In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
train=pd.read_csv('/Users/acer/Sandesh Pal/Data Science Assgn/Naive Bayes/SalaryData_Train.csv')
train
```

Out[2]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capitalgain	capitalloss	hoursperweek	nati
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	Unite Sta
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	Unite Sta
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	Unite Sta
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	Unite Sta
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cu
...	...	...	...	...	...	...	...	...	...	...	...	...	
30156	27	Private	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	Unite Sta
30157	40	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	Unite Sta
30158	58	Private	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	Unite Sta
30159	22	Private	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	Unite Sta
30160	52	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	Unite Sta

30161 rows × 14 columns

In [3]:

```
test=pd.read_csv('/Users/acer/Sandesh Pal/Data Science Assgn/Naive Bayes/SalaryData_Test.csv')
test
```

Out[3]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capitalgain	capitalloss	hoursperweek	na
0	25	Private	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	Un S
1	38	Private	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	Un S
2	28	Local-gov	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	Un S
3	44	Private	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	Un S
4	34	Private	10th	6	Never-married	Other-service	Not-in-family	White	Male	0	0	30	Un S
...	...	...	...	...	...	...	...	...	...	...	...	...	
15055	33	Private	Bachelors	13	Never-married	Prof-specialty	Own-child	White	Male	0	0	40	Un S
15056	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	36	Un S
15057	38	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	Un S
15058	44	Private	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0	40	Un S
15059	35	Self-emp-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	60	Un S

15060 rows × 14 columns

In [4]:

```
train.isnull().value_counts()
```

Out[4]:

```
age    workclass  education  educationno  maritalstatus  occupation  relationship  race   sex
capitalgain  capitalloss  hoursperweek  native  Salary
False  False      False      False        False          False       False         False  False  False
False        False        False         False     30161
dtype: int64
```

In [6]:

```
train.isnull().sum()
```

Out[6]:

```
age              0
workclass        0
education        0
educationno      0
maritalstatus    0
occupation       0
relationship     0
race             0
sex              0
capitalgain      0
capitalloss      0
hoursperweek     0
native           0
Salary           0
dtype: int64
```

In [7]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30161 entries, 0 to 30160
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   age            30161 non-null  int64 
 1   workclass      30161 non-null  object
 2   education      30161 non-null  object
 3   educationno    30161 non-null  int64 
 4   maritalstatus  30161 non-null  object
 5   occupation     30161 non-null  object
 6   relationship   30161 non-null  object
 7   race           30161 non-null  object
 8   sex            30161 non-null  object
 9   capitalgain    30161 non-null  int64 
 10  capitalloss    30161 non-null  int64 
 11  hoursperweek   30161 non-null  int64 
 12  native         30161 non-null  object
 13  Salary         30161 non-null  object
dtypes: int64(5), object(9)
memory usage: 3.2+ MB
```

In [8]:

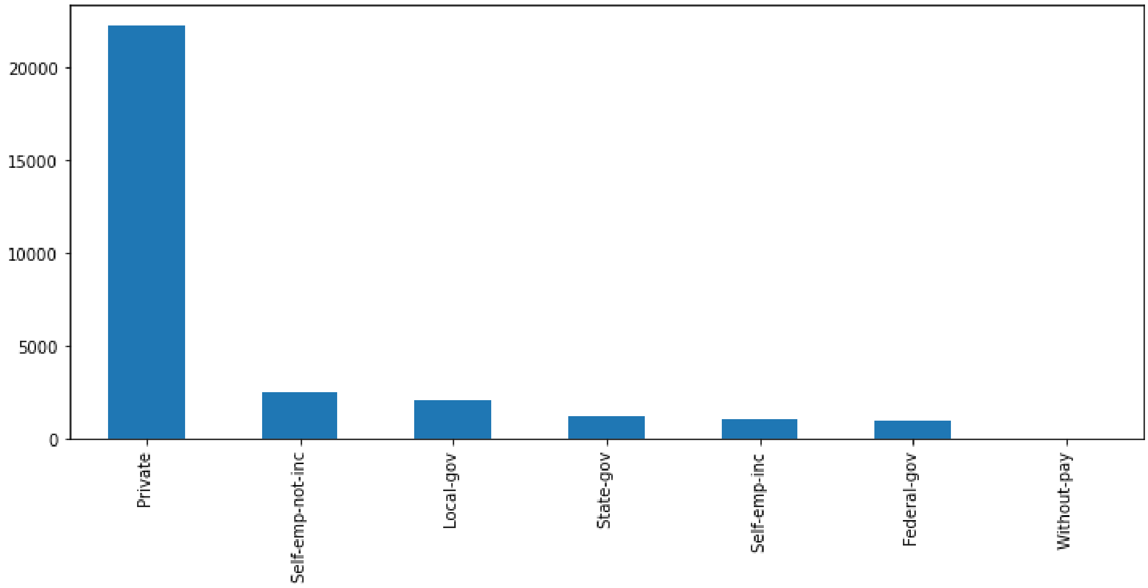
```
train.describe
```

Out[8]:

```
<bound method NDFrame.describe of        age          workclass    education  educationno        maritals
atus  \
0       39          State-gov    Bachelors           13        Never-married
1       50   Self-emp-not-inc    Bachelors           13   Married-civ-spouse
2       38            Private      HS-grad            9             Divorced
3       53            Private         11th            7   Married-civ-spouse
4       28            Private    Bachelors           13   Married-civ-spouse
...    ...                ...          ...          ...                  ...
30156   27            Private   Assoc-acdm           12   Married-civ-spouse
30157   40            Private      HS-grad            9   Married-civ-spouse
30158   58            Private      HS-grad            9              Widowed
30159   22            Private      HS-grad            9        Never-married
30160   52       Self-emp-inc      HS-grad            9   Married-civ-spouse

              occupation    relationship    race      sex  capitalgain  \
0           Adm-clerical   Not-in-family   White     Male         2174
1        Exec-managerial         Husband   White     Male            0
2      Handlers-cleaners   Not-in-family   White     Male            0
3      Handlers-cleaners         Husband   Black     Male            0
4         Prof-specialty            Wife   Black   Female            0
...                  ...             ...     ...      ...          ...
30156       Tech-support            Wife   White   Female            0
30157  Machine-op-inspct         Husband   White     Male            0
30158       Adm-clerical       Unmarried   White   Female            0
30159       Adm-clerical       Own-child   White     Male            0
30160    Exec-managerial            Wife   White   Female        15024

       capitalloss  hoursperweek          native  Salary
0                0            40   United-States   <=50K
1                0            13   United-States   <=50K
2                0            40   United-States   <=50K
3                0            40   United-States   <=50K
4                0            40            Cuba   <=50K
...            ...           ...             ...     ...
30156            0            38   United-States   <=50K
30157            0            40   United-States    >50K
30158            0            40   United-States   <=50K
30159            0            20   United-States   <=50K
30160            0            40   United-States    >50K

[30161 rows x 14 columns]>
```

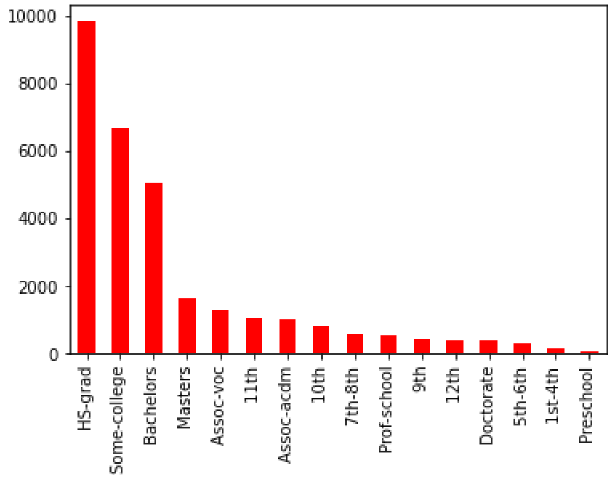
In [9]:

```
plt.figure(figsize=(12,5))
train.workclass.value_counts().plot.bar();
```



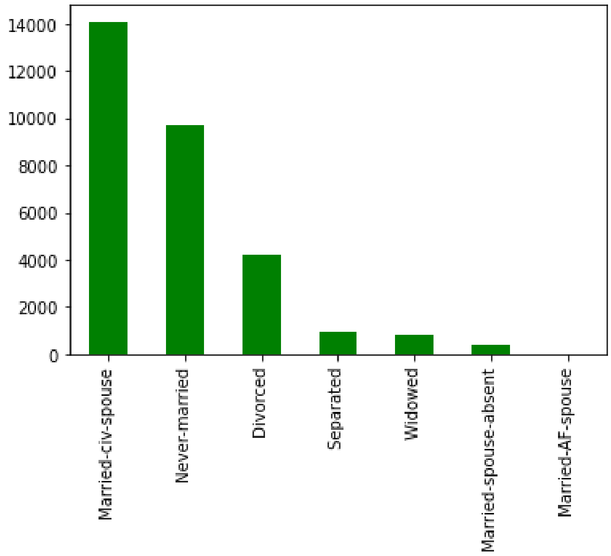
In [10]:

```
train.education.value_counts().plot.bar(color='red');
```



