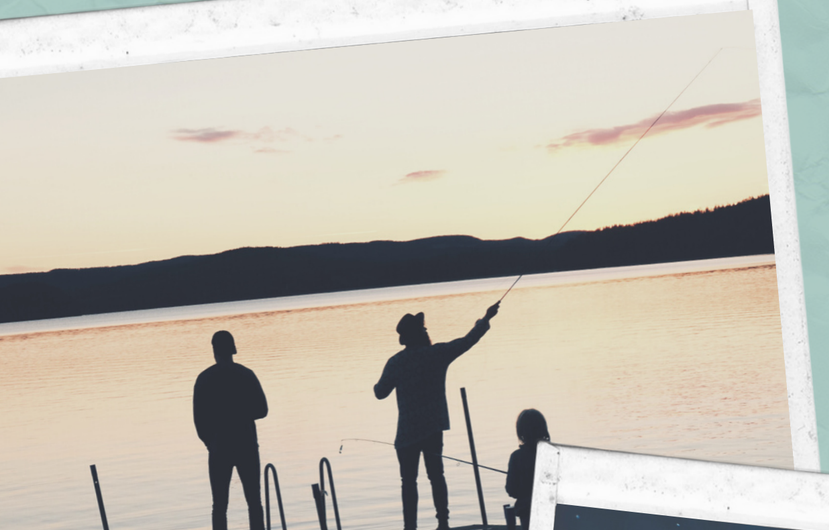
REGRESSION MACHINE LEARNING



PREDICTING FISH WEIGHT

By Dana Nicolas

(NUMERIC) DATA

WHAT'S IN MY DATASET

Vertical Length (cm)

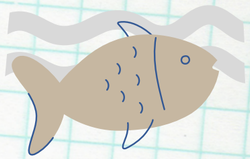
Diagonal Length (cm)

Cross Length (cm)

Width (cm)

(CATEGORICAL) DATA

WHAT'S IN MY DATASET



Species

Perch

Bream

Roach

Pike

Smelt

Parkki

Whitefish

DATA PREPARATION (EDA)

CONVERT CATEGORICAL TO BINARY

```
import category_encoders as ce
binary_encoder = ce.BinaryEncoder(cols=['Species'])
encoded_fishies = binary_encoder.fit_transform(fishies)
```

