```
#load libararies
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import pandas as pd
%matplotlib inline

#load datasets
data = pd.read_csv('/covtype.csv.zip')
data.head()
```

{"type":"dataframe","variable_name":"data"}

```
#list of columns
print(data.columns)
```

```
Index(['Elevation', 'Aspect', 'Slope',
'Horizontal_Distance_To_Hydrology',
       'Vertical_Distance_To_Hydrology',
'Horizontal_Distance_To_Roadways',
       'Hillshade_9am', 'Hillshade_Noon', 'Hillshade_3pm',
       'Horizontal_Distance_To_Fire_Points', 'Wilderness_Area1',
       'Wilderness_Area2', 'Wilderness_Area3', 'Wilderness_Area4',
       'Soil_Type1', 'Soil_Type2', 'Soil_Type3', 'Soil_Type4',
'Soil_Type5',
       'Soil_Type6', 'Soil_Type7', 'Soil_Type8', 'Soil_Type9',
'Soil_Type10',
       'Soil_Type11', 'Soil_Type12', 'Soil_Type13', 'Soil_Type14',
       'Soil_Type15', 'Soil_Type16', 'Soil_Type17', 'Soil_Type18',
       'Soil_Type19', 'Soil_Type20', 'Soil_Type21', 'Soil_Type22',
       'Soil_Type23', 'Soil_Type24', 'Soil_Type25', 'Soil_Type26',
       'Soil_Type27', 'Soil_Type28', 'Soil_Type29', 'Soil_Type30',
       'Soil_Type31', 'Soil_Type32', 'Soil_Type33', 'Soil_Type34',
       'Soil_Type35', 'Soil_Type36', 'Soil_Type37', 'Soil_Type38',
       'Soil_Type39', 'Soil_Type40', 'Cover_Type'],
      dtype='object')
```

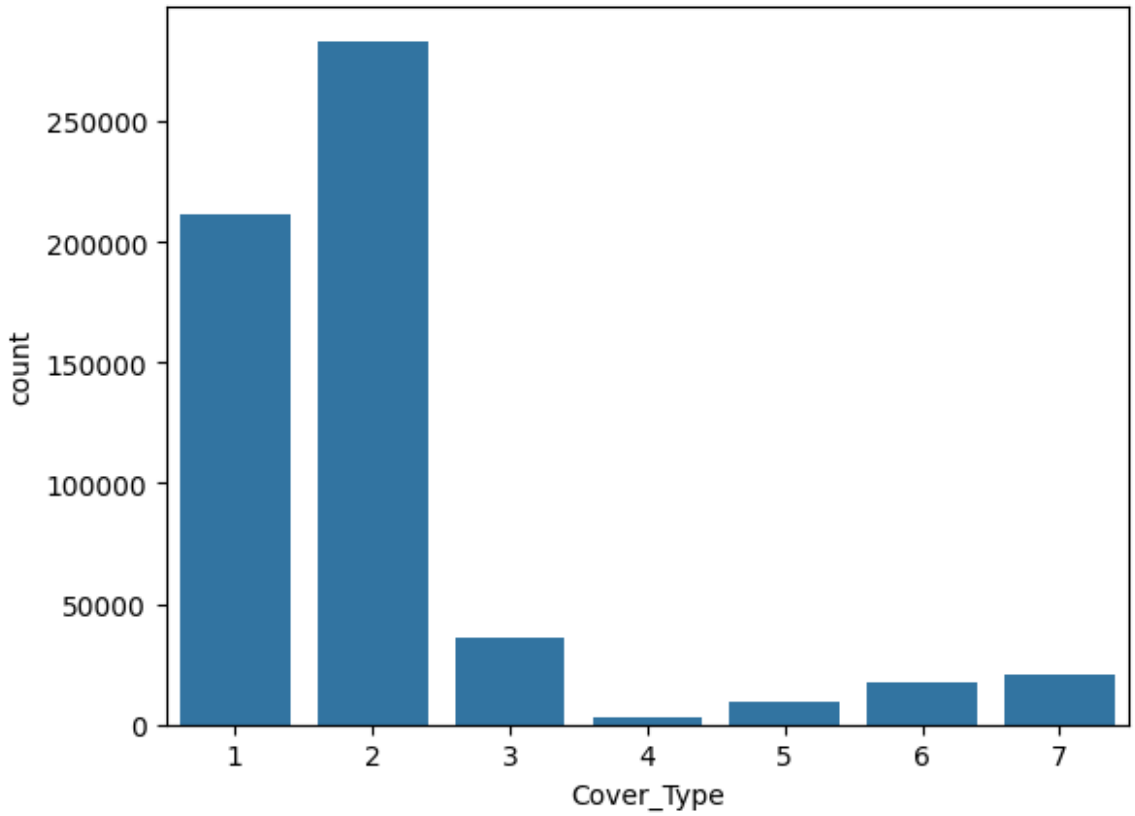
```
#shape of data
data.shape
```

```
(581012, 55)
```

