```
import pandas as pd
```

```
df=pd.read_csv('covtype.csv')
df.head()
```

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Roadwa
0	2596	51	3	258	0	5
1	2590	56	2	212	-6	3
2	2804	139	9	268	65	31
3	2785	155	18	242	118	30
4	2595	45	2	153	-1	3

5 rows × 55 columns

```
df.shape
```

```
(581012, 55)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 581012 entries, 0 to 581011
Data columns (total 55 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   Elevation                           581012 non-null  int64
 1   Aspect                              581012 non-null  int64
 2   Slope                               581012 non-null  int64
 3   Horizontal_Distance_To_Hydrology    581012 non-null  int64
 4   Vertical_Distance_To_Hydrology      581012 non-null  int64
 5   Horizontal_Distance_To_Roadways     581012 non-null  int64
 6   Hillshade_9am                       581012 non-null  int64
 7   Hillshade_Noon                      581012 non-null  int64
 8   Hillshade_3pm                       581012 non-null  int64
 9   Horizontal_Distance_To_Fire_Points  581012 non-null  int64
 10  Wilderness_Area1                    581012 non-null  int64
 11  Wilderness_Area2                    581012 non-null  int64
 12  Wilderness_Area3                    581012 non-null  int64
 13  Wilderness_Area4                    581012 non-null  int64
 14  Soil_Type1                          581012 non-null  int64
 15  Soil_Type2                          581012 non-null  int64
 16  Soil_Type3                          581012 non-null  int64
 17  Soil_Type4                          581012 non-null  int64
 18  Soil_Type5                          581012 non-null  int64
 19  Soil_Type6                          581012 non-null  int64
 20  Soil_Type7                          581012 non-null  int64
 21  Soil_Type8                          581012 non-null  int64
 22  Soil_Type9                          581012 non-null  int64
 23  Soil_Type10                         581012 non-null  int64
 24  Soil_Type11                         581012 non-null  int64
 25  Soil_Type12                         581012 non-null  int64
 26  Soil_Type13                         581012 non-null  int64
 27  Soil_Type14                         581012 non-null  int64
 28  Soil_Type15                         581012 non-null  int64
 29  Soil_Type16                         581012 non-null  int64
 30  Soil_Type17                         581012 non-null  int64
 31  Soil_Type18                         581012 non-null  int64
 32  Soil_Type19                         581012 non-null  int64
 33  Soil_Type20                         581012 non-null  int64
 34  Soil_Type21                         581012 non-null  int64
 35  Soil_Type22                         581012 non-null  int64
 36  Soil_Type23                         581012 non-null  int64
 37  Soil_Type24                         581012 non-null  int64
 38  Soil_Type25                         581012 non-null  int64
 39  Soil_Type26                         581012 non-null  int64
 40  Soil_Type27                         581012 non-null  int64
 41  Soil_Type28                         581012 non-null  int64
 42  Soil_Type29                         581012 non-null  int64
 43  Soil_Type30                         581012 non-null  int64
 44  Soil_Type31                         581012 non-null  int64
 45  Soil_Type32                         581012 non-null  int64
 46  Soil_Type33                         581012 non-null  int64
 47  Soil_Type34                         581012 non-null  int64
 48  Soil_Type35                         581012 non-null  int64
 49  Soil_Type36                         581012 non-null  int64
 50  Soil_Type37                         581012 non-null  int64
 51  Soil_Type38                         581012 non-null  int64
 52  Soil_Type39                         581012 non-null  int64
```

```
df.isnull().sum()
```

	0
Elevation	0
Aspect	0
Slope	0
Horizontal_Distance_To_Hydrology	0
Vertical_Distance_To_Hydrology	0
Horizontal_Distance_To_Roadways	0
Hillshade_9am	0
Hillshade_Noon	0
Hillshade_3pm	0
Horizontal_Distance_To_Fire_Points	0
Wilderness_Area1	0
Wilderness_Area2	0
Wilderness_Area3	0
Wilderness_Area4	0
Soil_Type1	0
Soil_Type2	0
Soil_Type3	0
Soil_Type4	0
Soil_Type5	0
Soil_Type6	0
Soil_Type7	0
Soil_Type8	0
Soil_Type9	0
Soil_Type10	0
Soil_Type11	0
Soil_Type12	0
Soil_Type13	0
Soil_Type14	0
Soil_Type15	0
Soil_Type16	0
Soil_Type17	0
Soil_Type19	0
Soil_Type21	0
Soil_Type23	0
Soil_Type25	0
Soil_Type27	0
Soil_Type28	0
Soil_Type29	0
Soil_Type30	0
Soil_Type32	0

```
df.describe()
```