HEATMAP

ROBUST SCALER

SCALED:
LENGTHS, HEIGHT, WEIGHT

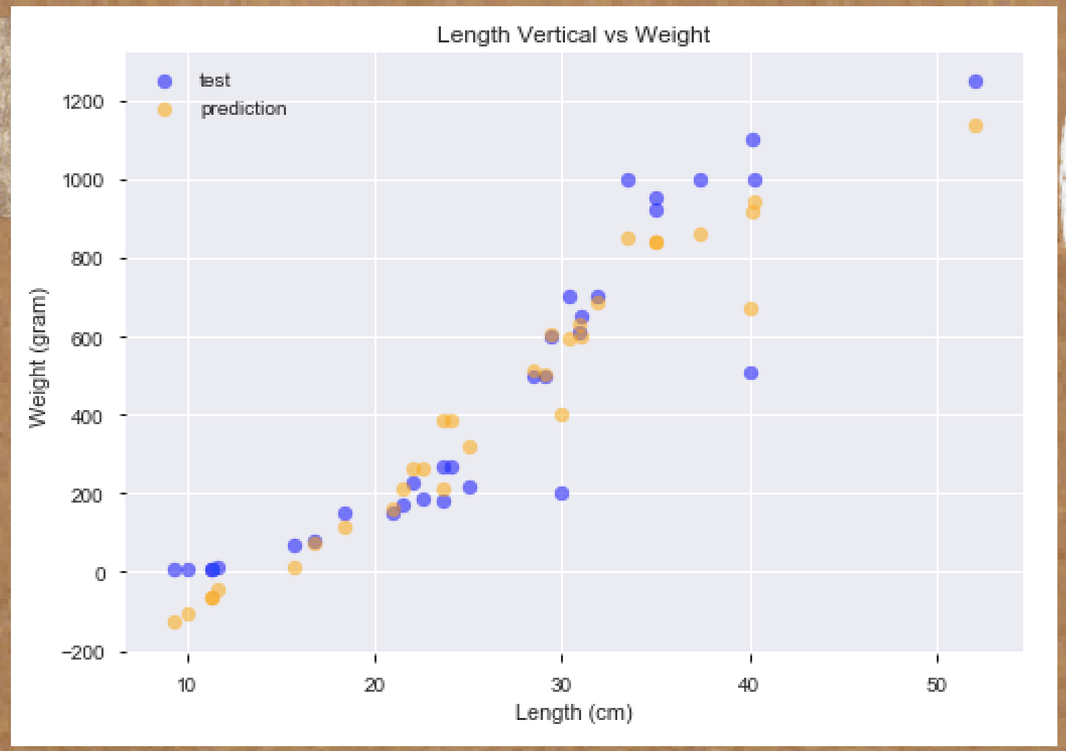
```
from sklearn.preprocessing import RobustScaler

f_columns = ['LengthVer', 'LengthDia', 'LengthCro', 'Height']
f_transformer = RobustScaler()
weight_transformer = RobustScaler()

f_transformer = f_transformer.fit(encoded_fishies[f_columns].to_numpy())
weight_transformer = weight_transformer.fit(encoded_fishies[['Weight']])

