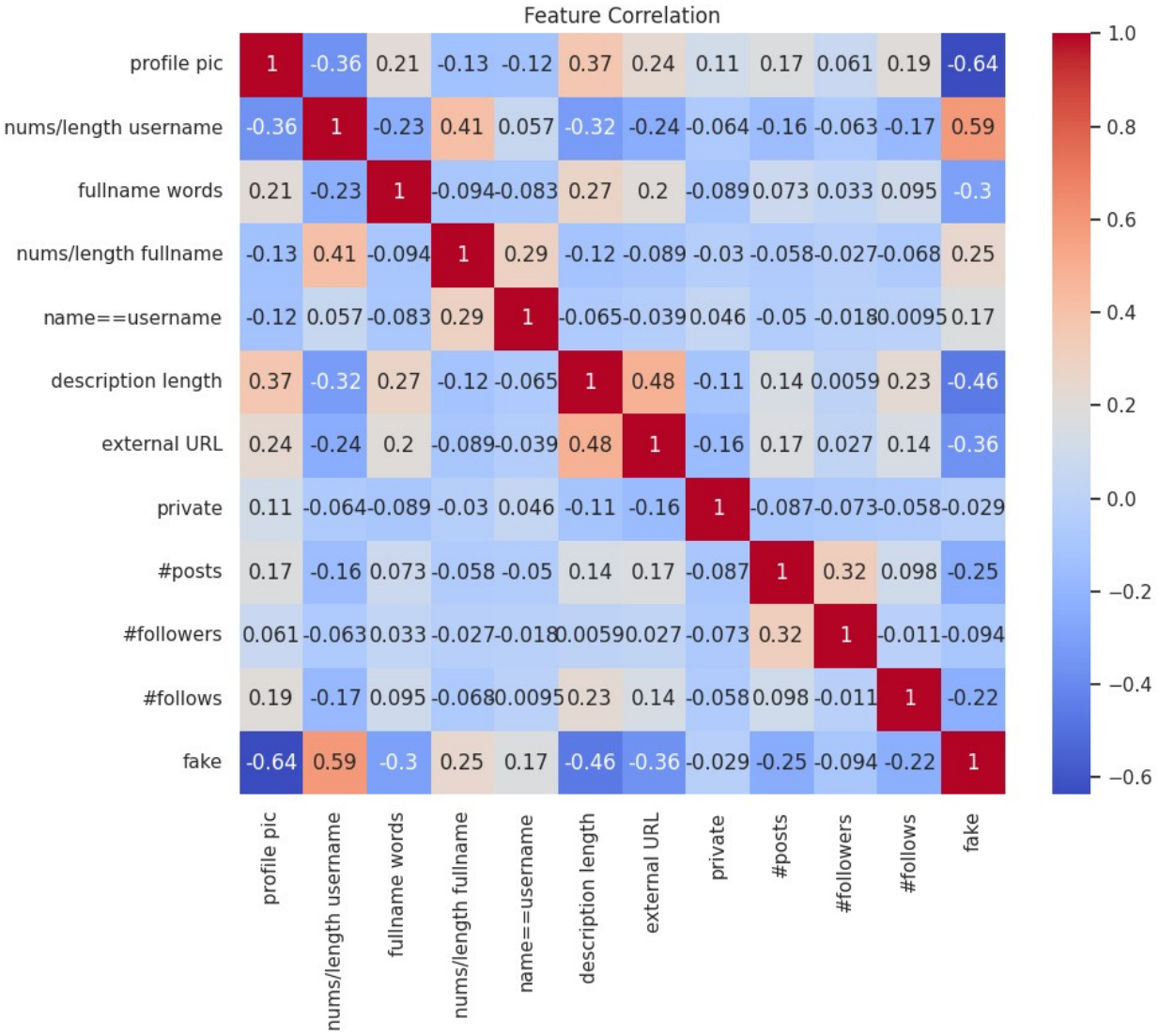
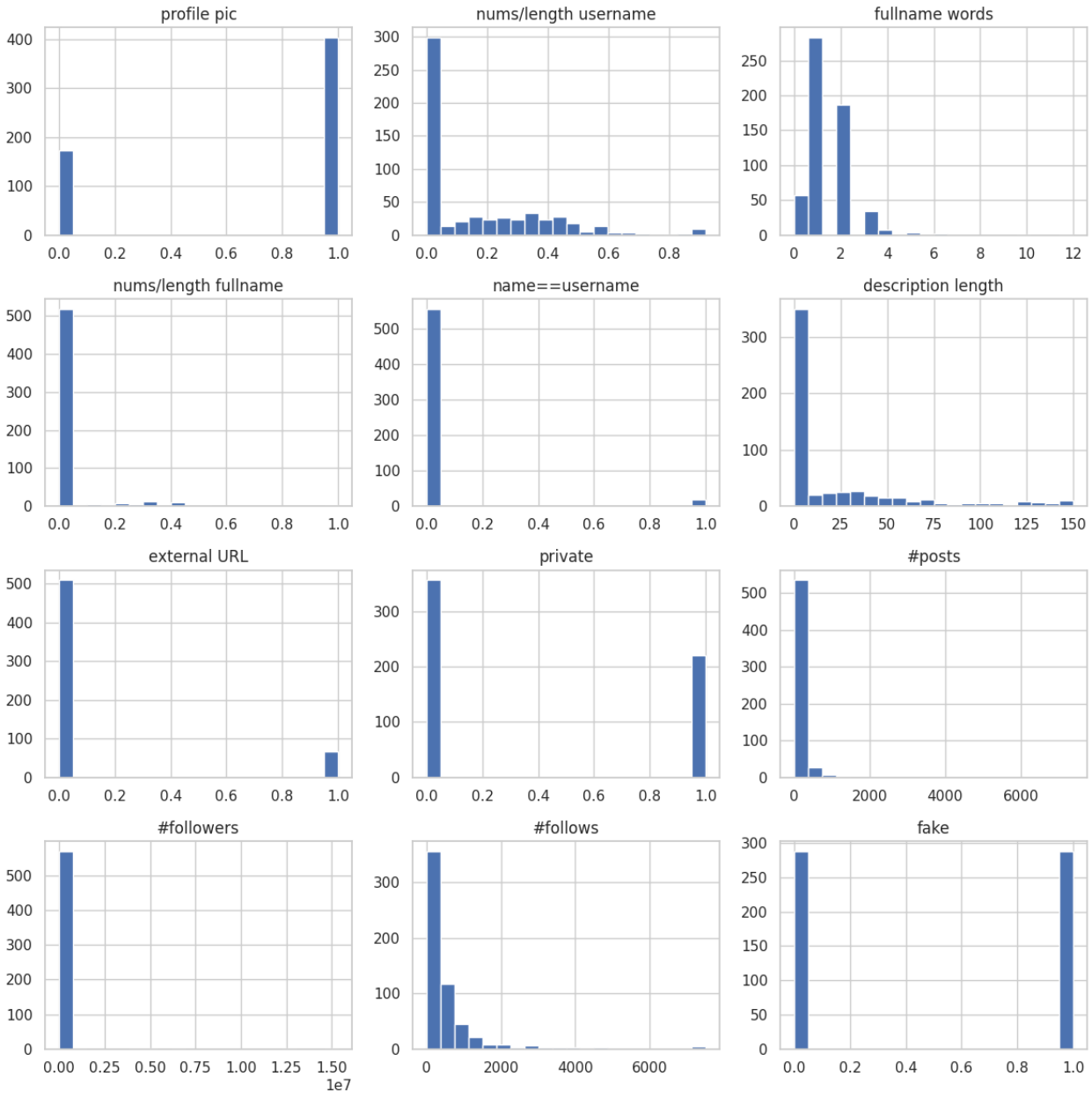
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pandas.plotting import scatter_matrix
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