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizon
count	581012.000000	581012.000000	581012.000000	581012.000000	581012.000000	
mean	2959.365301	155.656807	14.103704	269.428217	46.418855	
std	279.984734	111.913721	7.488242	212.549356	58.295232	
min	1859.000000	0.000000	0.000000	0.000000	-173.000000	
25%	2809.000000	58.000000	9.000000	108.000000	7.000000	
50%	2996.000000	127.000000	13.000000	218.000000	30.000000	
75%	3163.000000	260.000000	18.000000	384.000000	69.000000	
max	3858.000000	360.000000	66.000000	1397.000000	601.000000	

8 rows × 55 columns

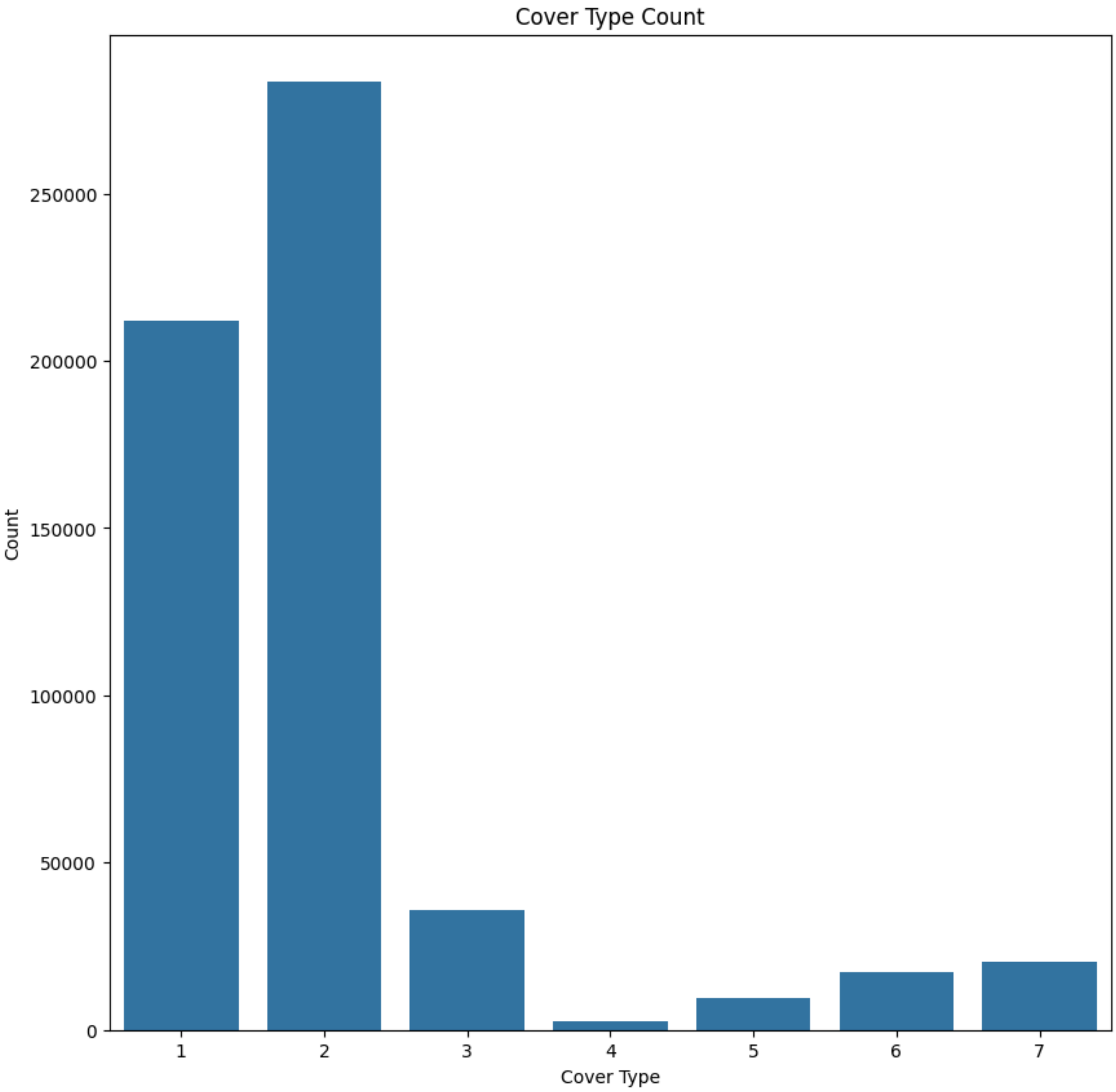
```
df['Cover_Type'].unique()
```

```
array([5, 2, 1, 7, 3, 6, 4])
```

