Project Name: Social Media Engagment Prediction

Develop By: Rudra Rathod

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
df = pd.read_csv('SME.csv')
df.head()
```

```
        post_id            timestamp day_of_week   platform
user_id  \
0  kcqbs6hxybia  2024-12-09 11:26:15      Monday  Instagram
user_52nwb0a6
1  vkmervg4ioos  2024-07-28 19:59:26      Sunday    Twitter
user_ucryct98
2  memhx4o1x6yu  2024-11-23 14:00:12    Saturday     Reddit
user_7rrev126
3  bhyo6piijqt9  2024-09-16 04:35:25      Monday    YouTube
user_4mxuq0ax
4  c9dkiomowakt  2024-09-05 21:03:01    Thursday    Twitter
user_l1vpox2k

               location language  \
0  Melbourne, Australia       pt
1          Tokyo, Japan       ru
2        Beijing, China       ru
3        Lagos, Nigeria       en
4       Berlin, Germany       hi

                                        text_content  \
0  Just tried the Chromebook from Google. Best pu...
1  Just saw an ad for Microsoft Surface Laptop du...
2  What's your opinion about Nike's Epic React?  ...
3  Bummed out with my new Diet Pepsi from Pepsi! ...
4  Just tried the Corolla from Toyota. Absolutely...

                   hashtags                          mentions  ...  \
```

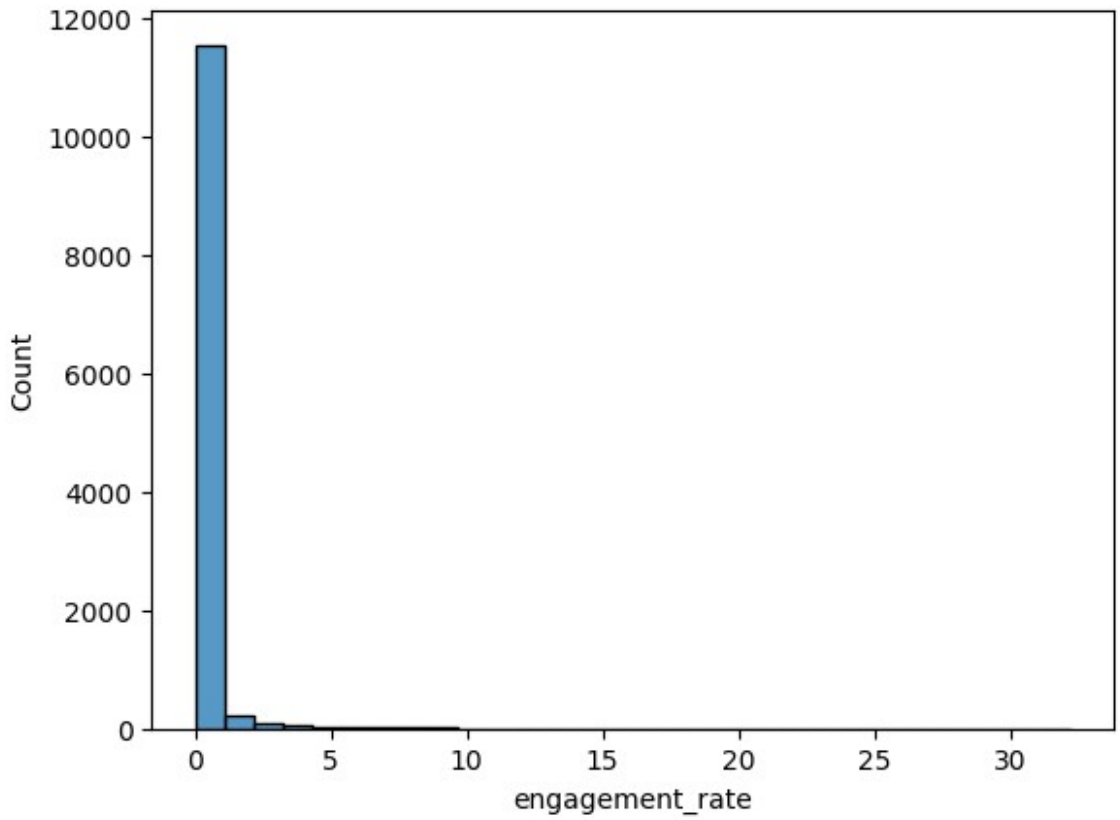
```
0                      #Food                                NaN  ...
1           #MustHave, #Food     @CustomerService, @BrandCEO  ...
2  #Promo, #Food, #Trending                                NaN  ...
3    #Reviews, #Sustainable      @StyleGuide, @BrandSupport  ...
4           #Health, #Travel  @BrandSupport, @InfluencerName  ...

