2024

Report

ML Milestone 1_Report

Team: SC_33

1. عبدالرحمن حمدى عبدالتواب احمد
   20201701096
2. عبدالرحمن اسامه عبدالكريم محمود
   20201701088
3. عبدالرحمن مجدي أحمد رزق
   2021170302
4. حبيبه احمد عبدالقادر السيد
   20201700218
5. زياد احمد السيد حسن ابراهيم
   20201700299

⚙ Preprocessing:

1- We need to remove the outliers from the dataset so it doesn't affect on the model ,and replaced the outliers with the mean of each column like in the Pic.

```
def detect_outliers_z_score(data):
    columns_to_check = [col for col in DF.columns
                        if col not in ['url', 'title', 'channel type', 'weekday', 'isWeekEnd']]
    outliers = {}
    threshold = 3
    for col in data.columns:
        if col in columns_to_check:
            z_scores = np.abs(stats.zscore(data[col]))
            outliers[col] = data.index[z_scores > threshold].tolist()
    return outliers


outliers = detect_outliers_z_score(DF)
for col, indices in outliers.items():
    for index in indices:
        DF.at[index, col] = DF[col].mean()
```

2- We applied scaling(Min-Max) factor to all the data so we don't have any zero's or negative values in the data set and to increase the model performance.

```
columns_to_scale = [col for col in DF.columns
                    if col not in ['url', 'title', 'channel type', 'weekday', 'isWeekEnd']]

scaler = MinMaxScaler()

DF[columns_to_scale] = scaler.fit_transform(DF[columns_to_scale])
```

3- We mapped the columns with string statements to float values so all the data set will be numerical Because the model can't Handle the string data.

```
channel_type_mapping = {
    ' data_channel_is_bus': 0.0,
    ' data_channel_is_socmed': 1.0,
    ' data_channel_is_lifestyle': 2.0,
    ' data_channel_is_world': 3.0,
    ' data_channel_is_entertainment': 4.0,
    ' data_channel_is_tech': 5.0,
    '[]': 6.0
}
weekday_mapping = {
    'monday': 0.0,
    'tuesday': 1.0,
    'wednesday': 2.0,
    'thursday': 3.0,
    'friday': 4.0,
    'saturday': 5.0,
    'sunday': 6.0
}
isWeekEnd = {
    'No': 0.0,
    'Yes': 1.0
}
```

```
DF['weekday'] = DF['weekday'].map(weekday_mapping)
DF['channel type'] = DF['channel type'].map(channel_type_mapping)
DF['isWeekEnd'] = DF['isWeekEnd'].map(isWeekEnd)
```

Analysis

- after performing some technique on the data the data hadn’t any **NULL** values .
- by apply correlation on the data some columns has some relationship and some are not but the relationship with the target “shares” not that powerful because correlation did not reach 0.3 .
- so , we use all the data except 3 columns “Title , URL, timedelta” because the first two will not really be useful because unique string and didn’t treat with string and third when we drop it , the mse has be better .

```
DF = DF.drop(columns=['title', 'url', ' timedelta'])
corr = DF.corr()
top_feature = corr.index[abs(corr[' shares']) > 0.1]
top_corr = DF[top_feature].corr()
```

Models

- we use two model regression one linear regression and the other polynomial regression

```
# Linear Regression Model
sln = linear_model.LinearRegression()
sln.fit(X_train, Y_train)
y_predicted = sln.predict(X_train)
prediction = sln.predict(X_test)
```

```
poly_features = PolynomialFeatures(degree=2)
X_train_poly = poly_features.fit_transform(X_train)
poly_model = linear_model.LinearRegression()
poly_model.fit(X_train_poly, Y_train)
y_train_predicted = poly_model.predict(X_train_poly)
y_test_predicted = poly_model.predict(poly_features.transform(X_test))
```

- the difference between the two is when applying polynomial regression the difference was not bad at all with polynomial regression of degree 2**.**
- we didn't increase the degree because overfitting and we need a much more powerful processor like **server** for this much data.
- we depend on the correlation of the data with the target and also Wrapper Methods for decided what feature the negative effect on the models .
- with also test size 10% from the data and 90% for train and the test was small but it make a better mse and we didn't use validation.

```
--- Linear Regression ---
Mean Square Error Train Model 1 :  2.0431010208321535e-05
Mean Square Error Test Model 1 :  1.5960432544826525e-06
--- Polynomial Regression ---
Mean Square Error Train Model 2 :  1.936162398298325e-05
Mean Square Error Test Model 2 :  6.777349922801626e-07

Process finished with exit code 0
```

Conclusion

This phase has disappointment us because really bad data and with mse such bad with regression and this data has no regression look ever .