**Video Likes Prediction**

**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 .**

Linear regression output

**A screenshot of a computer

Description automatically generated**

Polynomial re**A screenshot of a computer

Description automatically generated**gression output

**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,publis  h\_time) from dataset |

**Null values:**

fill null values with values of previous index of row

**Analysis:**

Apply correlation to dataset

* **Likes depend on**

1. Views, comment\_count
2. Category\_id
3. video\_id
4. channel\_title
5. 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: Chart, histogram

Description automatically generated**

**2-testing time**

**Chart, histogram

Description automatically generated**

**3-accuracy summary: Chart, bar chart

Description automatically generated**

**hyperparameter tuning affected :**

1-**Decesion tree**

|  |  |
| --- | --- |
| **Hyper parameters** | **Accuracy** |
| Max\_depth=**12** | 95.39 |
| 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** | 95.32 |
| 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, multi\_class='auto',  n\_jobs=None, penalty='l2', random\_state=None, tol=0.0001, verbose=0, warm\_start=False | 78.23 |

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.

Timeline

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence