
```

```
##
## Call:  glm(formula = SubstanceRelease ~ k + Year + Province, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Coefficients:
##           (Intercept)                    k2                    k3
##             337.44668               1.48110               0.48458
##                    k4                    k5                    k6
##               1.16565             -13.75484               0.68393
##                    k7                    k8                    k9
##               1.04692               0.01342               1.36090
##                   k10                  Year       ProvinceManitoba
##               0.87446              -0.16814               0.59162
## ProvinceNew Brunswick    ProvinceNova Scotia          ProvinceOther
##              18.67866              30.58288               0.47104
##
## Degrees of Freedom: 1135 Total (i.e. Null);  1121 Residual
## Null Deviance:       1571
## Residual Deviance: 1269  AIC: 1299
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ k + Year + Province, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0824  -0.9591  -0.5322   1.0046   2.0901
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            337.44668   38.29012   8.813  < 2e-16 ***
## k2                       1.48110    0.38502   3.847  0.00012 ***
## k3                       0.48458    0.38315   1.265  0.20597
## k4                       1.16565    0.39509   2.950  0.00317 **
## k5                     -13.75484 1027.96899  -0.013  0.98932
## k6                       0.68393    0.32397   2.111  0.03476 *
## k7                       1.04692    0.40236   2.602  0.00927 **
## k8                       0.01342    0.36098   0.037  0.97035
## k9                       1.36090    0.23207   5.864 4.51e-09 ***
## k10                      0.87446    0.31860   2.745  0.00606 **
```

```
## Year                    -0.16814    0.01901  -8.845  < 2e-16 ***
## ProvinceManitoba         0.59162    0.47431   1.247  0.21228
## ProvinceNew Brunswick   18.67866 1027.96926   0.018  0.98550
## ProvinceNova Scotia     30.58288 1084.45956   0.028  0.97750
## ProvinceOther            0.47104    0.28922   1.629  0.10340
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1268.5  on 1121  degrees of freedom
## AIC: 1298.5
##
## Number of Fisher Scoring iterations: 14
```

```
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.6905738
```

```
fit<-glm(SubstanceRelease ~ k + Year,family = binomial(link="logit"),data=cbind(train.x,train.y))
fit
```

```
##
## Call:  glm(formula = SubstanceRelease ~ k + Year, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Coefficients:
## (Intercept)           k2           k3           k4           k5           k6
##   344.96665      1.49129      0.01156      0.73072      3.99553      0.73308
##          k7           k8           k9          k10         Year
##     0.62547      0.02276      1.33295      0.61048     -0.17164
##
## Degrees of Freedom: 1135 Total (i.e. Null);  1125 Residual
## Null Deviance:       1571
## Residual Deviance: 1282  AIC: 1304
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = SubstanceRelease ~ k + Year, family = binomial(link = "logit"),
##     data = cbind(train.x, train.y))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9395  -0.9442  -0.5287   1.0102   2.1014
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) 344.96665   38.19241   9.032  < 2e-16 ***
## k2            1.49129    0.38443   3.879 0.000105 ***
## k3            0.01156    0.25178   0.046 0.963365
## k4            0.73072    0.29054   2.515 0.011901 *
## k5            3.99553    0.54713   7.303 2.82e-13 ***
## k6            0.73308    0.27291   2.686 0.007229 **
## k7            0.62547    0.30527   2.049 0.040469 *
## k8            0.02276    0.36133   0.063 0.949784
## k9            1.33295    0.23160   5.755 8.64e-09 ***
## k10           0.61048    0.27244   2.241 0.025039 *
## Year         -0.17164    0.01896  -9.053  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1570.8  on 1135  degrees of freedom
## Residual deviance: 1282.3  on 1125  degrees of freedom
## AIC: 1304.3
##
## Number of Fisher Scoring iterations: 5
```

```
yhat<-round(predict.glm(fit,newdata = test.x,type = "response"))
tb<-table(yhat,as.data.frame(test.y)[,1])
sum(diag(tb))/sum(tb)
```

```
## [1] 0.6864754
```

Conclusion finally , we did not include some of the variables (e.g `Release Type` or `Company`) in our model(we explained the reasoning behind these decisions before),other than those ,our model gave us an accuracy of approximately 70 %. this is the best result we could achieve due to the fact that our results vary from 50% to 70%.