sns.set(style='whitegrid')

train = pd.read_csv('/content/drive/MyDrive/train.csv')
test = pd.read_csv('/content/drive/MyDrive/test.csv')

plt.figure(figsize=(6,4))
sns.countplot(x='fake', data=train)
plt.title('Fake vs Genuine Accounts')
plt.show()
```

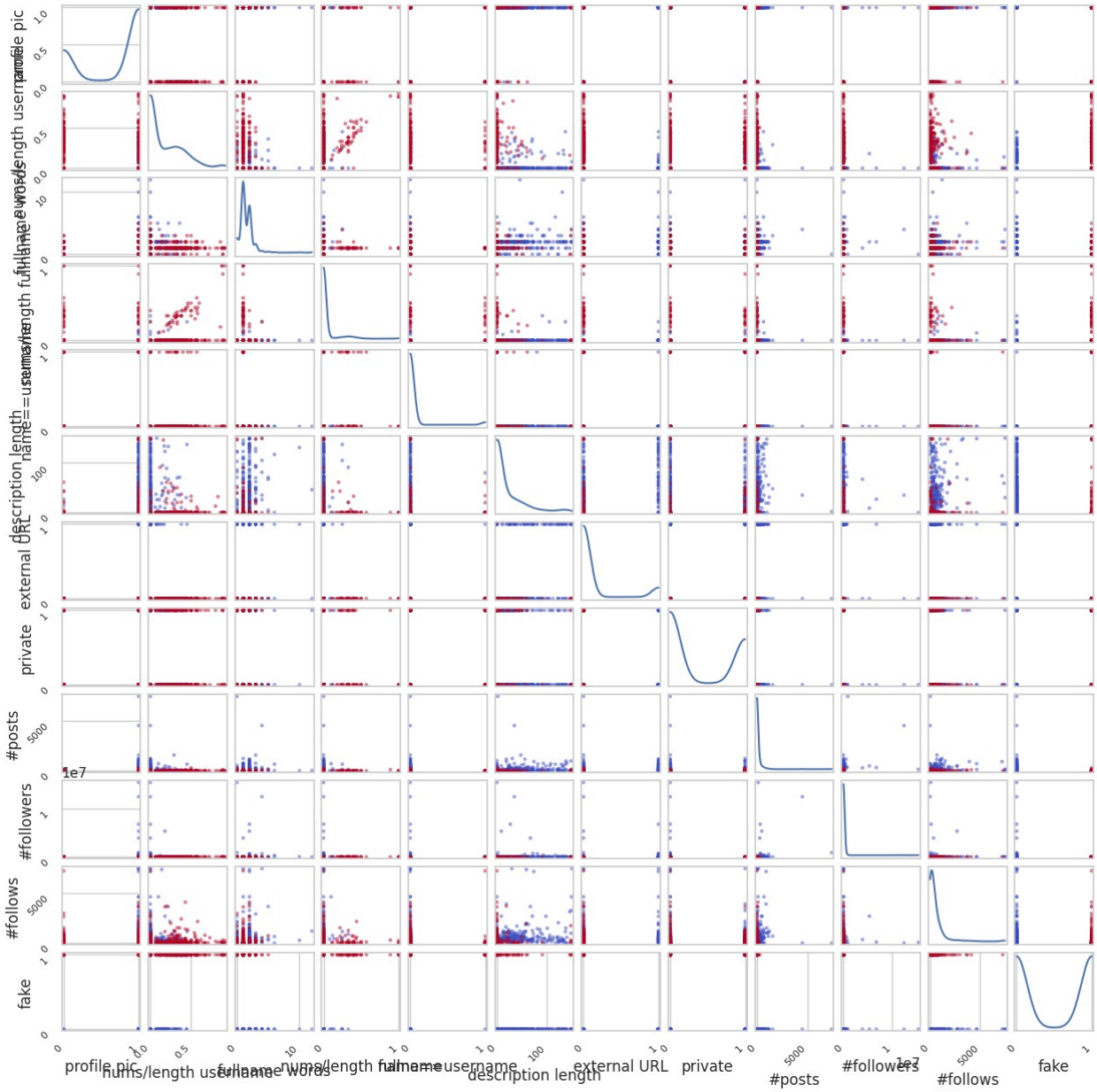

```
plt.figure(figsize=(10,8))
sns.heatmap(train.corr(), annot=True, cmap='coolwarm')
plt.title('Feature Correlation')
plt.show()
```