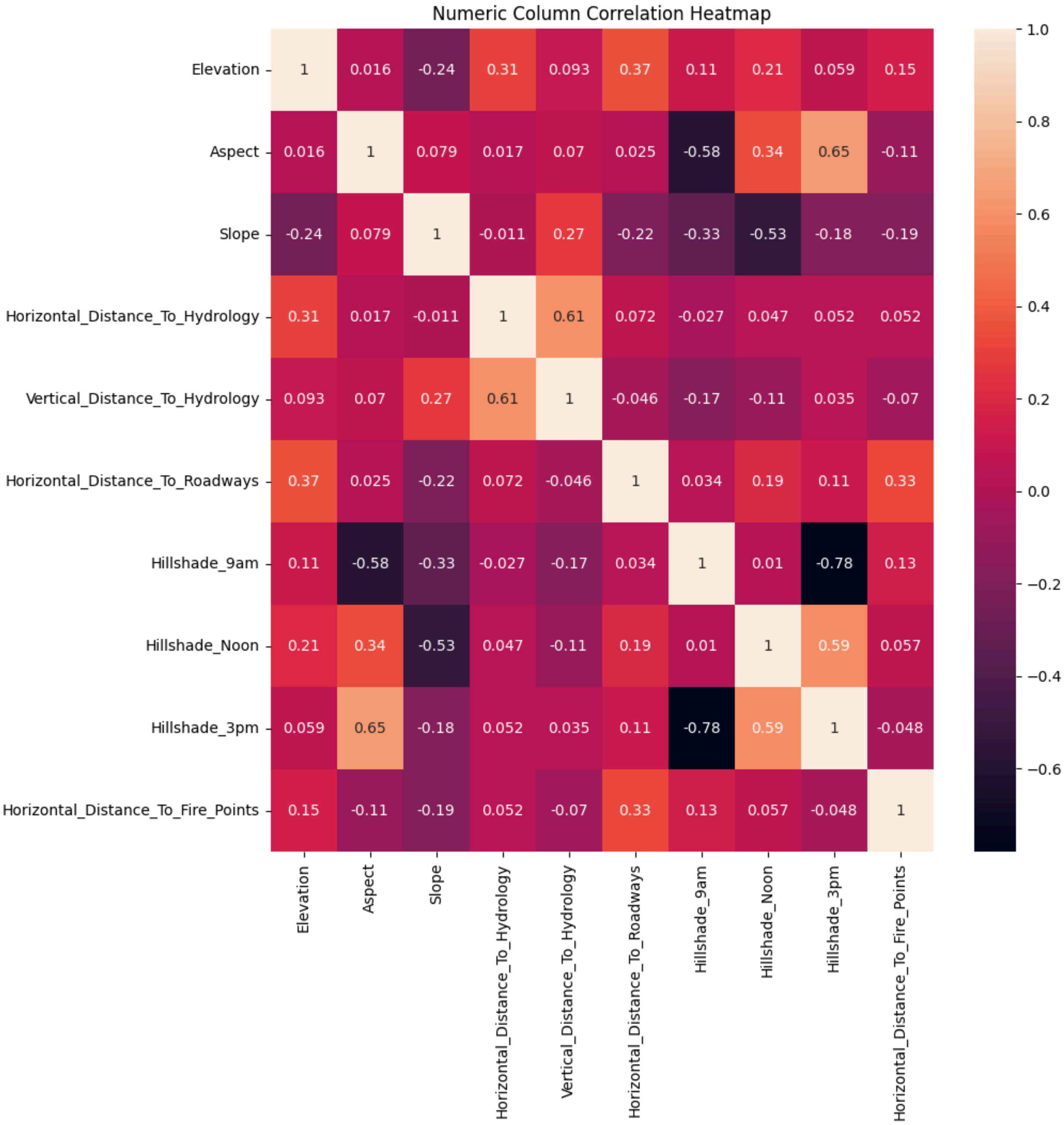
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(10,10))
sns.countplot(x=df['Cover_Type'])
plt.title('Cover Type Count')
plt.xlabel('Cover Type')