train.loc[:, f_columns] = f_transformer.transform(train[f_columns].to_numpy())
train['Weight'] = weight_transformer.transform(train[['Weight']])

test.loc[:, f_columns] = f_transformer.transform(test[f_columns].to_numpy())
test['Weight'] = weight_transformer.transform(test[['Weight']])
```

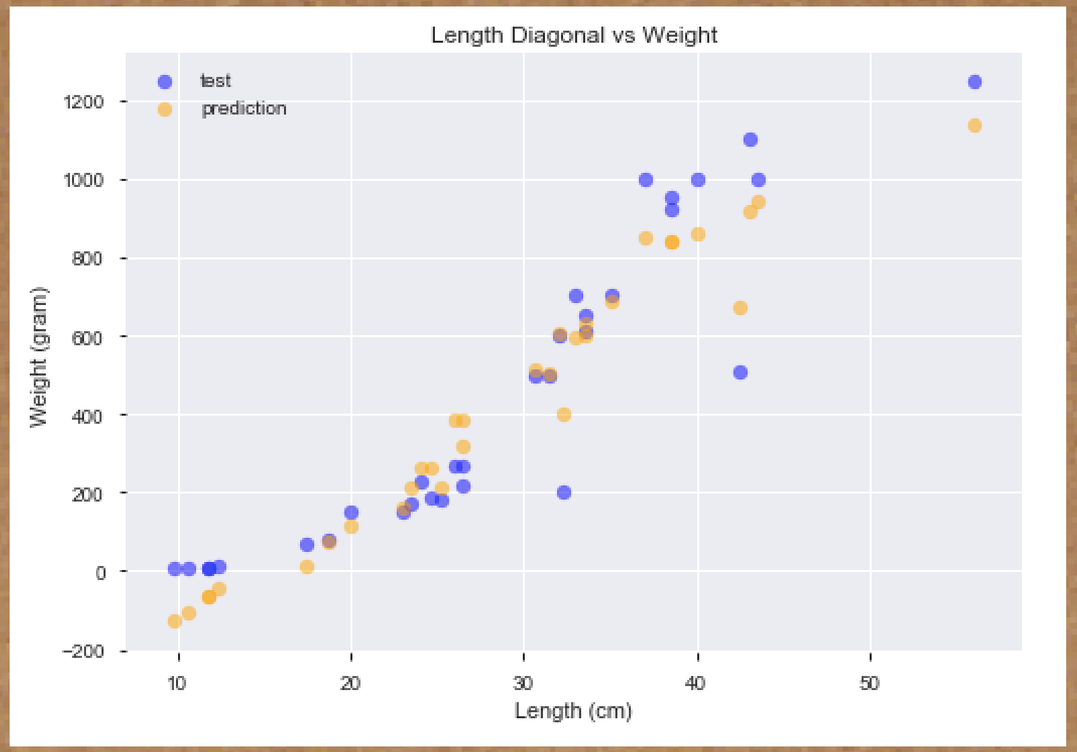
LINEAR REGRESSION

```
lr_model = LinearRegression()
lr_model.fit(x_train, y_train)
y_pred = lr_model.predict(x_test)
y_pred
```