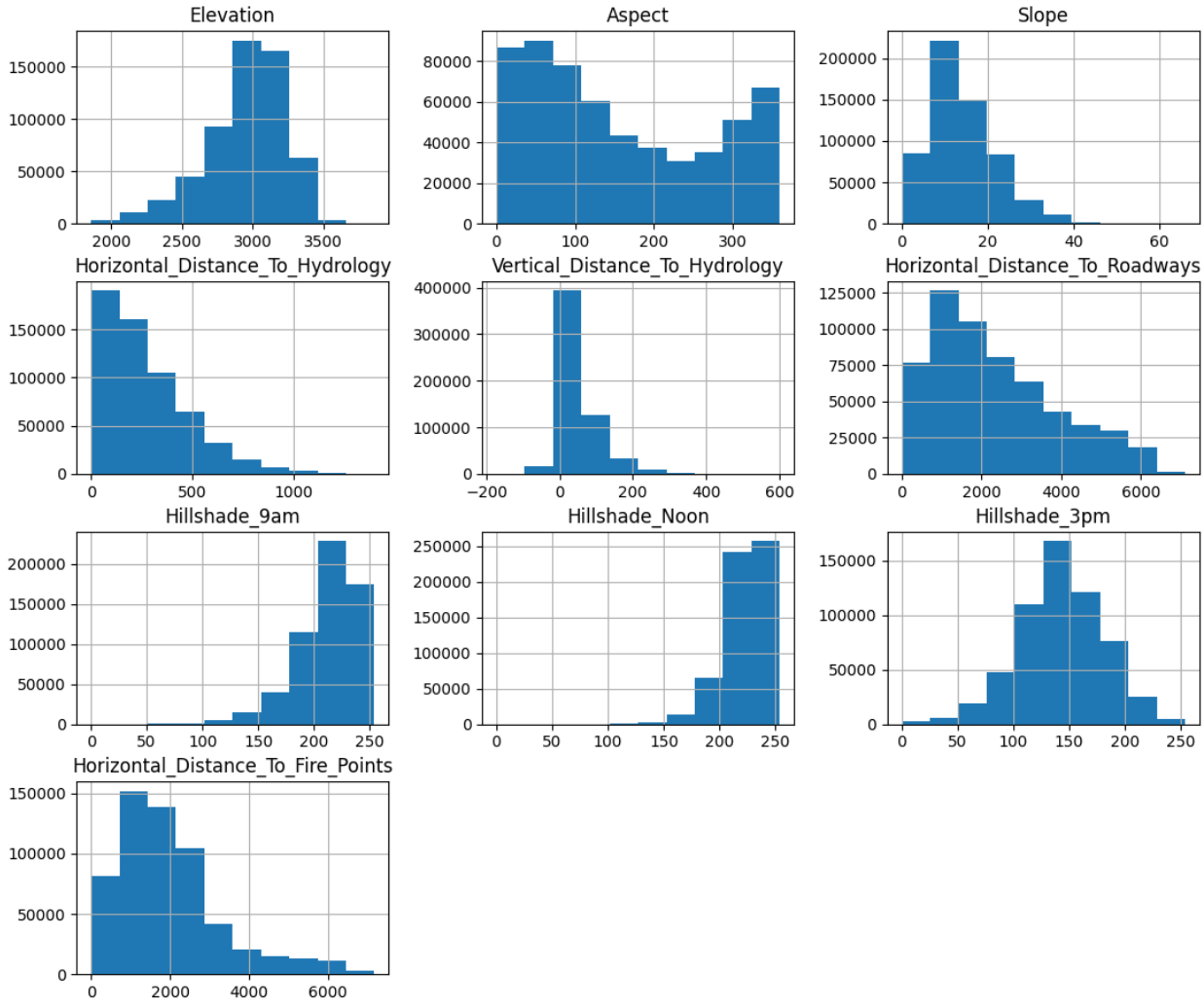
```
#check mising values
print(list(data.isnull().any()))
```

```
[False, False, False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False, False, False,
```

```
False, False, False, False, False, False, False, False, False, False,
False, False, False, False, False]

data.describe()

{"type":"dataframe"}

#About Target/Cover_Type variable
data.Cover_Type.value_counts()

Cover_Type
2    283301
1    211840
3     35754
7     20510
6     17367
5      9493
4      2747
Name: count, dtype: int64

#count plot of target
sb.countplot(x='Cover_Type', data=data)
plt.show()
```