plt.ylabel('Count')
plt.show()
```

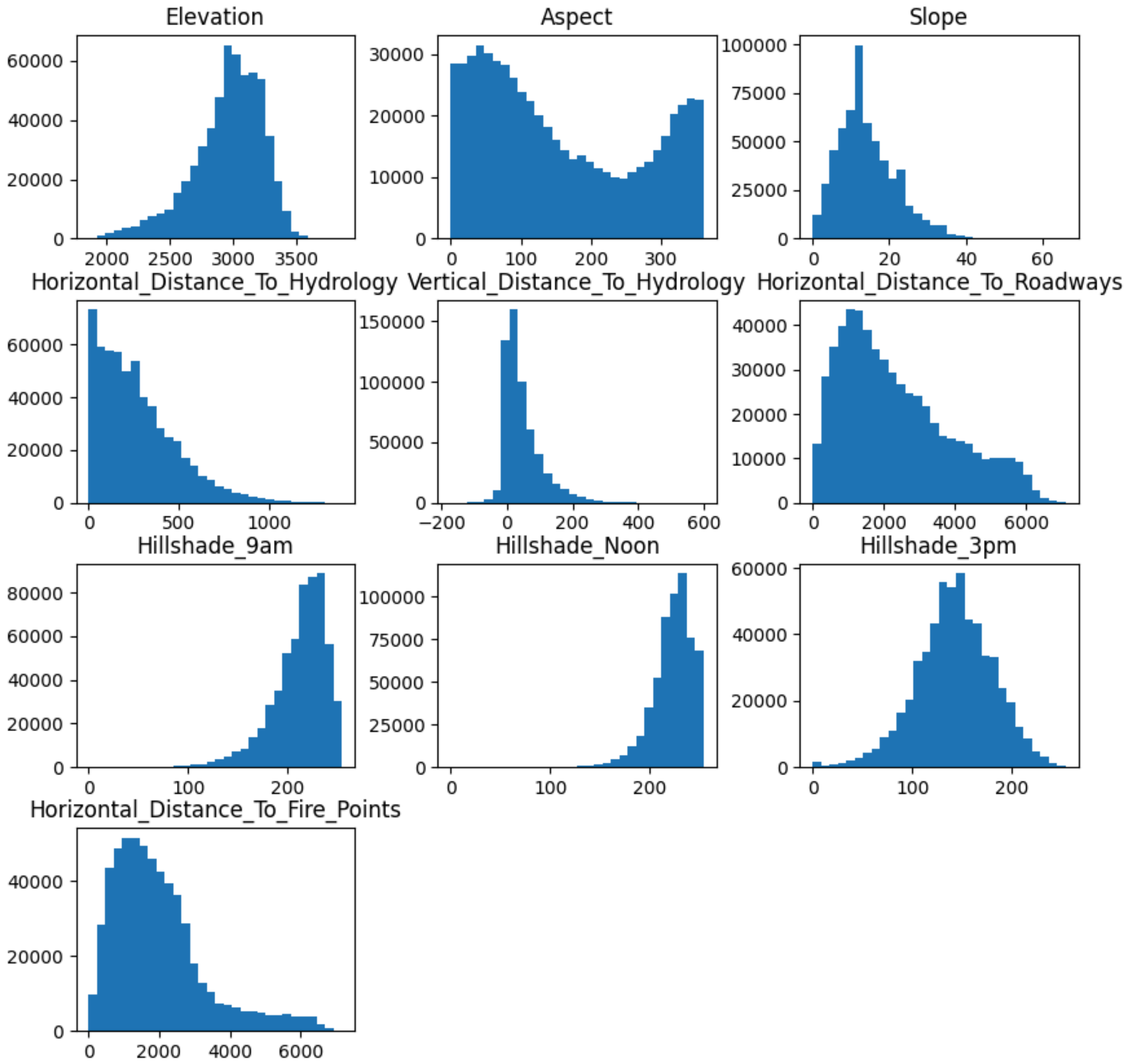
```
Soil_Type39          0
```



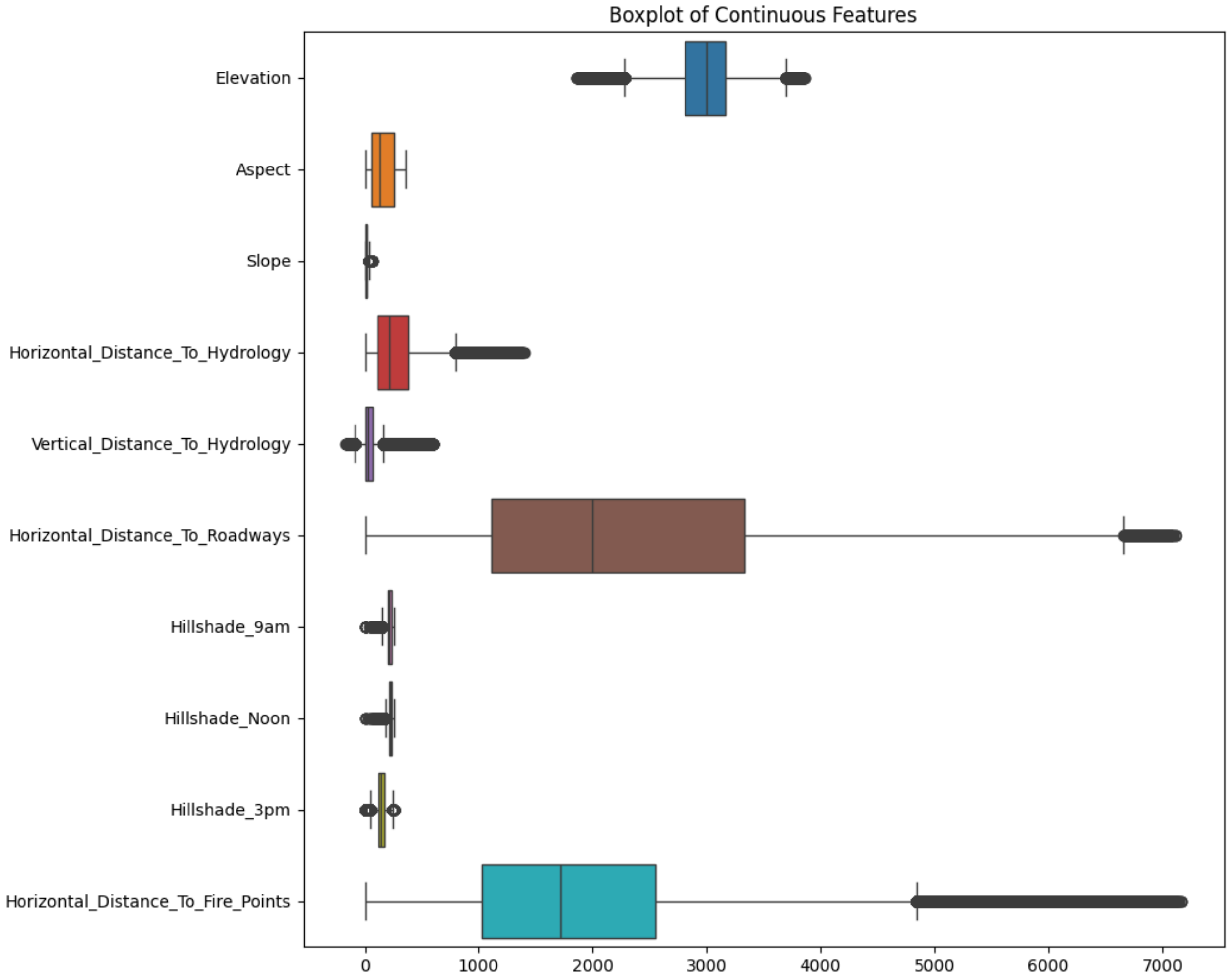
```
numeric_cols=df.iloc[:,:10]