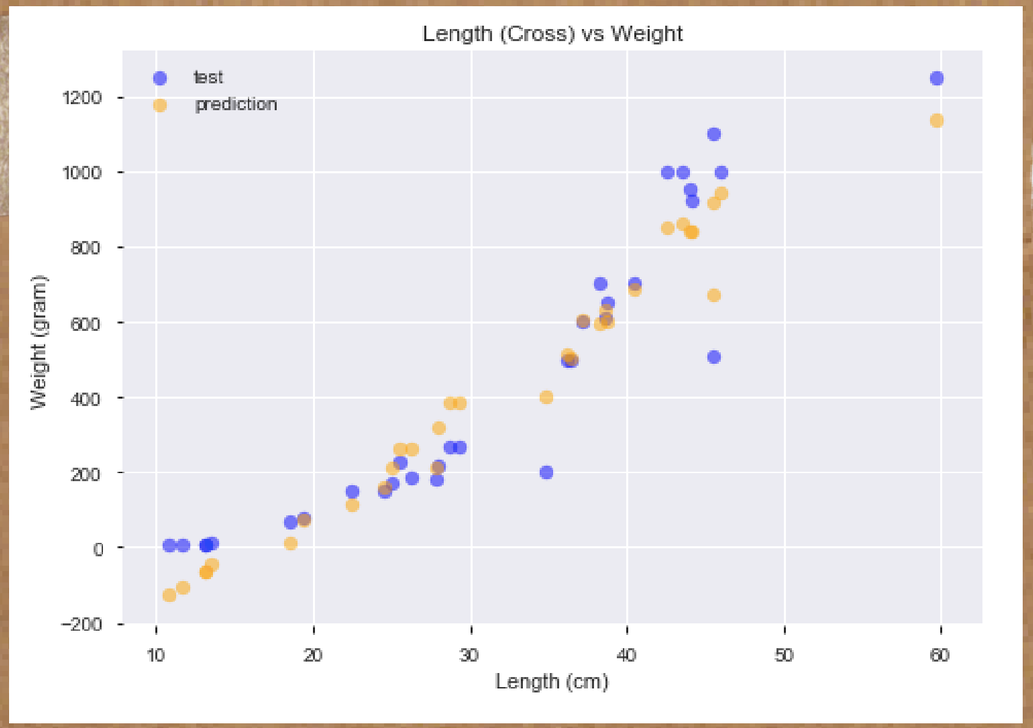
R2: 0.9373
MSE: 0.032

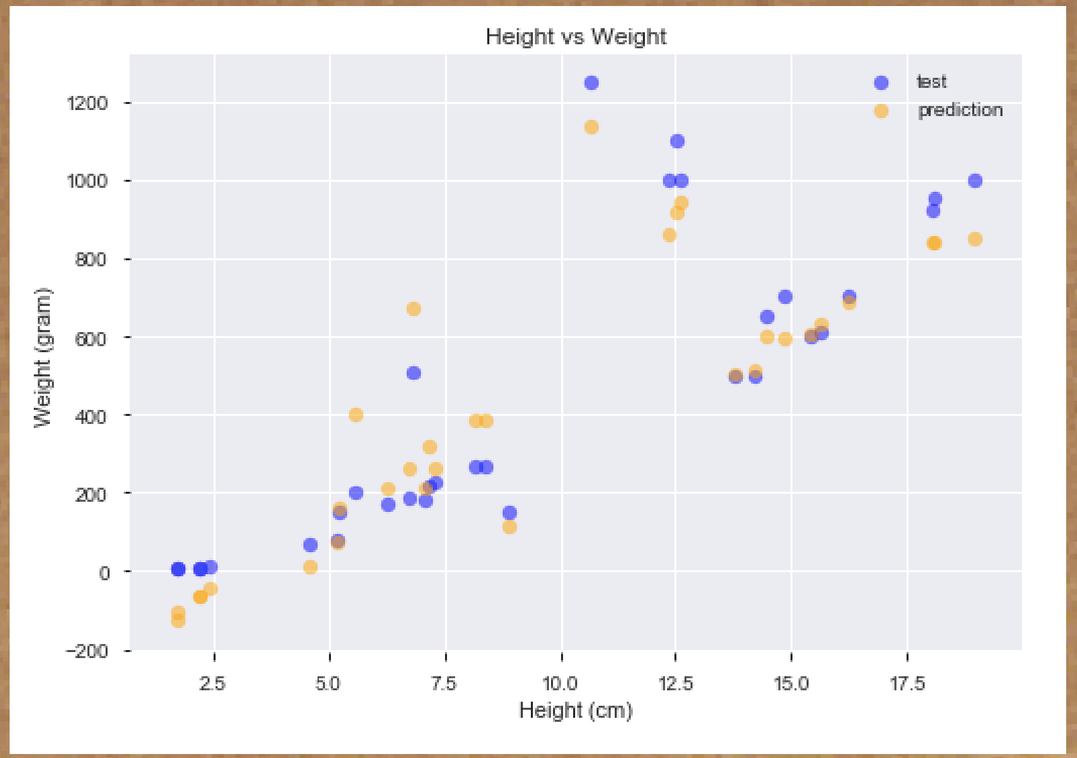
Vertical and Diagonal Length



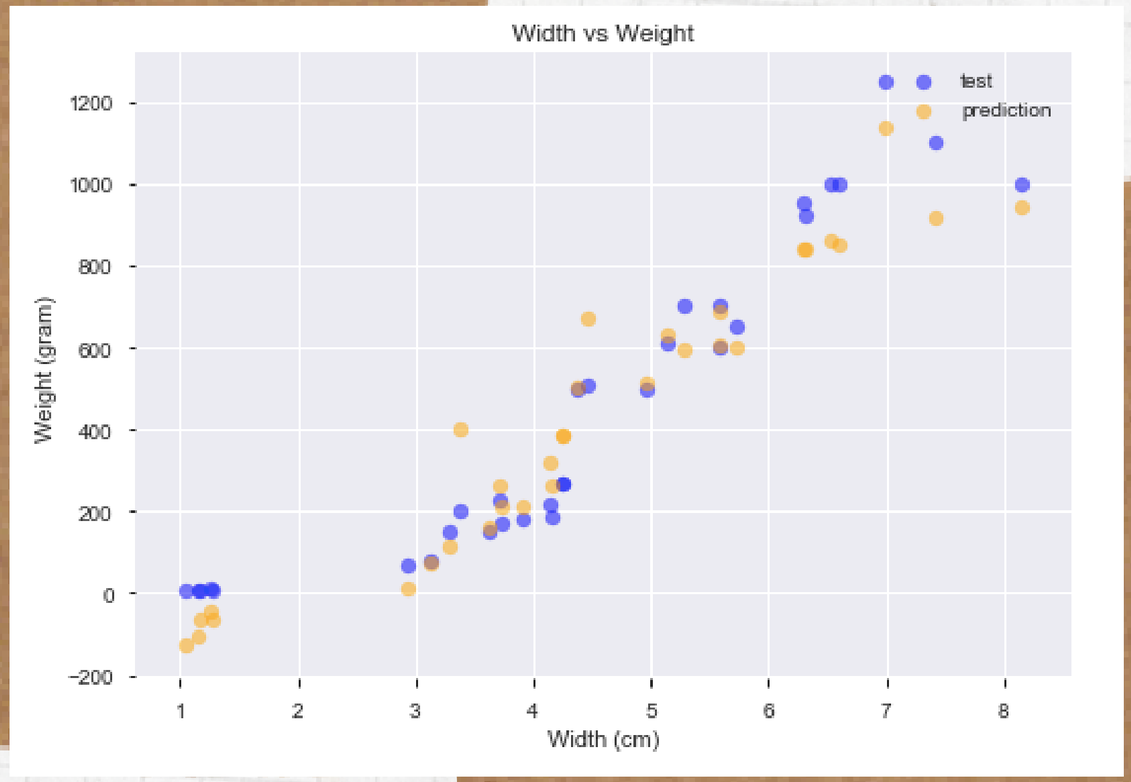


Cross Length and Height





Width


Width vs Weight
test
prediction
Weight (gram)
Width (cm)
1200
1000
800
600
400
200
0
−200
1
2
3
4
5
6
7
8

SUPPORT VECTOR REGRESSION

```python
from sklearn.svm import SVR

svr = SVR(kernel='linear')
svr.fit(x_train, y_train)
y_pred = svr.predict(x_test)
```

R2: 0.9349
MSE: 0.0330

R2 SCORES

C = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100]

```
[-0.1062827014720864,
 0.6161316444213405,
 0.8566696778533788,
 0.9116113655775042,
 0.934922204634037 3,
 0.9364409153802398,
 0.9361393307666539]
```

TOL = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100]

```
[0.934796392479635 2,
 0.934922204634037 3,
 0.935775631272202 1,
 0.930694862180255 9,
 0.856895567991252 2,
 -1.020038658211304 1,
 -1.020038658211304 1]
```

EPS = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100]

```
[0.925486865246082 7,
 0.925843535874266 5,
 0.928537483824037 8,
 0.934922204634037 3,
 0.348649922258149 4,
 -1.020038658211303 7,
 -1.020038658211316 1]
```

MEAN SQUARE ERROR

C = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100]