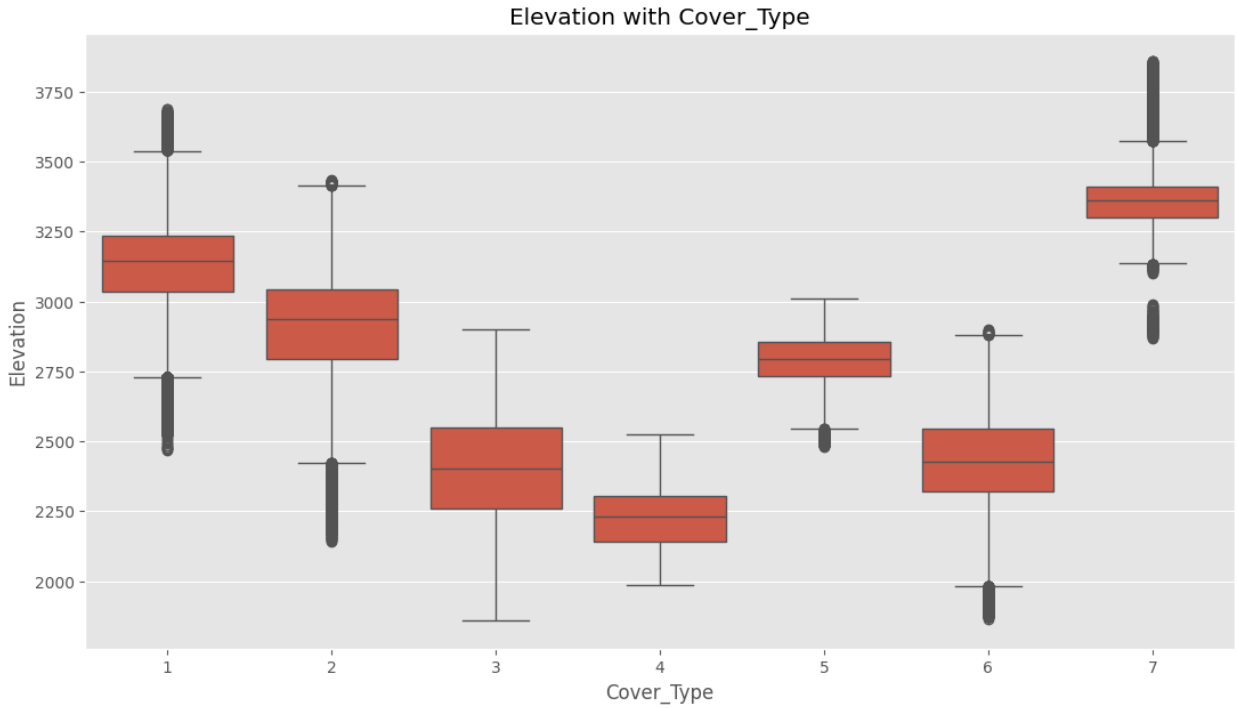


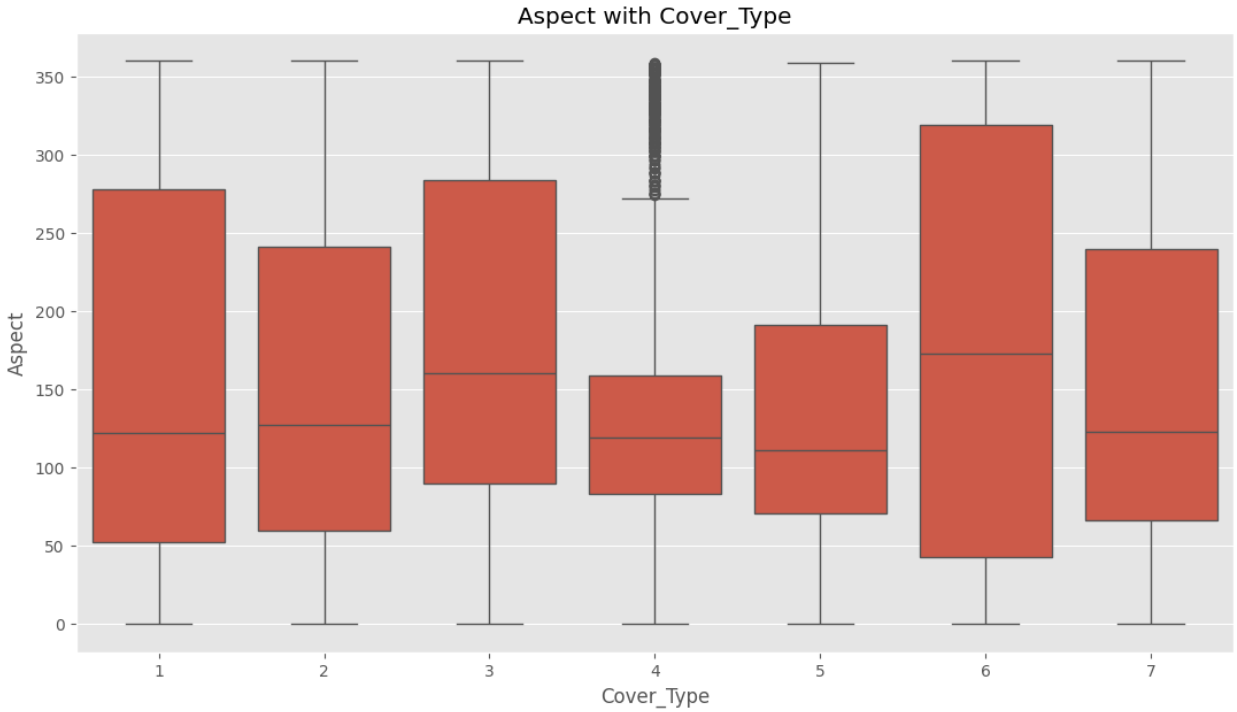
```
#Take some column
col = ['Elevation', 'Aspect', 'Slope',
'Horizontal_Distance_To_Hydrology',
       'Vertical_Distance_To_Hydrology',
'Horizontal_Distance_To_Roadways',
       'Hillshade_9am', 'Hillshade_Noon', 'Hillshade_3pm',
       'Horizontal_Distance_To_Fire_Points']

train = data[col]

#histogram
train.hist(figsize=(13, 11))
plt.show()
```