plt.figure(figsize=(10,10))
sns.heatmap(numeric_cols.corr(),annot=True)
plt.title('Numeric Column Correlation Heatmap')
plt.show()
```



```
numeric_cols.hist(figsize=(10,10),bins=30,grid=False)
plt.show()
```



```
plt.figure(figsize=(10,10))
sns.boxplot(data=numeric_cols,orient='h')
plt.title('Boxplot of Continuous Features')
plt.show()
```



```
wilderness_cols = [col for col in df.columns if "Wilderness_Area" in col]
df[wilderness_cols].sum()
```

```
                        0
Wilderness_Area1   260796
Wilderness_Area2    29884
Wilderness_Area3   253364
Wilderness_Area4    36968

dtype: int64
```

```
soil_cols = [col for col in df.columns if "Soil_Type" in col]
df[soil_cols].sum()
```

```
                 0
Soil_Type1    3031
Soil_Type2    7525
Soil_Type3    4823
Soil_Type4   12396
Soil_Type5    1597
Soil_Type6    6575
Soil_Type7     105
Soil_Type8     179
Soil_Type9    1147
Soil_Type10  32634
Soil_Type11  12410
Soil_Type12  29971
Soil_Type13  17431
Soil_Type14    599
Soil_Type15      3
Soil_Type16   2845
Soil_Type17   3422
Soil_Type18   1899
Soil_Type19   4021
Soil_Type20   9259
Soil_Type21    838
Soil_Type22  33373
Soil_Type23  57752
Soil_Type24  21278
Soil_Type25    474
Soil_Type26   2589
Soil_Type27   1086
Soil_Type28    946
Soil_Type29 115247
Soil_Type30  30170
Soil_Type31  25666
Soil_Type32  52519
Soil_Type33  45154
Soil_Type34   1611
Soil_Type35   1891
Soil_Type36    119
Soil_Type37    298
Soil_Type38  15573
Soil_Type39  13806
Soil_Type40   8750

dtype: int64
```

```
X=df.drop("Cover_Type",axis=1)
y=df["Cover_Type"]
```

```
X.head()
```

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Roadwa
0	2596	51	3	258	0	5
1	2590	56	2	212	-6	3
2	2804	139	9	268	65	31
3	2785	155	18	242	118	30
4	2595	45	2	153	-1	3

5 rows × 54 columns