  comments_count impressions  engagement_rate brand_name
product_name  \
0            701       18991          0.19319     Google
Chromebook
1            359       52764          0.05086  Microsoft  Surface
Laptop
2            643        8887          0.45425       Nike      Epic
React
3            743        6696          0.42293      Pepsi      Diet
Pepsi
4            703       47315          0.08773     Toyota
Corolla

      campaign_name  campaign_phase  user_past_sentiment_avg  \
0       BlackFriday          Launch                   0.0953
1      PowerRelease     Post-Launch                   0.1369
2       BlackFriday     Post-Launch                   0.2855
3        LaunchWave          Launch                  -0.2094
4  LocalTouchpoints          Launch                   0.6867

   user_engagement_growth  buzz_change_rate
0                 -0.3672              19.1
1                 -0.4510             -42.6
2                 -0.4112              17.4
3                 -0.0167              -5.5
4                  0.0807              38.8

[5 rows x 28 columns]
```

Data Summary and Visualization

```
print(df.info())
print(df.isnull().sum())
print(df.describe())
sns.histplot(df['engagement_rate'], bins=30)
plt.show()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12000 entries, 0 to 11999
Data columns (total 28 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
```

```
 0   post_id                   12000 non-null  object
 1   timestamp                 12000 non-null  object
 2   day_of_week               12000 non-null  object
 3   platform                  12000 non-null  object
 4   user_id                   12000 non-null  object
 5   location                  12000 non-null  object
 6   language                  12000 non-null  object
 7   text_content              12000 non-null  object
 8   hashtags                  12000 non-null  object
 9   mentions                  8059 non-null   object
 10  keywords                  12000 non-null  object
 11  topic_category            12000 non-null  object
 12  sentiment_score           12000 non-null  float64
 13  sentiment_label           12000 non-null  object
 14  emotion_type              12000 non-null  object
 15  toxicity_score            12000 non-null  float64
 16  likes_count               12000 non-null  int64
 17  shares_count              12000 non-null  int64
 18  comments_count            12000 non-null  int64
 19  impressions               12000 non-null  int64
 20  engagement_rate           12000 non-null  float64
 21  brand_name                12000 non-null  object
 22  product_name              12000 non-null  object
 23  campaign_name             12000 non-null  object
 24  campaign_phase            12000 non-null  object
 25  user_past_sentiment_avg   12000 non-null  float64
 26  user_engagement_growth    12000 non-null  float64
 27  buzz_change_rate          12000 non-null  float64
dtypes: float64(6), int64(4), object(18)
memory usage: 2.6+ MB
None
post_id                        0
timestamp                      0
day_of_week                    0
platform                       0
user_id                        0
location                       0
language                       0
text_content                   0
hashtags                       0
mentions                    3941
keywords                       0
topic_category                 0
sentiment_score                0
sentiment_label                0
emotion_type                   0
toxicity_score                 0
likes_count                    0
shares_count                   0
```

```
comments_count                0
impressions                   0
engagement_rate               0
brand_name                    0
product_name                  0
campaign_name                 0
campaign_phase                0
user_past_sentiment_avg       0
user_engagement_growth        0
buzz_change_rate              0
dtype: int64
       sentiment_score  toxicity_score  likes_count  shares_count  \
count     12000.000000    12000.000000  12000.00000  12000.000000
mean          0.000553        0.503868   2490.72025   1007.167167
std           0.583563        0.288198   1441.53253    575.072282
min          -0.999800        0.000000      0.00000      0.000000
25%          -0.503200        0.251400   1236.00000    510.000000
50%          -0.006200        0.505950   2496.00000   1018.000000
75%           0.513525        0.756200   3723.25000   1501.000000
max           0.999900        0.999900   5000.00000   2000.000000

