Video Likes Prediction team(SC-15)

Milestone 1

- ## Preprocessing:

  - **Drop rows contain null values.**

Features	Preprocessing technique
Video id, Channel_title	Category encoding technique
trending_date,publish_time	we will make a new column (days_to _be_trend)resulting from subtracting the trending_date from publish_time and insert this column(days_to _be_trend)to dataset and drop these colmns(trending_date,publish_time) from dataset
Title, Tags , Video_description	➢ convert to lowercase ➢ remove Special Characters ➢ remove Single Characters ➢ remove Single Characters from the start ➢ Replace multiple spaces with single space Removing prefixed 'b' ➢ Removing links ➢ Applying natural language processing(TfidfVectorizer)

Comment_disapled, Rating_disabled, video_error_or_removed	Category encoding technique
Category_id, views,comment_count, likes,video_id,channel_title,days_to_be _trend	Normalization technique

- **Analysis:**

Apply correlation to dataset

- ➢ Likes depend on (The first is the most depend)
    1. views, comment_count
    2. Category_id ,days_to_be_trend
    3. Tags , Video_description

- **The sizes of your training, testing:**

  Split dataset to 30% -> test and 70%-> train and validation

- **Regression techniques:**
    - ➢ Polynomial Regression(degree = 2):
      Runtime of the train polynomial_regression degree=2 model is **0.06905579566955566**
      Runtime of the test polynomial_regression degree=2 model : **0.0**
      Model polynomial_regression degree=2 Cross Validation scores : **0.00012936835227556537**
      Model polynomial_regression degree=2 train Mean Square Error : **0.00012462695700895727**
      Model polynomial_regression degree=2 test Mean Square Error : **0.00013380101833585214**

    - ➢ Polynomial Regression(degree = 3):
      Runtime of the train polynomial_regression degree=3 model is **0.5636563301086426**

Runtime of the test polynomial_regression degree=3 model :
0.042963504791259766
Model polynomial_regression degree=3 Cross Validation scores :
323901330769653.8
Model polynomial_regression degree=3 train Mean Square Error :
9.917281305003476e-05
Model polynomial_regression degree=3 test Mean Square Error :
9.803103797561847e-05

- Polynomial Regression(degree = 4):
  Runtime of the train polynomial_regression degree=4 model is
  3.442033052444458
  Runtime of the test polynomial_regression degree=4 model :
  0.08992218971252441
  Model polynomial_regression degree=4 Cross Validation scores :
  9313511042294.768
  Model polynomial_regression degree=4 train Mean Square Error :
  0.0003788647859759009
  Model polynomial_regression degree=4 test Mean Square Error :
  0.0018258648095086402

- Polynomial Regression(degree = 5):(Overfitting)
  train_mean_square_error : 3.894829
  test_mean_square_error : 668837.39