```
train.hist(figsize=(12,12), bins=20)
plt.tight_layout()
plt.show()
```



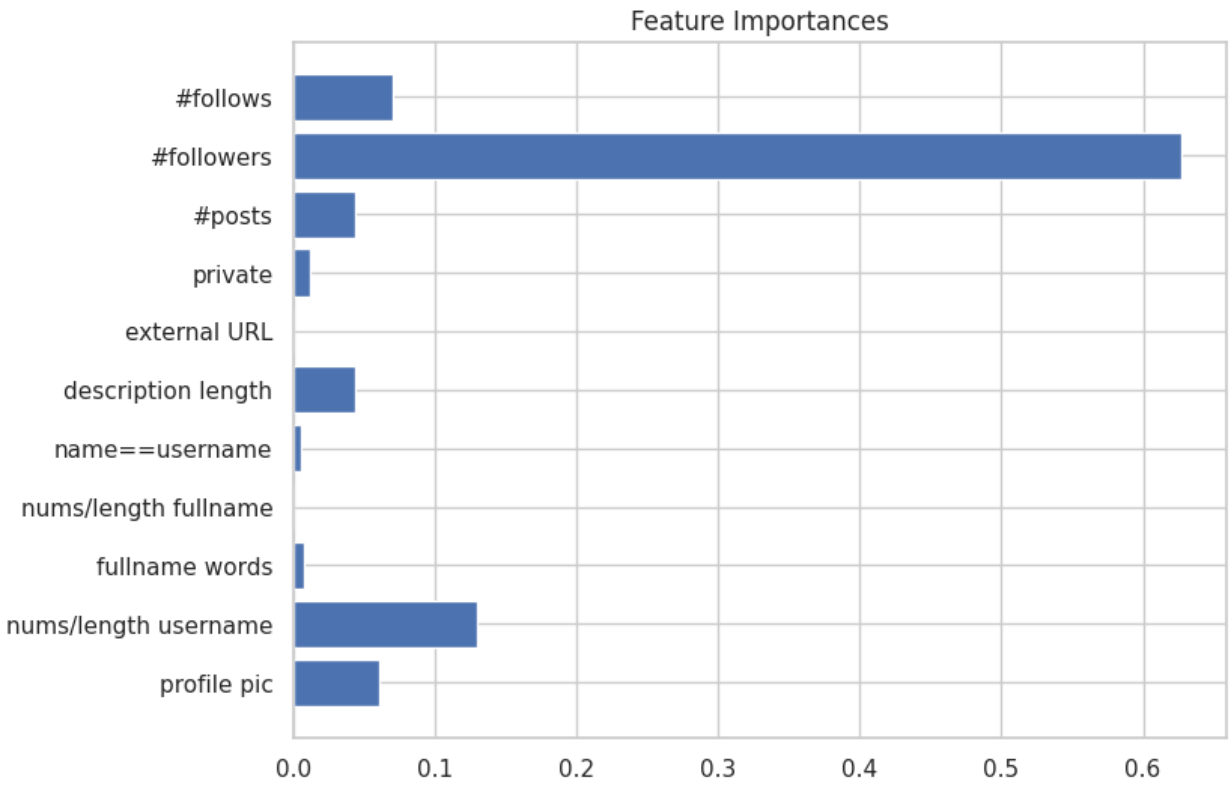

```
scatter_matrix(train, figsize=(15,15), diagonal='kde',
c=train['fake'], cmap='coolwarm')
plt.subplots_adjust(wspace=0.1, hspace=0.1)
for ax in plt.gcf().axes:
    plt.sca(ax)
    plt.xticks(rotation=45, ha='right')
    plt.yticks(rotation=45, va='top')
plt.show()
```




```
model = DecisionTreeClassifier(random_state=42)
X_train = train.drop('fake', axis=1)
y_train = train['fake']
X_test = test.drop('fake', axis=1)
y_test = test['fake']

model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print('Accuracy:', accuracy_score(y_test, y_pred))
print('Confusion Matrix:\n', confusion_matrix(y_test, y_pred))
print('Classification Report:\n', classification_report(y_test,
y_pred))
```

```
Accuracy: 0.8916666666666667
Confusion Matrix:
 [[55  5]
 [ 8 52]]
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.92      0.89        60
           1       0.91      0.87      0.89        60

    accuracy                           0.89       120
   macro avg       0.89      0.89      0.89       120
weighted avg       0.89      0.89      0.89       120
```

```
plt.figure(figsize=(8,6))
plt.barh(X_train.columns, model.feature_importances_)
plt.title('Feature Importances')
plt.show()
```




```
plt.figure(figsize=(20,20))
plot_tree(model, filled=True, feature_names=X_train.columns,
class_names=['genuine','fake'], rounded=True)
plt.show()
```