       comments_count   impressions  engagement_rate
user_past_sentiment_avg  \
count     12000.00000  12000.000000     12000.000000
12000.000000
mean        504.34575  49811.338500         0.278137
0.001472
std         288.68416  28930.289451         1.149206
0.576627
min           0.00000    130.000000         0.001880                -
0.999600
25%         253.00000  24716.500000         0.049100                -
0.495975
50%         503.00000  49674.000000         0.080605
0.001950
75%         755.00000  74815.000000         0.163123
0.501725
max        1000.00000  99997.000000        32.211710
0.999400

       user_engagement_growth  buzz_change_rate
count            12000.000000      12000.000000
mean                 0.000998          0.729692
std                  0.289940         57.787219
min                 -0.499900        -99.900000
25%                 -0.248400        -48.700000
50%                  0.002800          0.900000
75%                  0.250700         50.100000
max                  0.499900         99.900000
```



Drop rows with missing target or key columns

```
df = df.dropna(subset=['engagement_rate', 'text_content'])
```

Label encode categorical columns

```
label_encoders = {}
categorical_cols = ['platform', 'day_of_week', 'topic_category']
for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))
    label_encoders[col] = le
```

Model Training and Testing

```
tfidf = TfidfVectorizer(max_features=5000, stop_words='english')
X_text = tfidf.fit_transform(df['text_content'].astype(str))
```

```
numeric_features = ['platform', 'day_of_week', 'topic_category',
'likes_count', 'shares_count', 'comments_count', 'impressions',
'sentiment_score', 'toxicity_score', 'user_past_sentiment_avg',
'user_engagement_growth', 'buzz_change_rate']
X_numeric = df[numeric_features].fillna(0)
scaler = StandardScaler()
X_numeric_scaled = scaler.fit_transform(X_numeric)
from scipy.sparse import hstack
X = hstack([X_numeric_scaled, X_text])
y = df['engagement_rate']

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

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

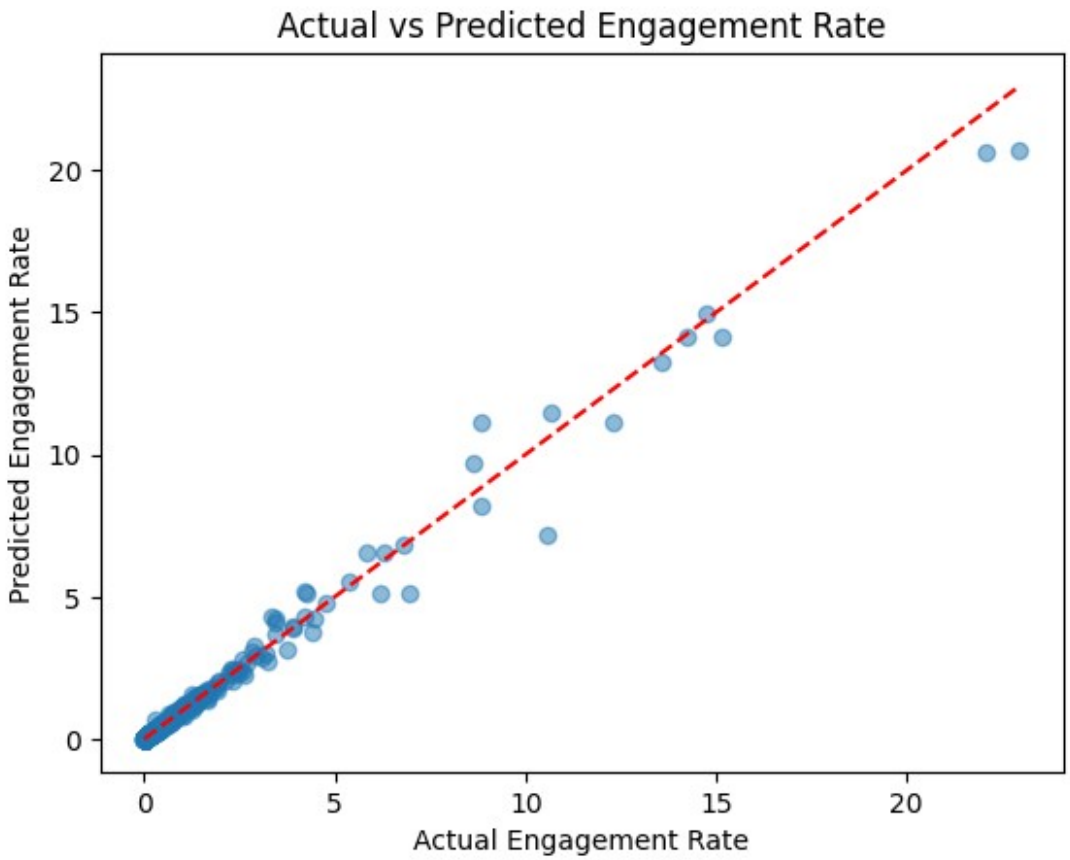
```
RandomForestRegressor(random_state=42)
```

```
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print(f'RMSE: {rmse}')
print(f'R^2 Score: {r2}')
```