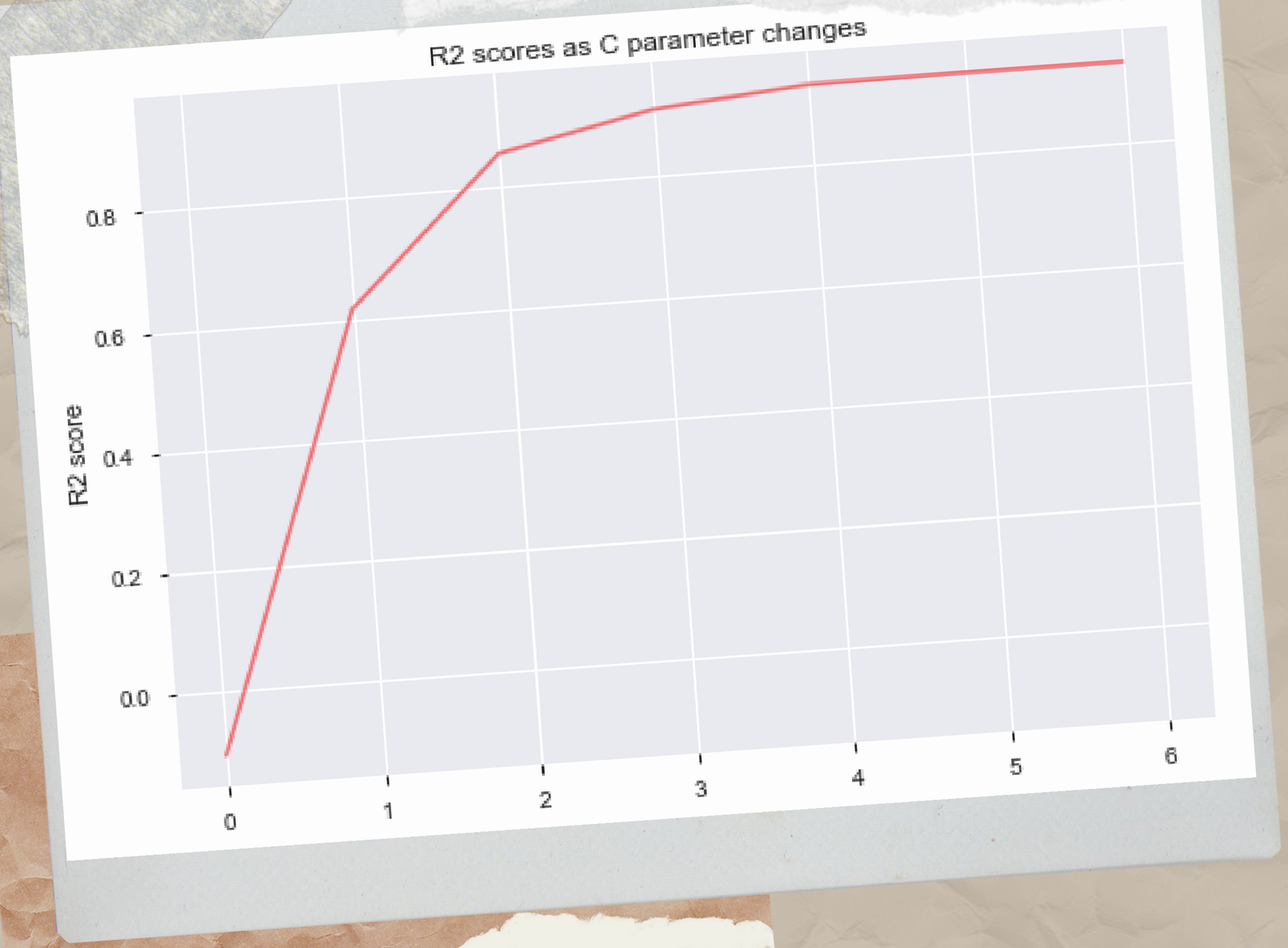
```
[0.5601881897341542,
 0.1943793561281331 5,
 0.072578151669991 4,
 0.04475733828643 0026,
 0.0329534325443 5603,
 0.0321844032333 9159,
 0.032337116584 6096]
```

TOL = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100]