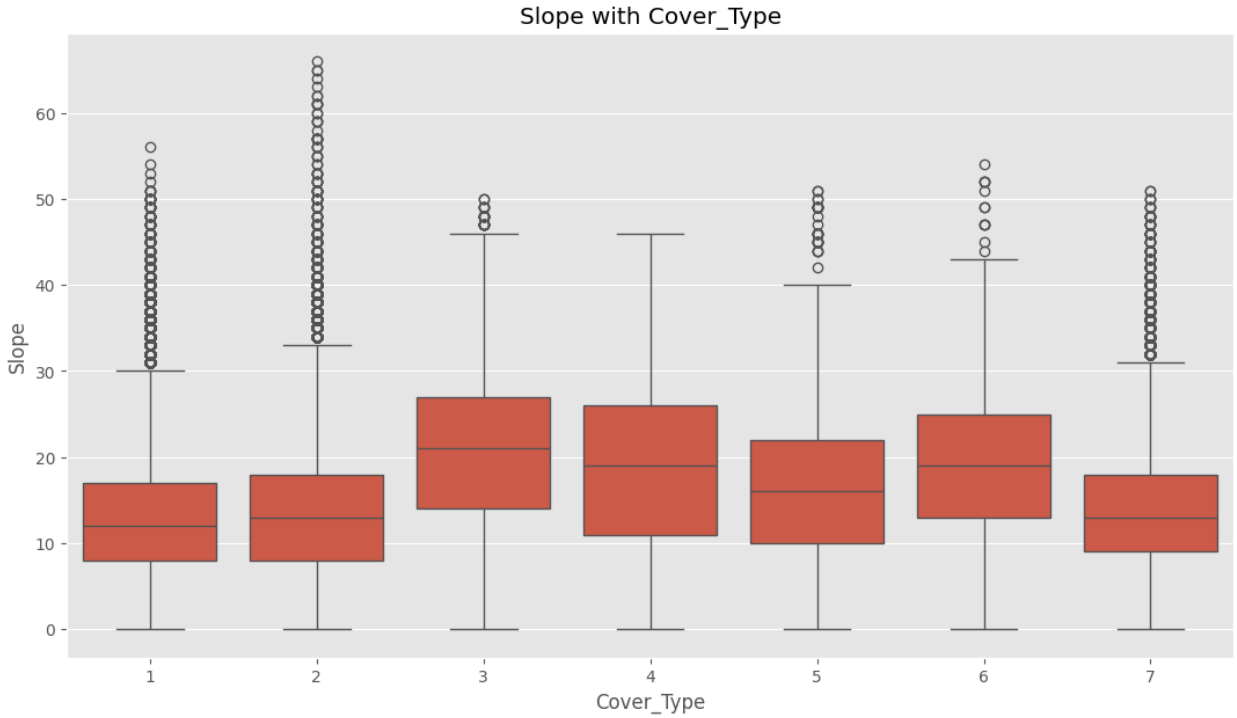


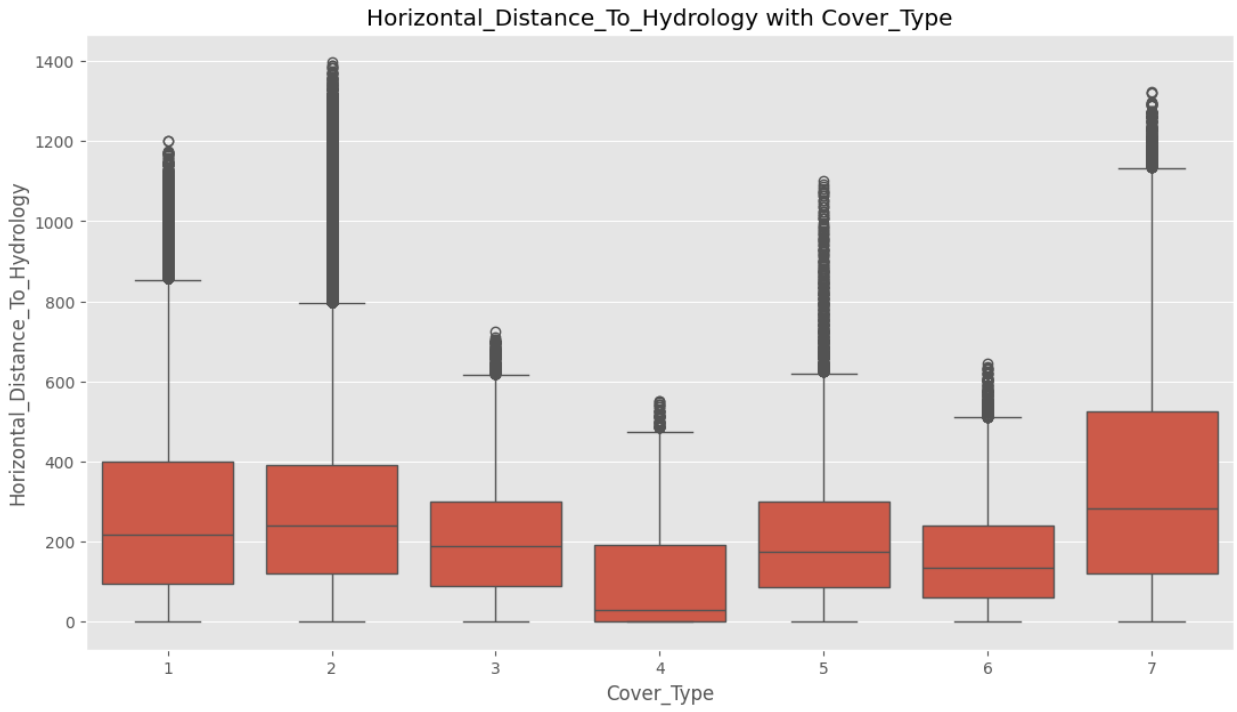

```
#Boxplot
plt.style.use('ggplot')
for i in col:
    plt.figure(figsize=(13, 7))
```