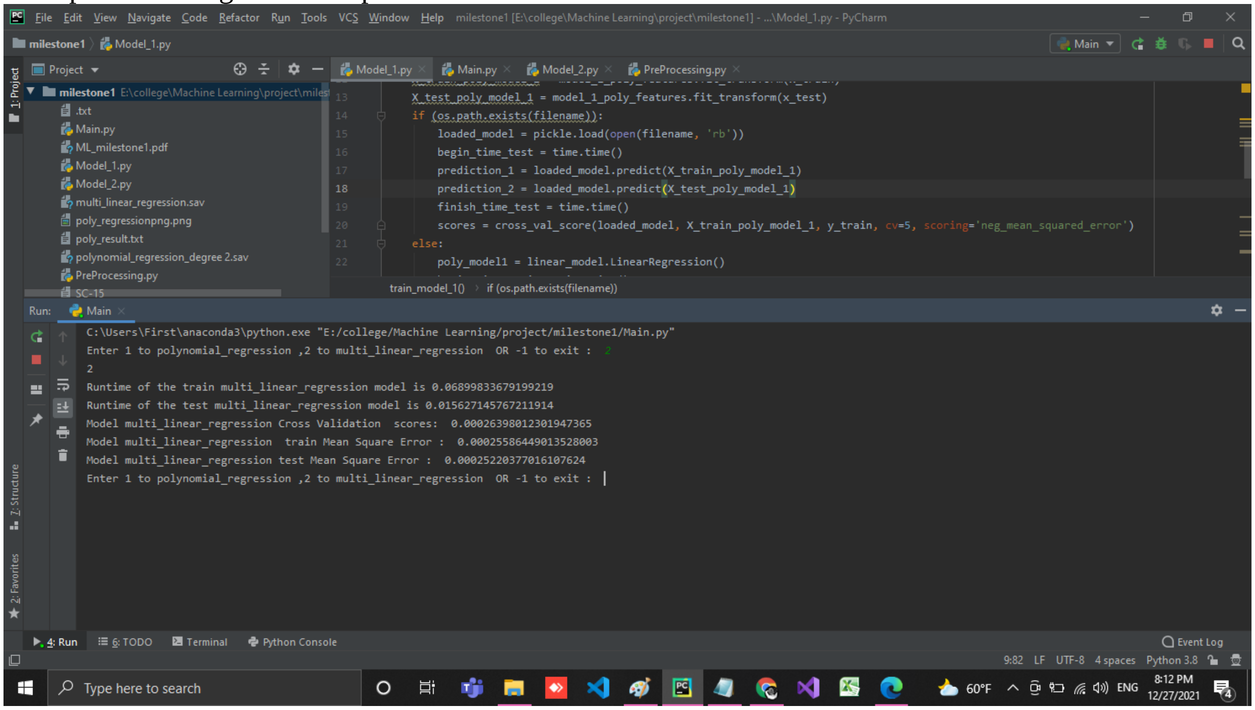
- Multiple Regression:
  Runtime of the train multi_linear_regression model is
  0.06899833679199219
  Runtime of the test multi_linear_regression model is
  0.015627145767211914
  Model multi_linear_regression Cross Validation scores:
  0.00026398012301947365
  Model multi_linear_regression train Mean Square Error :
  0.00025586449013528003
  Model multi_linear_regression test Mean Square Error :
  0.00025220377016107624

What we use:

We Use for Mode1 ->Polynomial Regression(deg = 2)
We Use for Mode2 ->Multiple Regression

Polynomial Regression is the best model .