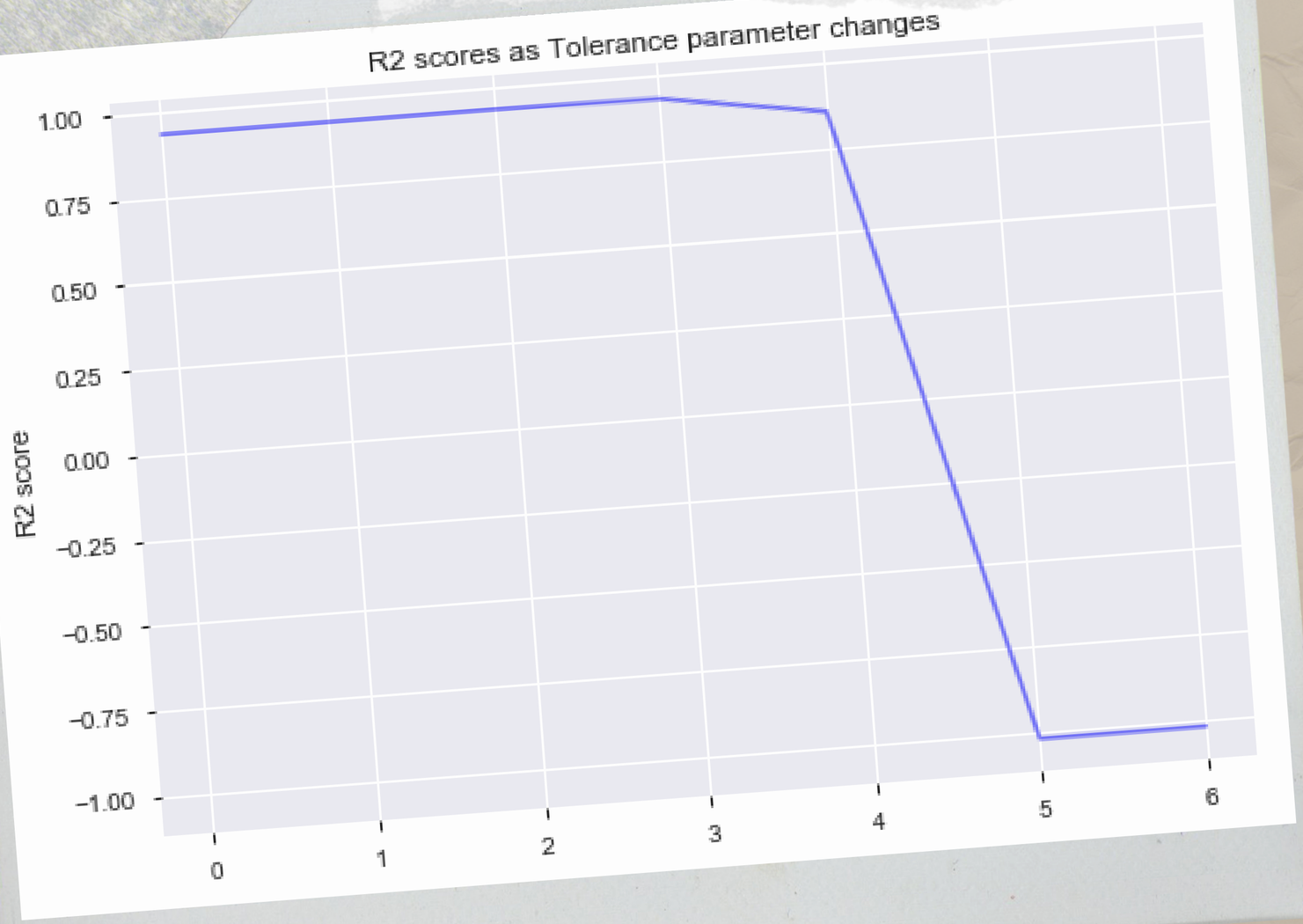
```
[0.0330171400243 0427,
 0.0329534325443 5603,
 0.0325212830378 4359,
 0.0350940312479 4151,
 0.072463767718,
 1.0228866433784265,
 1.0228866433784265]
```

EPS = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100]