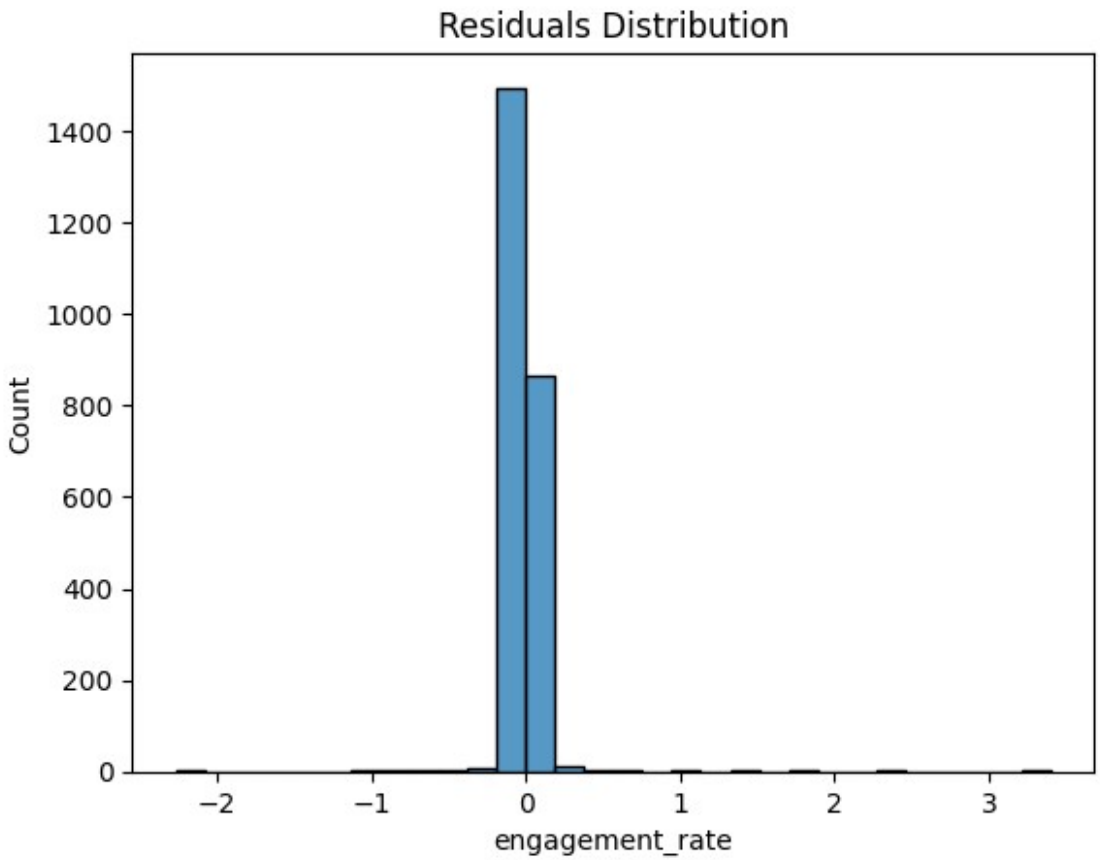
```
RMSE: 0.1302534098174815
R^2 Score: 0.9865717632999228
```