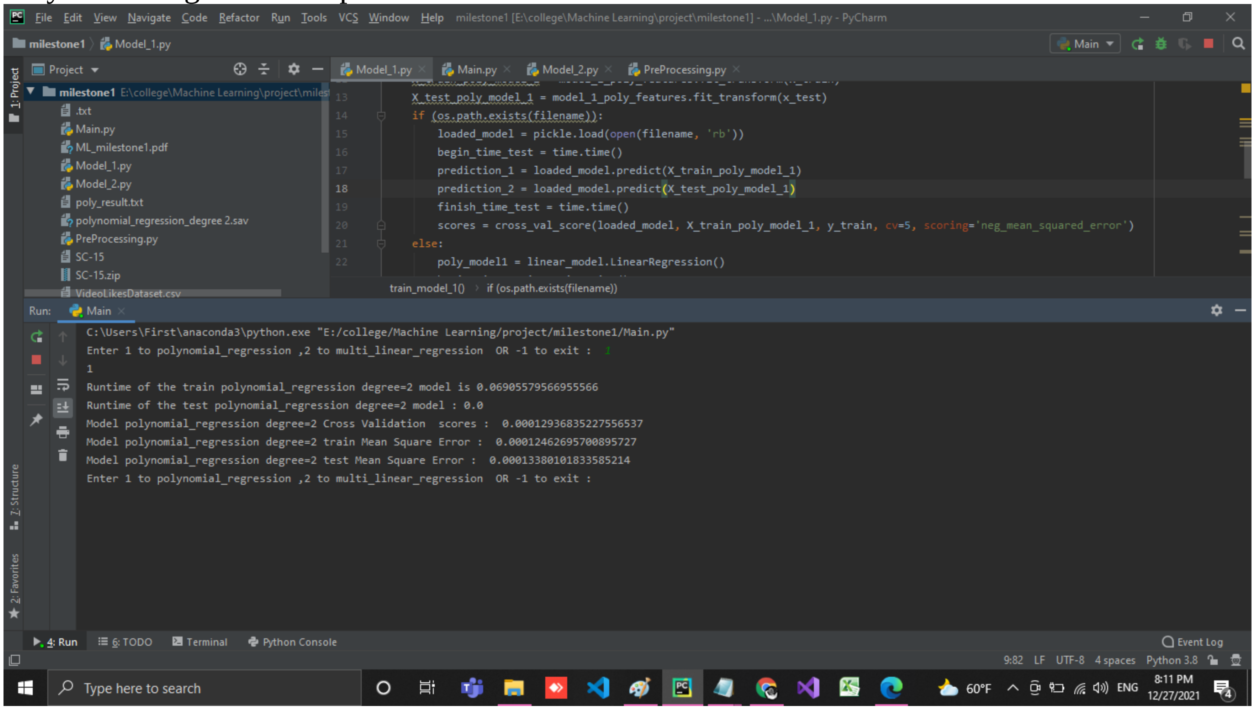
Multiple Linear regression output


File Edit View Navigate Code Refactor Run Tools VCS Window Help milestone1 [E:\college\Machine Learning\project\milestone1] - ...\Model_1.py - PyCharm
milestone1
Model_1.py
Main
Project
Model_1.py
Main.py
Model_2.py
PreProcessing.py
milestone1 E:\college\Machine Learning\project\milest
.txt
Main.py
ML_milestone1.pdf
Model_1.py
Model_2.py
multi_linear_regression.sav
poly_regressionpng.png
poly_result.txt
polynomial_regression_degree 2.sav
PreProcessing.py
SC-15
train_model_1() › if (os.path.exists(filename))
Run:
Main
4: Run
6: TODO
Terminal
Python Console
Event Log
9:82 LF UTF-8 4 spaces Python 3.8
Type here to search
60°F
ENG
8:12 PM
12/27/2021

```python
    X_test_poly_model_1 = model_1_poly_features.fit_transform(x_test)
    if (os.path.exists(filename)):
        loaded_model = pickle.load(open(filename, 'rb'))
        begin_time_test = time.time()
        prediction_1 = loaded_model.predict(X_train_poly_model_1)
        prediction_2 = loaded_model.predict(X_test_poly_model_1)
        finish_time_test = time.time()
        scores = cross_val_score(loaded_model, X_train_poly_model_1, y_train, cv=5, scoring='neg_mean_squared_error')
    else:
        poly_model1 = linear_model.LinearRegression()
```

```
C:\Users\First\anaconda3\python.exe "E:/college/Machine Learning/project/milestone1/Main.py"
Enter 1 to polynomial_regression ,2 to multi_linear_regression  OR -1 to exit :  2
2
Runtime of the train multi_linear_regression model is 0.06899833679199219
Runtime of the test multi_linear_regression model is 0.015627145767211914
Model multi_linear_regression Cross Validation  scores:  0.00026398012301947365
Model multi_linear_regression  train Mean Square Error :  0.00025586449013528003
Model multi_linear_regression test Mean Square Error :  0.00025220377016107624
Enter 1 to polynomial_regression ,2 to multi_linear_regression  OR -1 to exit : 
```

Polynomial regression output


File Edit View Navigate Code Refactor Run Tools VCS Window Help milestone1 [E:\college\Machine Learning\project\milestone1] - ...\Model_1.py - PyCharm
milestone1
Model_1.py
Main
Project
Model_1.py
Main.py
Model_2.py
PreProcessing.py
milestone1 E:\college\Machine Learning\project\milest
.txt
Main.py
ML_milestone1.pdf
Model_1.py
Model_2.py
poly_result.txt
polynomial_regression_degree 2.sav
PreProcessing.py
SC-15
SC-15.zip
VideoLikesDataset.csv
train_model_1() › if (os.path.exists(filename))
Run:
Main
4: Run
6: TODO
Terminal
Python Console
Event Log
9:82 LF UTF-8 4 spaces Python 3.8
Type here to search
60°F
ENG
8:11 PM
12/27/2021

```python
    X_test_poly_model_1 = model_1_poly_features.fit_transform(x_test)
    if (os.path.exists(filename)):
        loaded_model = pickle.load(open(filename, 'rb'))
        begin_time_test = time.time()
        prediction_1 = loaded_model.predict(X_train_poly_model_1)
        prediction_2 = loaded_model.predict(X_test_poly_model_1)
        finish_time_test = time.time()
        scores = cross_val_score(loaded_model, X_train_poly_model_1, y_train, cv=5, scoring='neg_mean_squared_error')
    else:
        poly_model1 = linear_model.LinearRegression()
```

```
C:\Users\First\anaconda3\python.exe "E:/college/Machine Learning/project/milestone1/Main.py"
Enter 1 to polynomial_regression ,2 to multi_linear_regression  OR -1 to exit :  1
1
Runtime of the train polynomial_regression degree=2 model is 0.06905579566955566
Runtime of the test polynomial_regression degree=2 model : 0.0
Model polynomial_regression degree=2 Cross Validation  scores :  0.00012936835227556537
Model polynomial_regression degree=2 train Mean Square Error :  0.00012462695700895727
Model polynomial_regression degree=2 test Mean Square Error :  0.00013380101833585214
Enter 1 to polynomial_regression ,2 to multi_linear_regression  OR -1 to exit :
```

Further techniques that were used to improve the results:

- Using Ridge Regularization To Avoide Overfitting.
- Using Text in Features To predict likes.