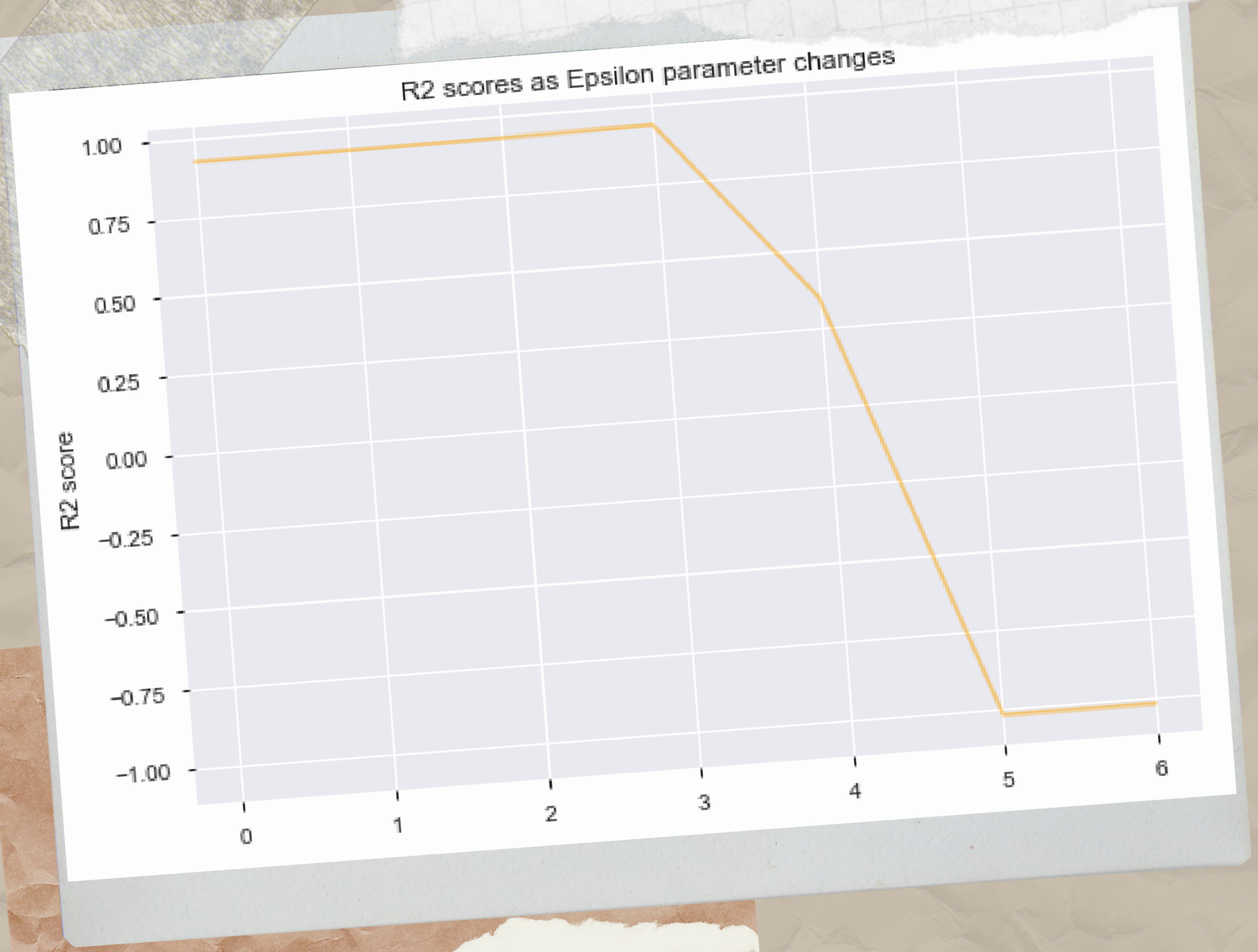
```
[0.0377312037995 9882,
 0.0375505965522 21864,
 0.0361864625716 2872,
 0.0329534325443 5603,
 0.3298240318210512,
 1.0228866433784263,
 1.0228866433784325]
```



Plotting R2 Scores with C changes



Plotting R2 Scores with Tolerance changes



Plotting R2 Scores with Epsilon changes

WHAT I'VE LEARNED

R2 SCORE AND MSE

R2 is the accuracy in regression. MSE is how far your prediction is with actual values.

C, TOLERANCE, EPSILON

Own ways of affecting R2 and MSE

SVR IS MORE FLEXIBLE

Has more hyper parameters that you can change compared to LinearRegression

ROBUST SCALER

When dealing with huge values, it's easier to interpret MSE when you've scaled your features

THANKS

QUESTIONS?