```
y.head()
```

	Cover_Type
0	5
1	5
2	2
3	2
4	5

dtype: int64

```
X_cont=X.iloc[:,:10]
X_bin=X.iloc[:,10:]
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_cont_scaled = scaler.fit_transform(X_cont)
```

```
import numpy as np

X = np.hstack([X_cont_scaled, X_bin.values])
```

```
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42,stratify=y)
```

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(
    multi_class='multinomial',
    solver='lbfgs',
    max_iter=200,
    n_jobs=-1
)

lr.fit(X_train, y_train)
```

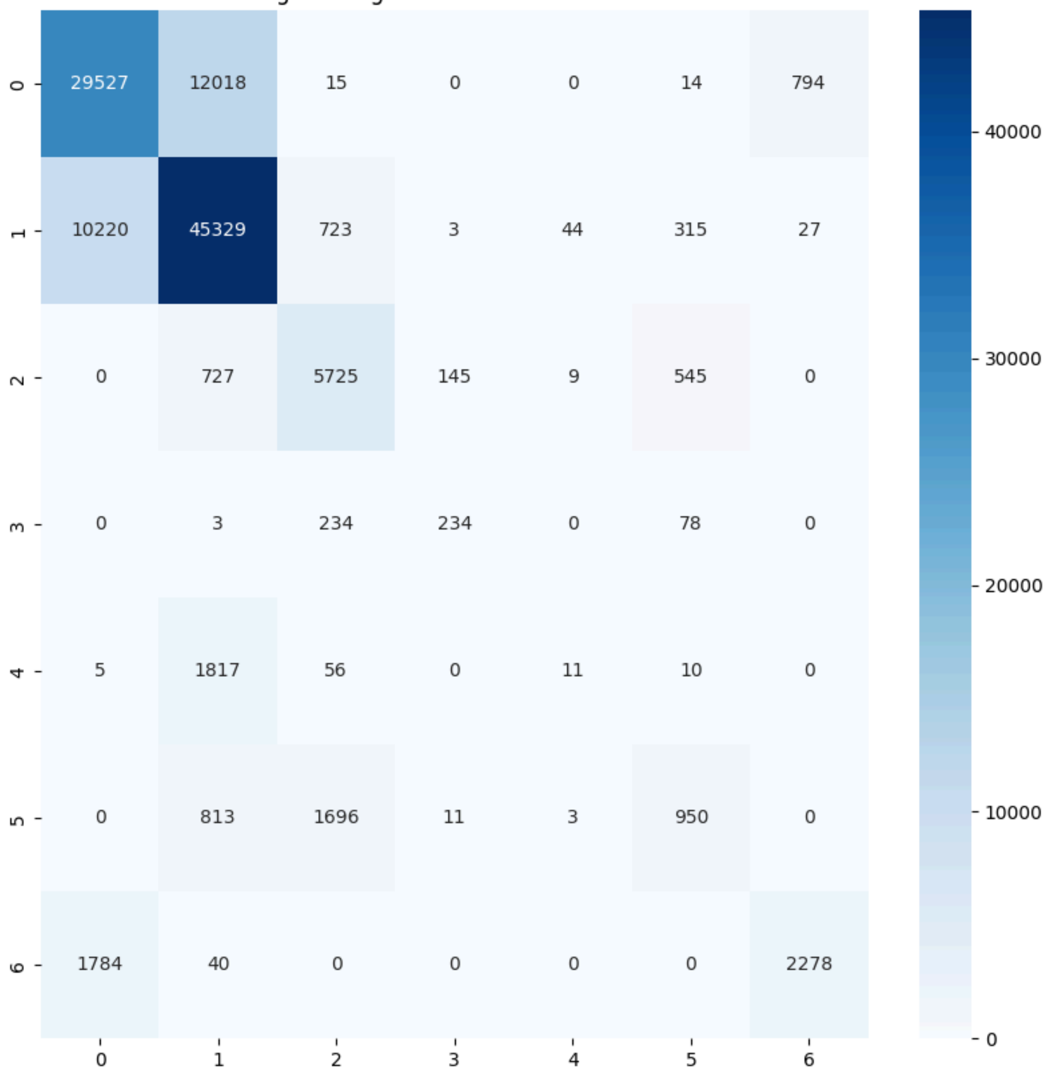
```
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_logistic.py:1247: FutureWarning: 'multi_class' was deprecated
  warnings.warn(
```

```
▾                         LogisticRegression                         i ?
LogisticRegression(max_iter=200, multi_class='multinomial', n_jobs=-1)
```

```
from sklearn.metrics import classification_report, accuracy_score,confusion_matrix

y_pred_lr = lr.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred_lr))
print(classification_report(y_test, y_pred_lr))

plt.figure(figsize=(10, 10))
sns.heatmap(confusion_matrix(y_test, y_pred_lr), annot=True, fmt='d', cmap='Blues')
plt.title('Logistic Regression Confusion Matrix')
plt
```

```
Accuracy: 0.7233376074628022
              precision    recall  f1-score   support

           1       0.71      0.70      0.70     42368
           2       0.75      0.80      0.77     56661
           3       0.68      0.80      0.73      7151
           4       0.60      0.43      0.50       549
           5       0.16      0.01      0.01      1899
           6       0.50      0.27      0.35      3473
           7       0.74      0.56      0.63      4102

    accuracy                           0.72    116203
   macro avg       0.59      0.51      0.53    116203
weighted avg       0.71      0.72      0.71    116203

<module 'matplotlib.pyplot' from '/usr/local/lib/python3.12/dist-packages/matplotlib/pyplot.py'>
```