Milestone 2

Preprocessing:

Features	Preprocessing technique
Video id, Channel_title, Comment_disapled, Rating_disabled, video_error_or_removed, VideoPopularity	Category encoding technique
trending_date,publish_time	we will make a new column (days_to _be_trend)resulting from subtracting the trending_date from publish_time and insert this column(days_to _be_trend)to dataset and drop these colmns(trending_date,publish_time) from dataset

Null values:

fill null values with values of previous index of row

Analysis:

Apply correlation to dataset

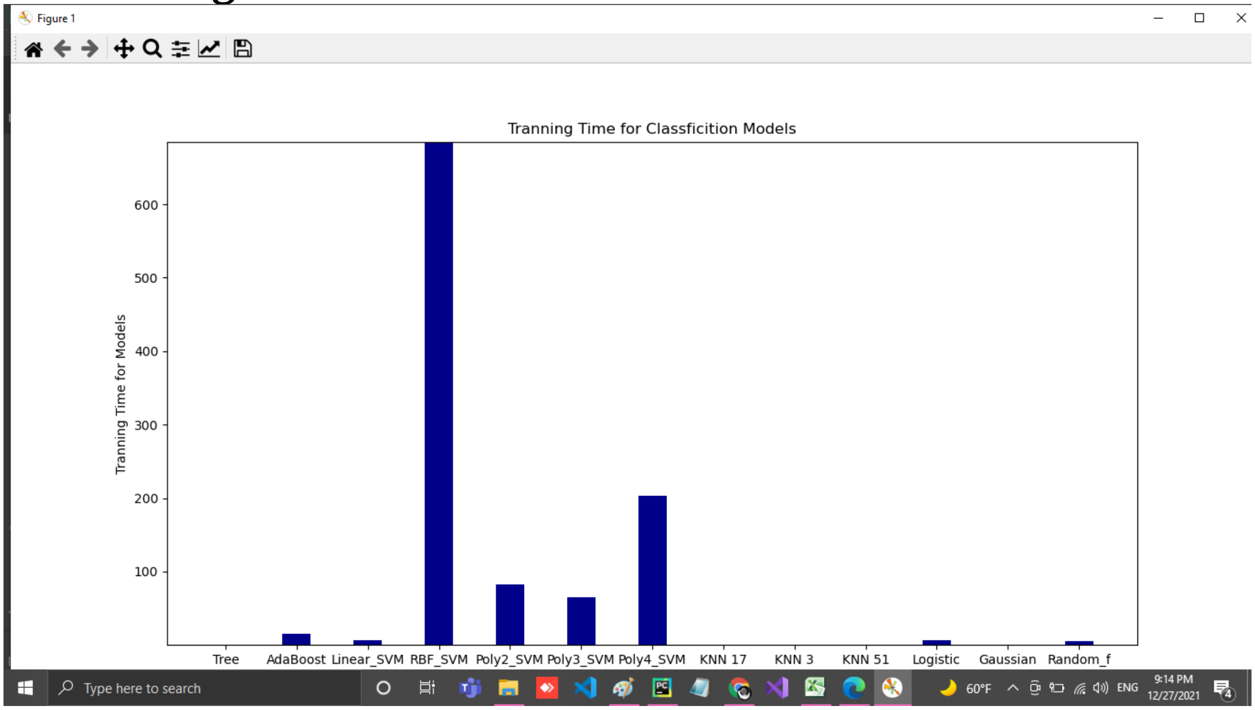
- **Likes depend on**
    4. Views, comment_count
    5. Category_id
    6. video_id
    7. channel_title
    8. video_error_or_removed**,** ratings_disabled**,** comments_disabled**,** days_to _be_trend

The sizes of training, testing:

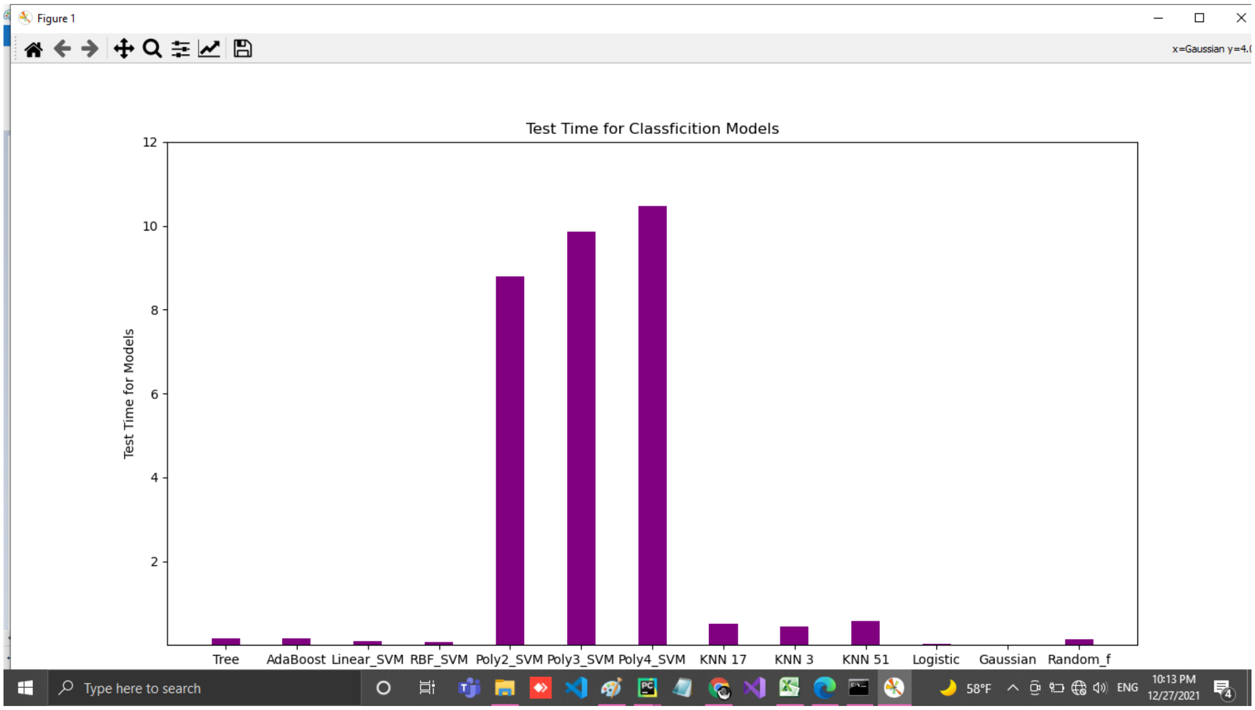
Split dataset to 20% -> test and 80%-> train and validation.

Techniques behavior summary:

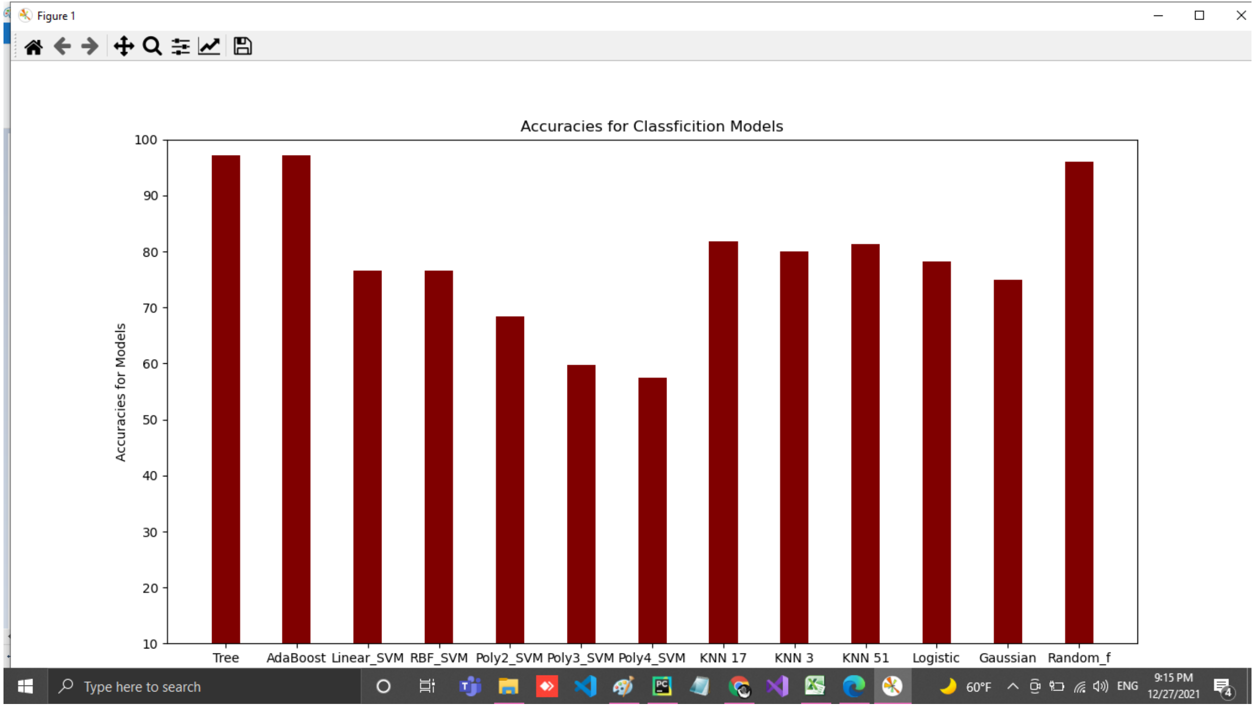
1-trainning time:



2-testing time



3-accuracy summary:



hyperparameter tuning affected :

1-**Decesion tree**

Hyper parameters	**Accuracy**
Max_depth=**12**	97.39
Max_depth=**10**	90.04
Max_depth=**6**	85.11
Max_depth=**3**	82.33
Max_depth=**1**	77.42

Model Tree Decision Test Mean Square Error : 0.124472

2-**Adaboost after DT**

Hyper parameters	**Accuracy**
Max_depth=**8**	97.32
Max_depth=**6**	96
Max_depth=**3**	82.04
Max_depth=**1**	78.16

Model AdaBoost with Tree Decision Test Mean Square Error : 0.040126

3-random forest

Hyper parameter	**accuracy**
Min-samples-leaf=**150**	86.19
Min-samples-leaf=**30**	90.7
Min-samples-leaf=**10**	93.6
Min-samples-leaf=**5**	94.9
Min-samples-leaf=**2**	95.92

Min-samples-leaf=**1**	96.14
n_estimators=100, oob_score=True, n_jobs=-1, random_state=101, max_features=None, min_samples_leaf=1	96.198

Model Random Forest Test Mean Square Error : 0.06019007391763464

4-KNN

Hyper parameter		Accuracy
K=**10**	Leaf-size=**200**	81.58
	Leaf-size=**100**	
	Leaf-size=**50**	
Leaf-size=**30**	K=**15**	81.9
	K=**3**	80.14
	K=**51**	81.41

Model KNN k=17 Test Mean Square Error : 0.299762

Model KNN k=3 Test Mean Square Error : .0.3256

Model KNN k=51 Test Mean Square Error : 0.31256

5-logistic regression

Hyper parameter	accurcay
C=5	78.2
C=10	78.16
C=20	78.22
solver='lbfgs', max_iter=800, C=0.1, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1,	78.23

multi_class='auto',
n_jobs=None, penalty='l2',
random_state=None, tol=0.0001,
verbose=0, warm_start=False	

Model Logistics regression Test Mean Square Error : 0.3554646251319958

6-svm

1- liner SVM (OneVsOneClassifier)

Hyper parameter		accuracy
C=10	max_iter=7000	73
	max_iter=5000	79.31
	max_iter=2000	74.53
	max_iter=1000	73.85
C=20	max_iter=1500	66.6
C=15		74.5
C=5		76.1

Model LinearSVC OneVsOne SVM Test Mean Square Error : 0.394667370

2- rbf

Hyper parameter		accuracy
C=1	Gamma=0.8	76.61
C=10	Gamma=5	52.71
C=10	Gamma=10	52.719
C=0.1	Gamma=0.8	52.6

Model SVC with RBF kernel Test Mean Square Error : 0.39466737064413

3-**polynomial SVM degree=2 kernal=poly**

Hyper parameter	accuracy
C=1	66.03
C=10	66.43
C=1000	68.45

Model SVC with polynomial kernel degree 2 Test Mean Square Error : 0.42832629355860613