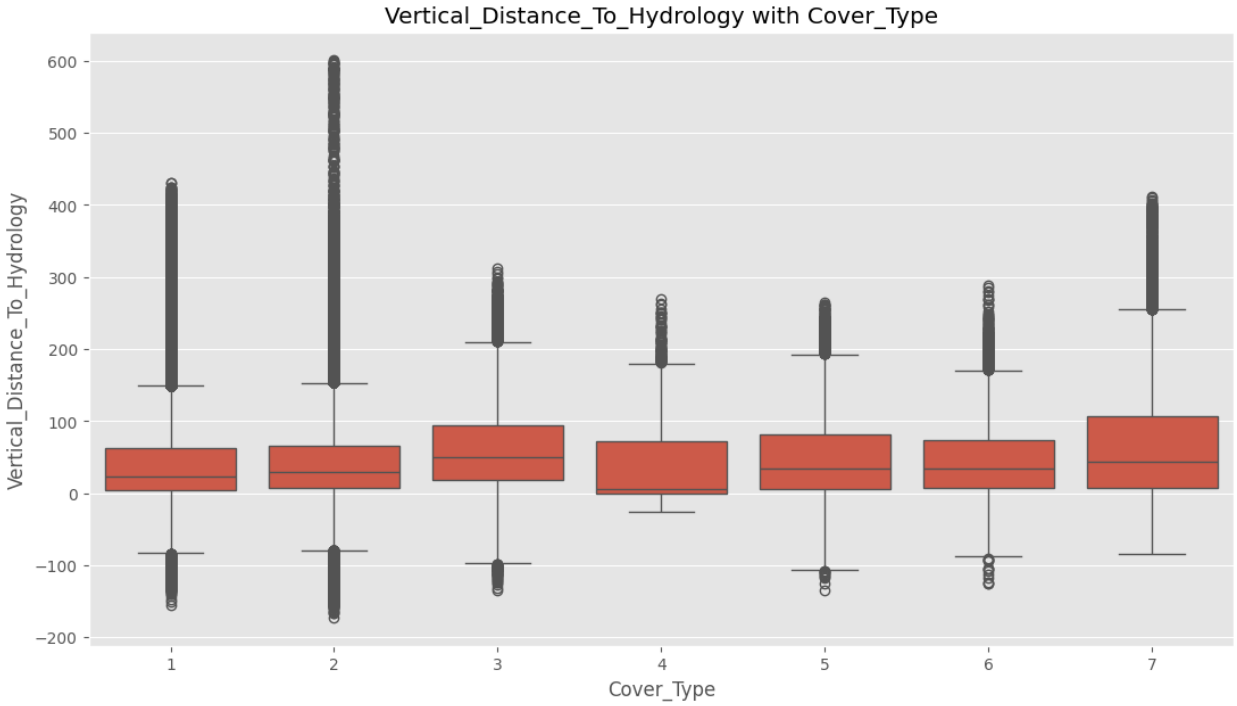
```
    plt.title(str(i) + " with " + str('Cover_Type'))
    sb.boxplot(x=data.Cover_Type, y=train[i])
    plt.show()
```