Logistic Regression Confusion Matrix



	0	1	2	3	4	5	6
0	29527	12018	15	0	0	14	794
1	10220	45329	723	3	44	315	27
2	0	727	5725	145	9	545	0
3	0	3	234	234	0	78	0
4	5	1817	56	0	11	10	0
5	0	813	1696	11	3	950	0
6	1784	40	0	0	0	0	2278

```
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score

svm = LinearSVC(
    C=1.0,
    max_iter=5000,
    random_state=42
)

svm.fit(X_train, y_train)
```

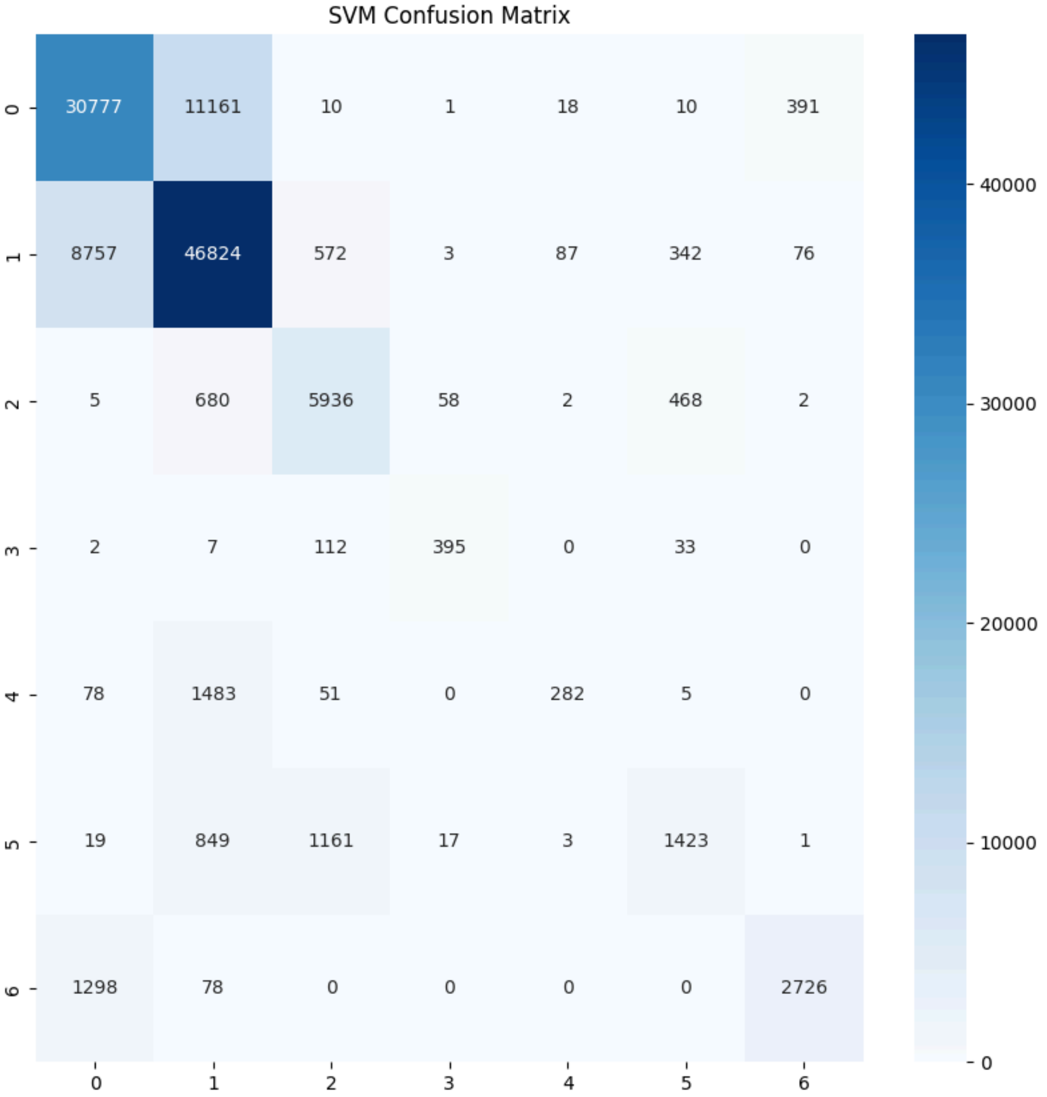
```
▾ LinearSVC
LinearSVC(max_iter=5000, random_state=42)
```

```
from sklearn.kernel_approximation import RBFSampler
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline

model = Pipeline([
    ("rbf", RBFSampler(gamma=0.1, n_components=300)),
    ("svm", LinearSVC(C=10.0, max_iter=8000))
])
model.fit(X_train, y_train)
```

```
▸ Pipeline
  ▸ RBFSampler
  ▸ LinearSVC
```

```
y_pred_svm = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred_svm))
print(classification_report(y_test, y_pred_svm))

plt.figure(figsize=(10, 10))
sns.heatmap(confusion_matrix(y_test, y_pred_svm), annot=True, fmt='d', cmap='Blues')
plt.title('SVM Confusion Matrix')
plt.show()
```

```
Accuracy: 0.760419266284003
              precision    recall  f1-score   support

           1       0.75      0.73      0.74     42368
           2       0.77      0.83      0.80     56661
           3       0.76      0.83      0.79      7151
           4       0.83      0.72      0.77       549
           5       0.72      0.15      0.25      1899
           6       0.62      0.41      0.49      3473
           7       0.85      0.66      0.75      4102

    accuracy                           0.76    116203
   macro avg       0.76      0.62      0.66    116203
weighted avg       0.76      0.76      0.75    116203
```



```
from sklearn.neural_network import MLPClassifier

mlp = MLPClassifier(
    hidden_layer_sizes=(256, 128, 64),