4- **polynomial SVM degree=3 kernal=poly**

Hyper parameter	accuracy
C=1	56.1
C=500	59.72

Model SVC with polynomial kernel degree 3 Test Mean Square Error : 0.4949841605068638

5- **polynomial SVM degree=4 kernal=poly**

Hyper parameter	accuracy
C=1	55.66
C=500	57.51

Model SVC with polynomial kernel degree 4 Test Mean Square Error : 0.48270855332629353

7- GaussianNB

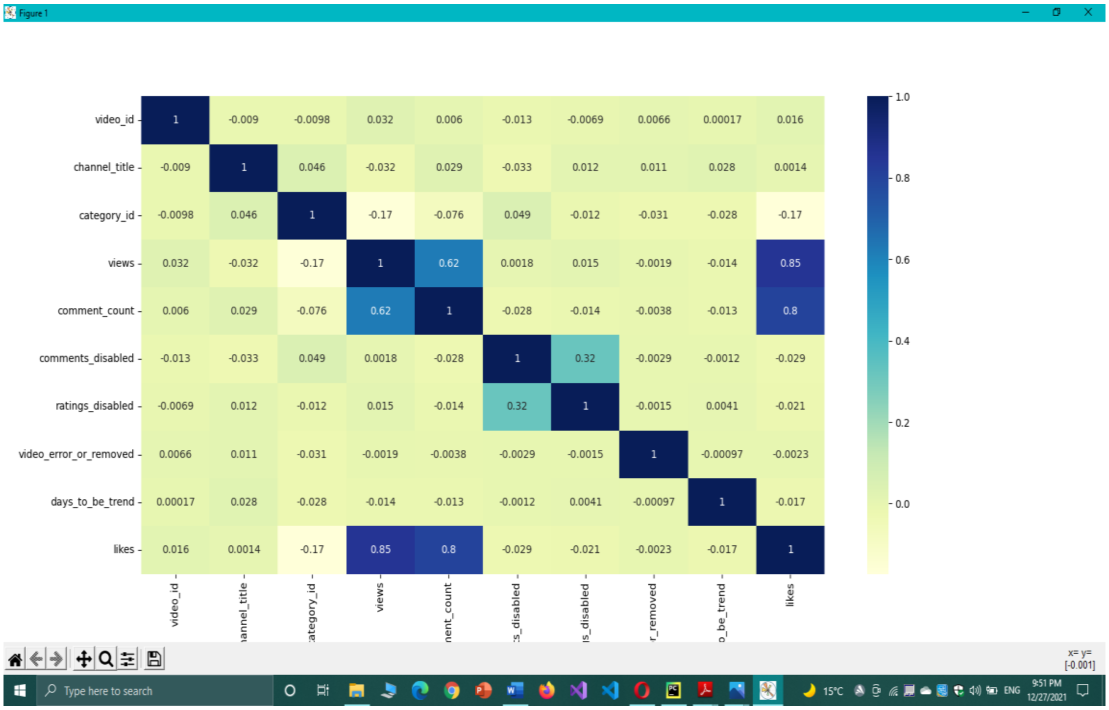
Mean square error :0.3636483

Accuracy:74.9604

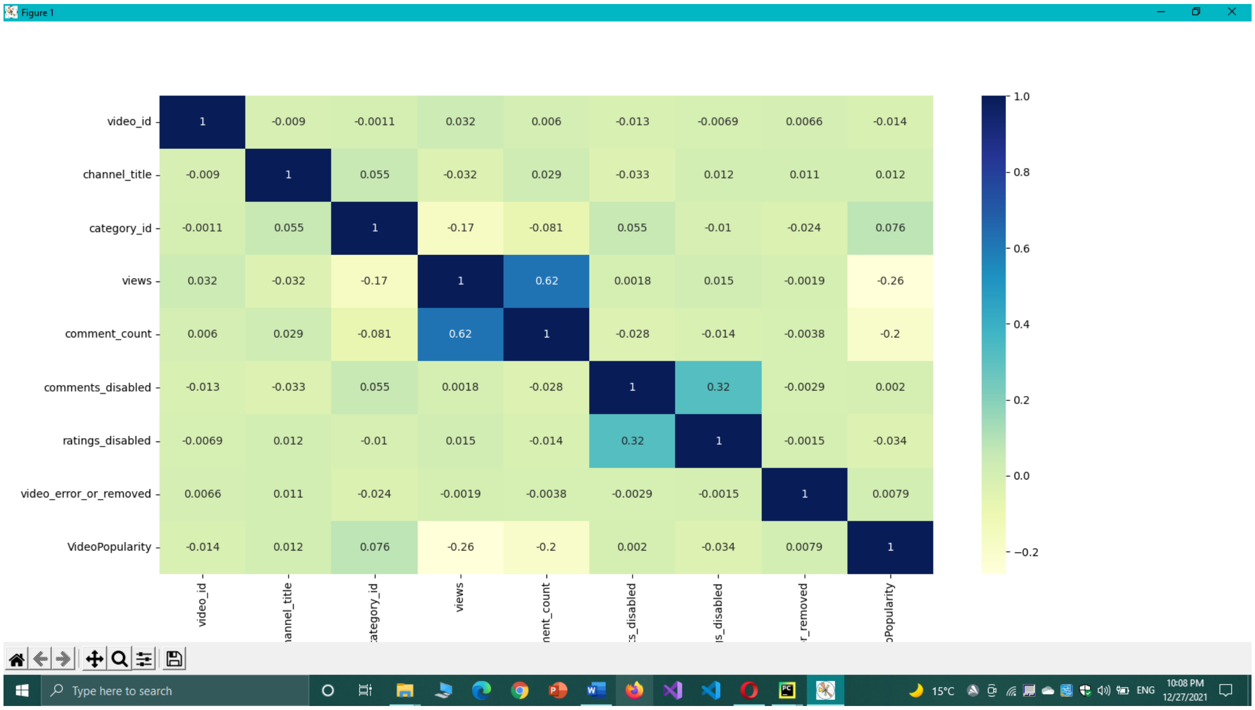
Conclusion:

After Showing correlation figure we Found that Likes most dependent on views and comments_count and the preprocessing on features improve accuracy of the models .

about classification , choosing good hyper parameter make good effect.



	video_id	annel_title	ategory_id	views	nent_count	:s_disabled	is_disabled	r_removed	o_be_trend	likes
video_id	1	-0.009	-0.0098	0.032	0.006	-0.013	-0.0069	0.0066	0.00017	0.016
channel_title	-0.009	1	0.046	-0.032	0.029	-0.033	0.012	0.011	0.028	0.0014
category_id	-0.0098	0.046	1	-0.17	-0.076	0.049	-0.012	-0.031	-0.028	-0.17
views	0.032	-0.032	-0.17	1	0.62	0.0018	0.015	-0.0019	-0.014	0.85
