First, we import the proper modules and our data.

In [10]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

abalone = pd.read_csv("abalone.csv")
#abalone = abalone.to_numpy()

abalone.head()
#abalone[:5,:]
```

Out[10]:

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	I
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	

Here, we separate the the data's labels from the other factors.

In [2]:
```
abalone_age = abalone[:,-1]
abalone_vars = abalone[:, :-1]

#abalone_age.head()
abalone_vars[:5,:]
abalone_age[:5]
```

Out[2]:
```
array([15, 7, 9, 10, 7], dtype=object)
```

In [3]:
```
#  Sklearn modules
from sklearn.linear_model import LinearRegression, Lasso, ElasticNet, Ridge
from sklearn.metrics import mean_squared_error
```

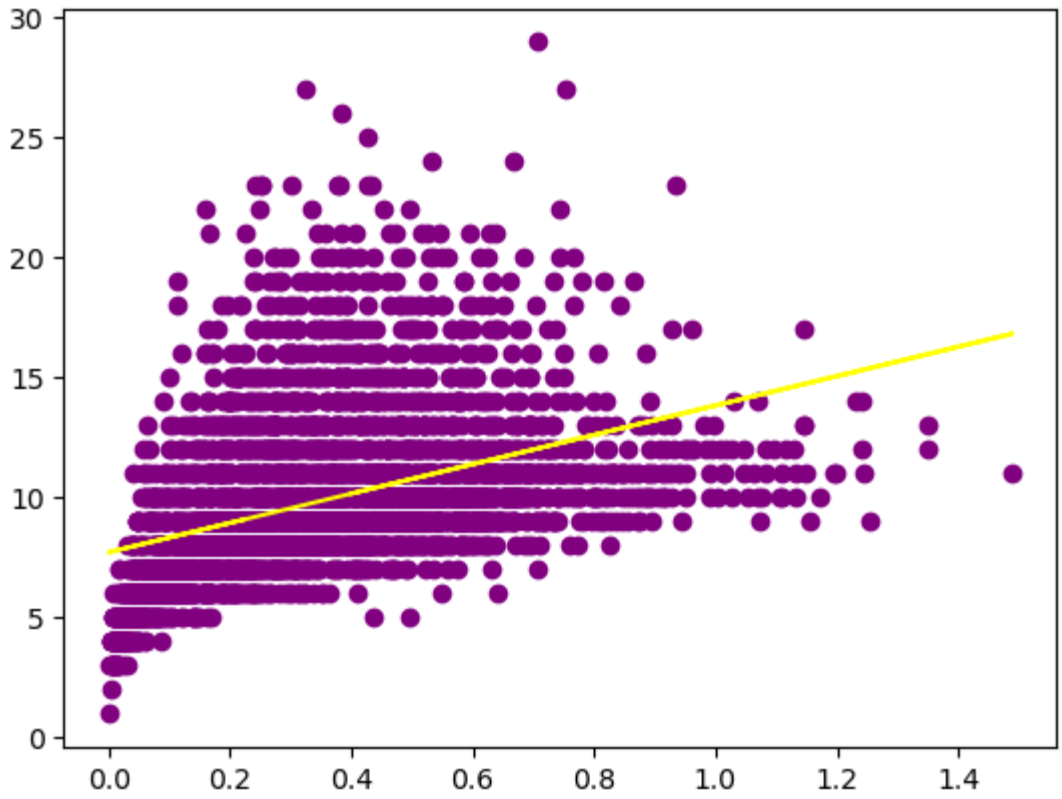
In [4]:
```
#  Single Regression

lin_reg = LinearRegression()
#abalone_age.reshape(-1,1)
lin_reg.fit(abalone_vars[:,5].reshape(-1,1), abalone_age)
lin_reg.intercept_, lin_reg.coef_
```

Out[4]:
```
(7.736388472350656, array([6.11631977]))
```

In [7]:
```
b, m = lin_reg.intercept_, lin_reg.coef_.item()
print(b, m)
plt.scatter(abalone_vars[:,5], abalone_age, color="purple")
plt.plot(abalone_vars[:,5], m*abalone_vars[:,5]+b, color="yellow")
```

Out[7]:
```
7.736388472350656 6.116319771361458
[<matplotlib.lines.Line2D at 0x17ec7be28e0>]
```



In [38]:
```
#   Algorithm work
predictors = [LinearRegression(),                  # Option 0
              Lasso(alpha=0.1),                    # Option 1
              ElasticNet(alpha=0.1, l1_ratio=0.5), # Option 2
              Ridge(alpha=1, solver="cholesky")]   # Option 3

option = 3

Y = abalone_age
for i in range(0,4):
    X = abalone_vars[:,1:4]#.reshape(-1,1)
    weak_lin_reg = predictors[i]
    weak_lin_reg.fit(X, Y)
    print(weak_lin_reg.intercept_, weak_lin_reg.coef_)
```

```
2.833186495591878 [-11.93600773  25.77527434  20.36768217]
5.736804012490044 [8.01010376 0.         0.        ]
8.0109564412069 [2.22999818 1.78621685 0.18761012]
2.536080465032086 [-0.69599474 13.82891264 15.20822347]
```

In [19]:
```
#   Algorithm work

#  Data split from Geron's "Hands-on Machinle Learning with Scikit-Learn,
#  Keras & Tensorflow
def split_train_test(data, test_ratio):
    shuffled_indices = np.random.permutation(len(data))
    test_set_size = int(len(data) * test_ratio)
    test_indices = shuffled_indices[:test_set_size]
    train_indices = shuffled_indices[test_set_size:]
    return data.iloc[train_indices], data.iloc[test_indices]

train_set, test_set = split_train_test(abalone, 0.2)

train_set.shape, test_set.shape
```

Out[19]:
```
((3340, 9), (834, 9))
```

In [ ]:
```

```