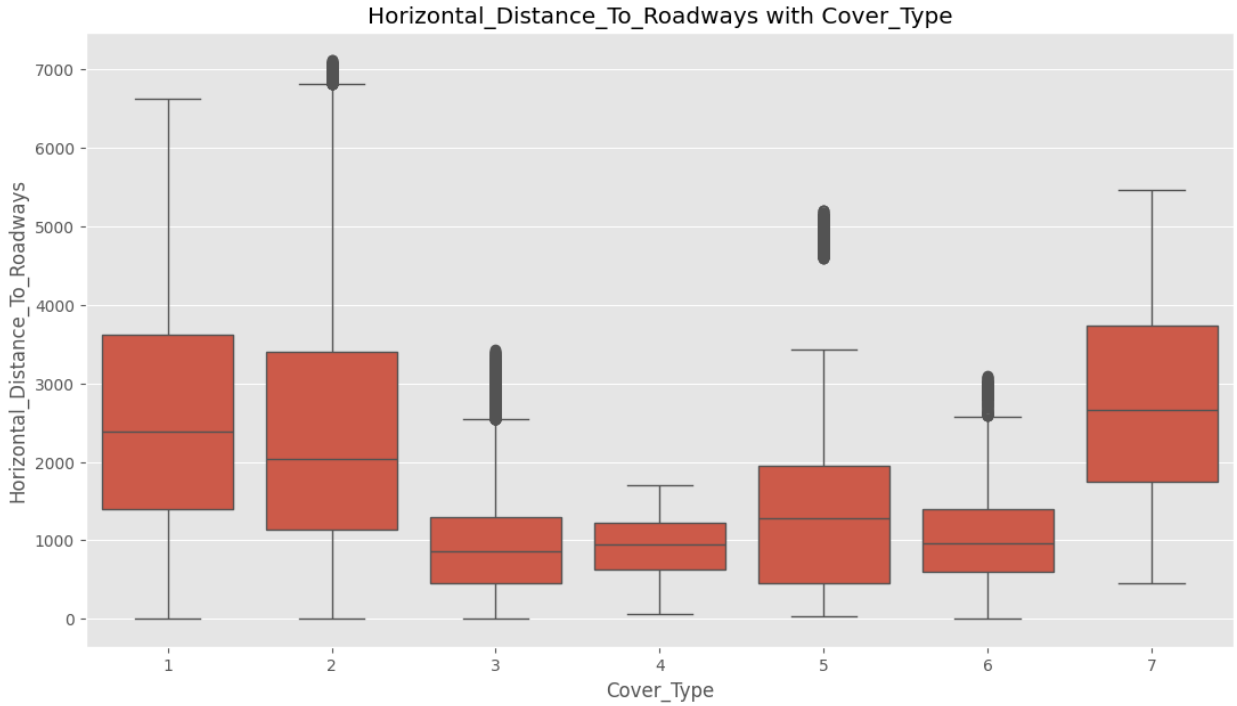


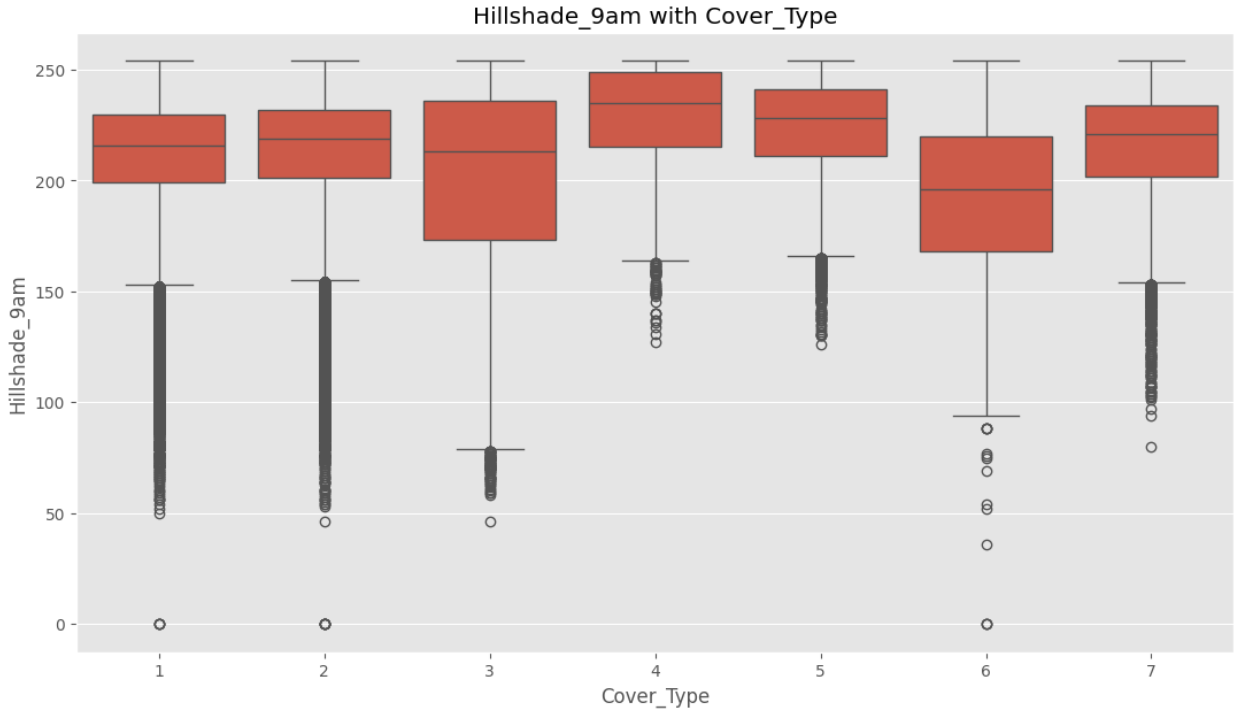


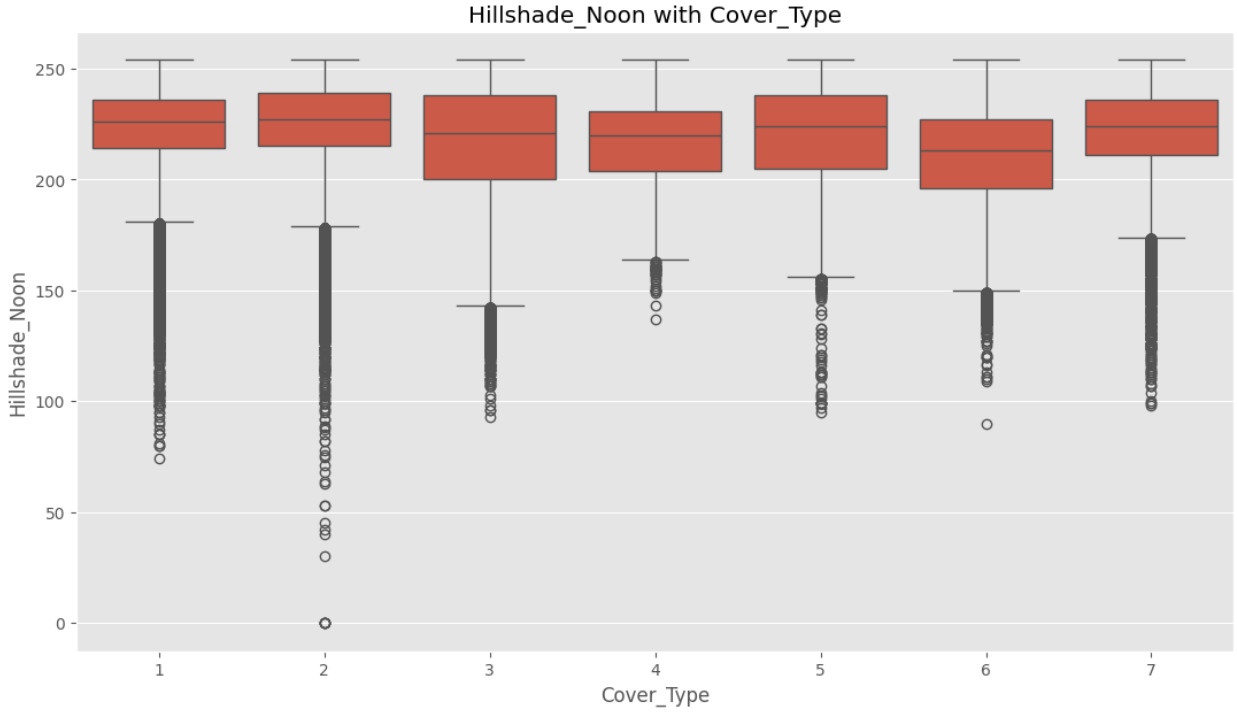

Slope with Cover_Type
Slope
0
10
20
30
40
50
60
1
2
3
4
5
6
7
Cover_Type

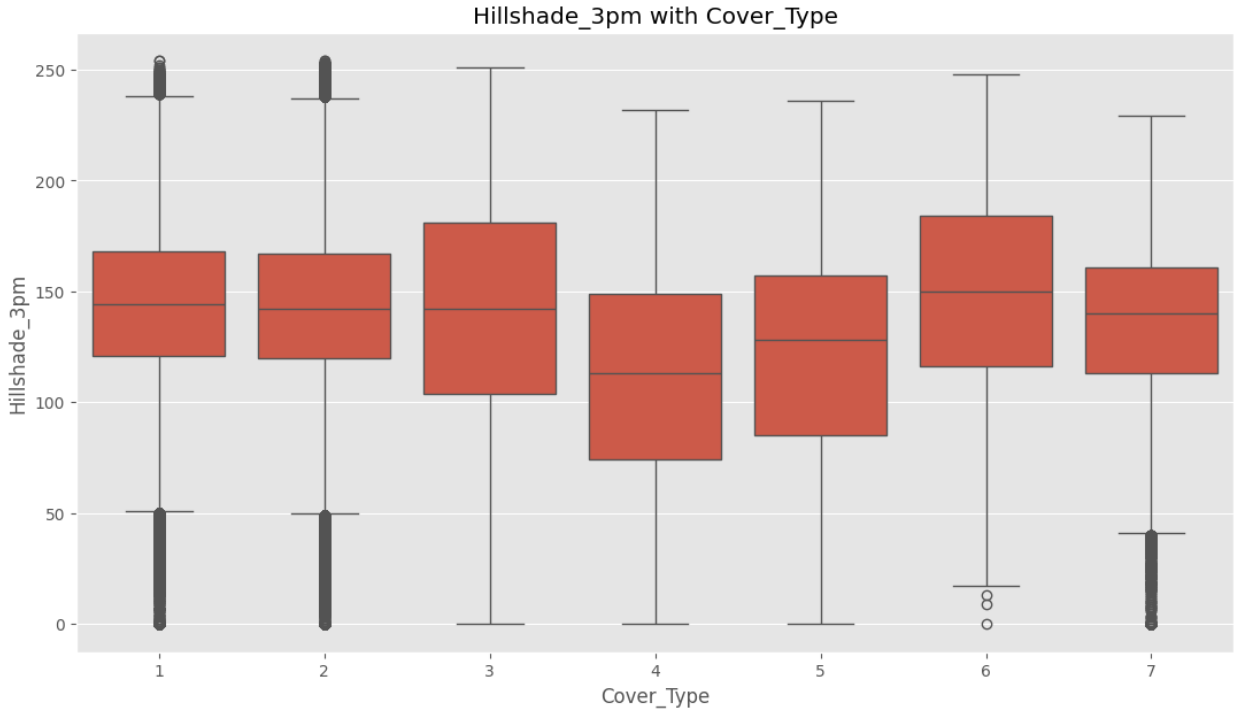

Horizontal_Distance_To_Hydrology with Cover_Type
Horizontal_Distance_To_Hydrology
0
200
400
600
800
1000
1200
1400
1
2
3
4
5
6
7
Cover_Type