comment_count	0.006	0.029	-0.076	0.62	1	-0.028	-0.014	-0.0038	-0.013	0.8
comments_disabled	-0.013	-0.033	0.049	0.0018	-0.028	1	0.32	-0.0029	-0.0012	-0.029
ratings_disabled	-0.0069	0.012	-0.012	0.015	-0.014	0.32	1	-0.0015	0.0041	-0.021
video_error_or_removed	0.0066	0.011	-0.031	-0.0019	-0.0038	-0.0029	-0.0015	1	-0.00097	-0.0023
days_to_be_trend	0.00017	0.028	-0.028	-0.014	-0.013	-0.0012	0.0041	-0.00097	1	-0.017
likes	0.016	0.0014	-0.17	0.85	0.8	-0.029	-0.021	-0.0023	-0.017	1



	video_id	annel_title	ategory_id	views	nent_count	:s_disabled	is_disabled	r_removed	oPopularity
video_id	1	-0.009	-0.0011	0.032	0.006	-0.013	-0.0069	0.0066	-0.014
channel_title	-0.009	1	0.055	-0.032	0.029	-0.033	0.012	0.011	0.012
category_id	-0.0011	0.055	1	-0.17	-0.081	0.055	-0.01	-0.024	0.076
views	0.032	-0.032	-0.17	1	0.62	0.0018	0.015	-0.0019	-0.26
comment_count	0.006	0.029	-0.081	0.62	1	-0.028	-0.014	-0.0038	-0.2
comments_disabled	-0.013	-0.033	0.055	0.0018	-0.028	1	0.32	-0.0029	0.002
ratings_disabled	-0.0069	0.012	-0.01	0.015	-0.014	0.32	1	-0.0015	-0.034
video_error_or_removed	0.0066	0.011	-0.024	-0.0019	-0.0038	-0.0029	-0.0015	1	0.0079
VideoPopularity	-0.014	0.012	0.076	-0.26	-0.2	0.002	-0.034	0.0079	1