Plotting Actual vs Predcited Engagement Rate

```
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
'r--')  # Diagonal line
plt.xlabel('Actual Engagement Rate')
plt.ylabel('Predicted Engagement Rate')
plt.title('Actual vs Predicted Engagement Rate')
plt.show()
```

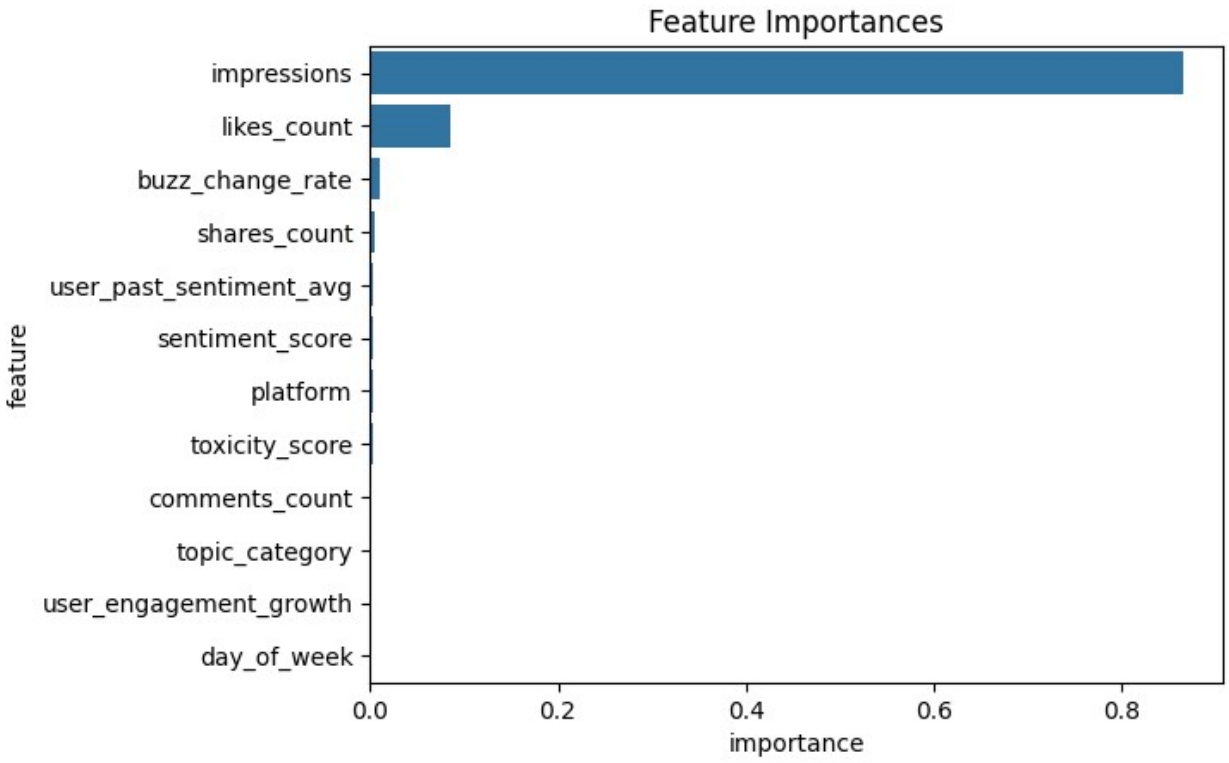



```
residuals = y_test - y_pred
sns.histplot(residuals, bins=30)
plt.title('Residuals Distribution')
plt.show()
```



Feature Importance

```
importances = model.feature_importances_
feature_names = numeric_features
importance_df = pd.DataFrame({'feature': feature_names, 'importance':
importances[:len(feature_names)]})
importance_df = importance_df.sort_values(by='importance',
ascending=False)
sns.barplot(data=importance_df, x='importance', y='feature')
plt.title('Feature Importances')
plt.show()
```



Model Saving

```
import joblib
joblib.dump(model, 'engagement_model.pkl')
joblib.dump(tfidf, 'tfidf_vectorizer.pkl')
joblib.dump(scaler, 'scaler.pkl')
joblib.dump(label_encoders, 'label_encoders.pkl')
```

```
['label_encoders.pkl']
```