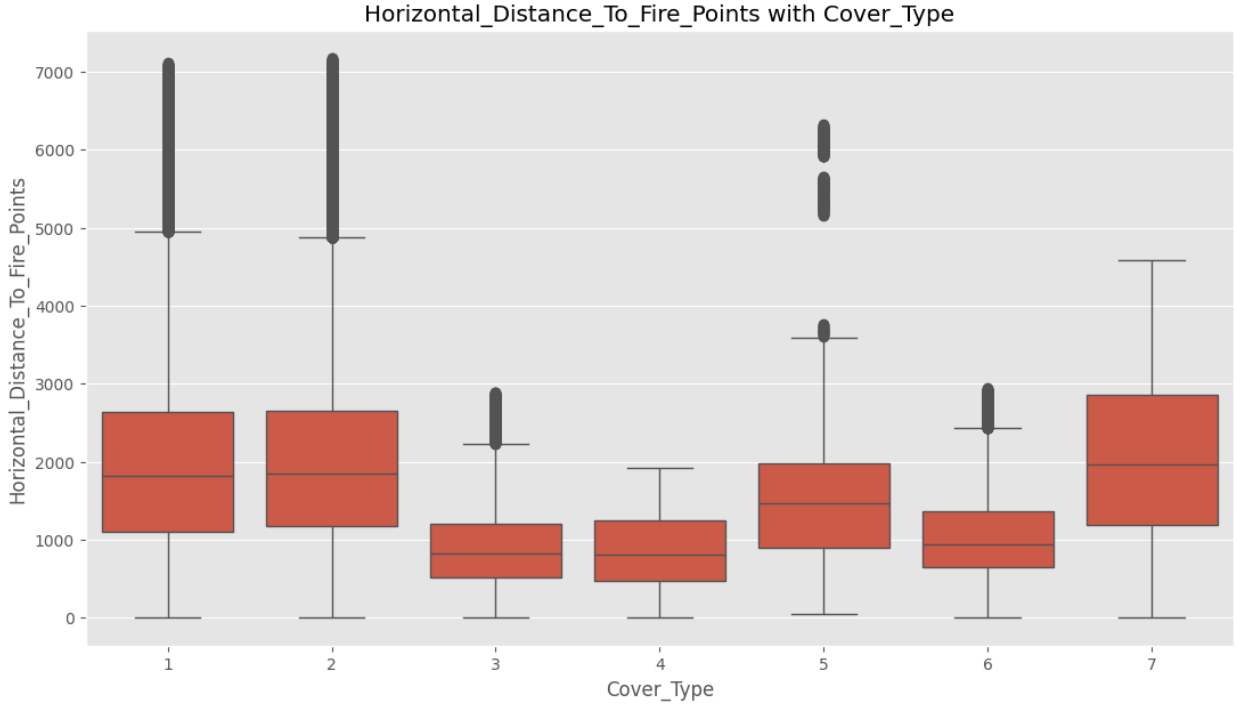

Vertical_Distance_To_Hydrology with Cover_Type
Vertical_Distance_To_Hydrology
600
500
400
300
200
100
0
−100
−200
1
2
3
4
5
6
7
Cover_Type


Horizontal_Distance_To_Roadways with Cover_Type
Horizontal_Distance_To_Roadways
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
6
7
Cover_Type


Hillshade_9am with Cover_Type
Hillshade_9am
Cover_Type
0
50
100
150
200
250
1
2
3
4
5
6
7