    activation='relu',
    solver='adam',
    alpha=0.0001,
    batch_size=256,
    learning_rate='adaptive',
    max_iter=60,
    random_state=42,
    verbose=True
)

mlp.fit(X_train, y_train)
```

```
Iteration 1, loss = 0.52199600
Iteration 2, loss = 0.39329957
Iteration 3, loss = 0.34517935
Iteration 4, loss = 0.31320149
Iteration 5, loss = 0.28869768
Iteration 6, loss = 0.27093971
Iteration 7, loss = 0.25723624
Iteration 8, loss = 0.24592746
Iteration 9, loss = 0.23718298
Iteration 10, loss = 0.22789859
Iteration 11, loss = 0.22116977
Iteration 12, loss = 0.21274346
Iteration 13, loss = 0.20900019
Iteration 14, loss = 0.20238444
Iteration 15, loss = 0.19777628
Iteration 16, loss = 0.19362906
Iteration 17, loss = 0.18990266
Iteration 18, loss = 0.18509880
Iteration 19, loss = 0.18225518
Iteration 20, loss = 0.17881240
Iteration 21, loss = 0.17541104
Iteration 22, loss = 0.17194500
Iteration 23, loss = 0.17081550
Iteration 24, loss = 0.16749915
Iteration 25, loss = 0.16601883
Iteration 26, loss = 0.16347969
Iteration 27, loss = 0.16164925
Iteration 28, loss = 0.15832712
Iteration 29, loss = 0.15698567
Iteration 30, loss = 0.15563752
Iteration 31, loss = 0.15356033
Iteration 32, loss = 0.15190098
Iteration 33, loss = 0.15045740
Iteration 34, loss = 0.14927976
Iteration 35, loss = 0.14755092
Iteration 36, loss = 0.14625432
Iteration 37, loss = 0.14523078
Iteration 38, loss = 0.14384845
Iteration 39, loss = 0.14161750
Iteration 40, loss = 0.14075813
Iteration 41, loss = 0.14054031
Iteration 42, loss = 0.13768509
Iteration 43, loss = 0.13767734
Iteration 44, loss = 0.13658039
Iteration 45, loss = 0.13536043
Iteration 46, loss = 0.13516526
Iteration 47, loss = 0.13283774
Iteration 48, loss = 0.13174029
Iteration 49, loss = 0.13150904
Iteration 50, loss = 0.13172712
Iteration 51, loss = 0.13025835
```