Hillshade_Noon with Cover_Type
Hillshade_Noon
Cover_Type
0
50
100
150
200
250
1
2
3
4
5
6
7






```
#Corralation
plt.figure(figsize=(12, 8))
corr = train.corr()
sb.heatmap(corr, annot=True)
plt.show()
```


```
#Feature Selection
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import SelectFromModel

#separate features and target
feature = data.iloc[:, :54] #Features of data
y = data.iloc[:, 54]  #Target of data

# Features Reduction
ETC = ExtraTreesClassifier()
ETC = ETC.fit(feature, y)

model = SelectFromModel(ETC, prefit=True)
X = model.transform(feature) #new features
```

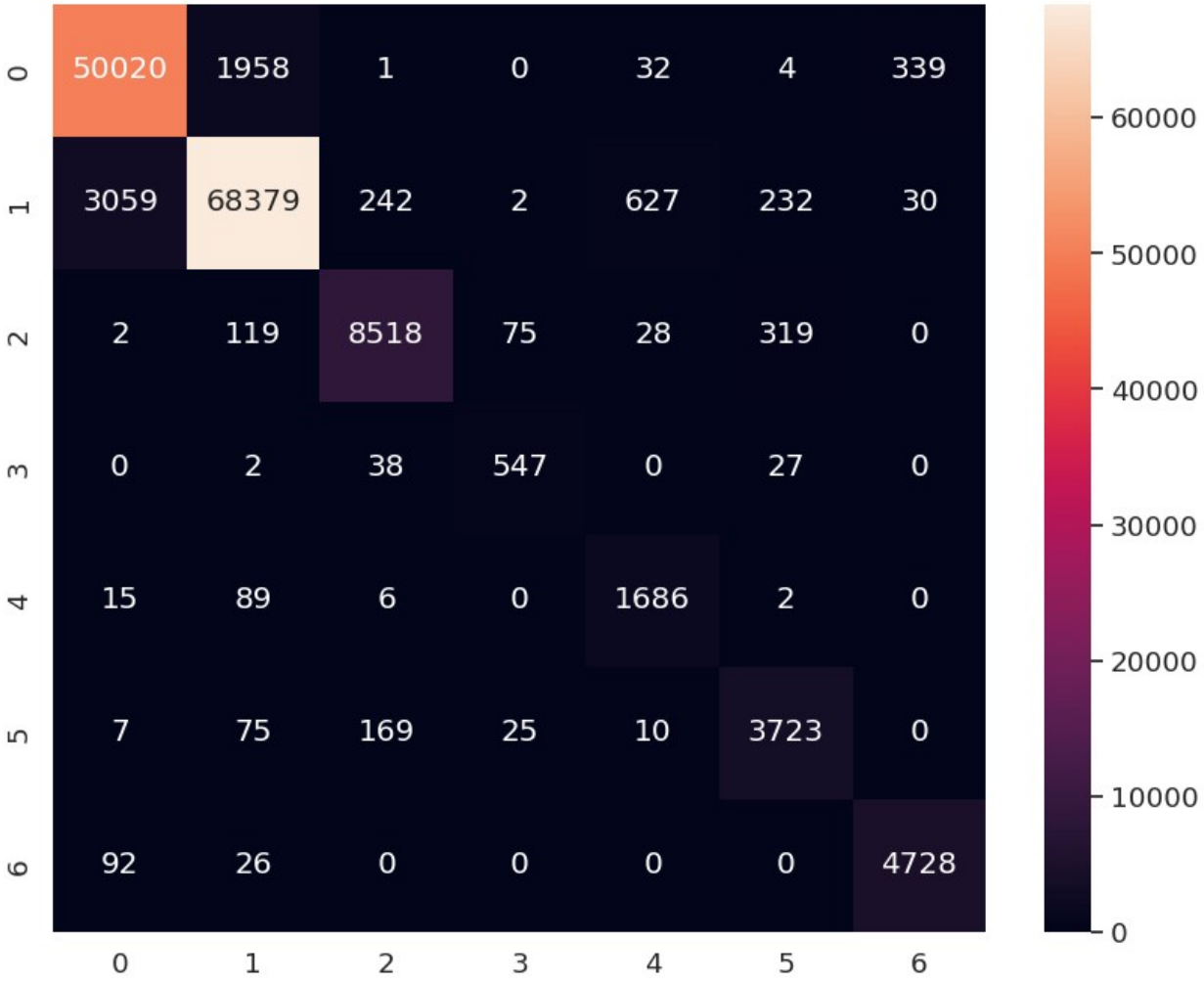
```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:486:
UserWarning: X has feature names, but SelectFromModel was fitted
without feature names
  warnings.warn(
```

```
#shape of new feature
X.shape
```

```
(581012, 12)
```

```
#Split the data into test and train formate
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25, random_state=10)

#Random Forest
from sklearn.ensemble import RandomForestClassifier
RFC = RandomForestClassifier(n_estimators=100)

#fit
RFC.fit(X_train, y_train)

#prediction
y_pred = RFC.predict(X_test)

#score
print("Accuracy -- ", RFC.score(X_test, y_test)*100)

#confusion
cm = confusion_matrix(y_pred, y_test)
plt.figure(figsize=(10, 8))
sb.set(font_scale=1.2)
sb.heatmap(cm, annot=True, fmt='g')
plt.show()
```

```
Accuracy --  94.73195045885456
```


0
1
2
3
4
5
6
50020 1958 1 0 32 4 339
3059 68379 242 2 627 232 30
2 119 8518 75 28 319 0
0 2 38 547 0 27 0
15 89 6 0 1686 2 0
7 75 169 25 10 3723 0
92 26 0 0 0 0 4728
0
1
2
3
4
5
6
60000
50000
40000
30000
20000
10000
0