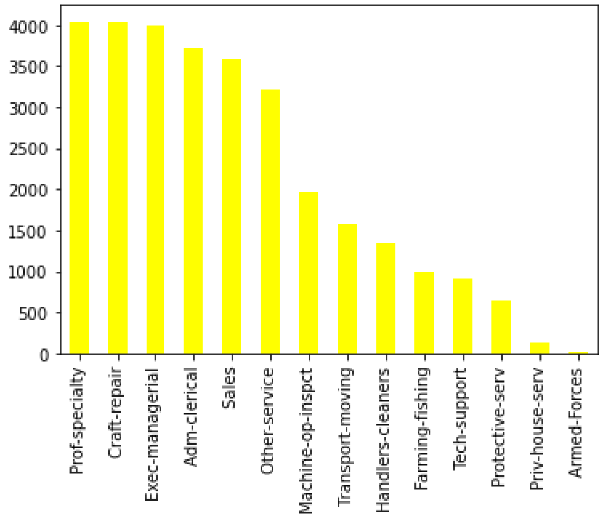
In [11]:

```
train.maritalstatus.value_counts().plot.bar(color='green');
```



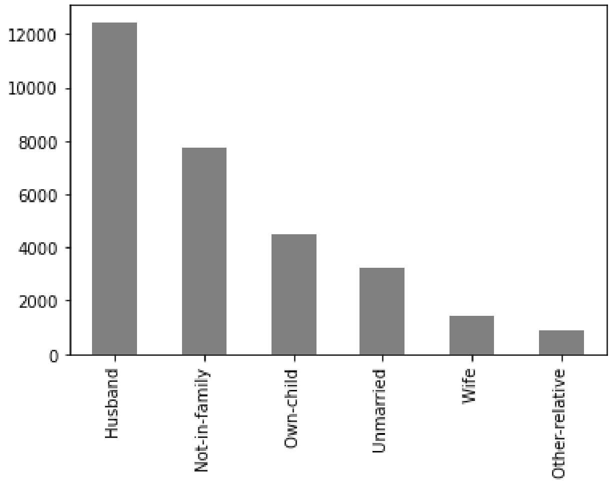
In [12]:

```
train.occupation.value_counts().plot.bar(color='yellow');
```



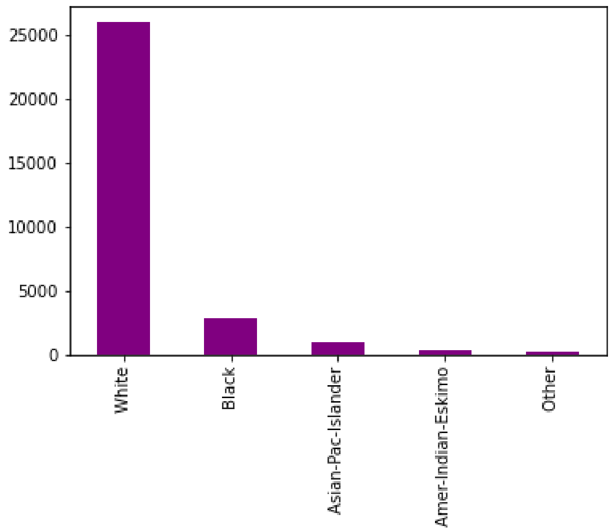
In [13]:

```
train.relationship.value_counts().plot.bar(color='grey');
```



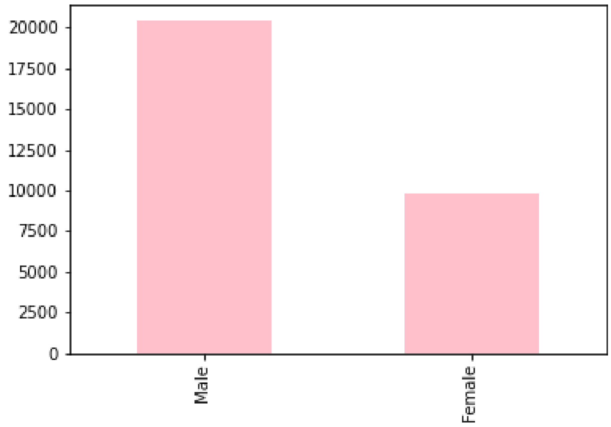
In [14]:

```
train.race.value_counts().plot.bar(color='purple');
```



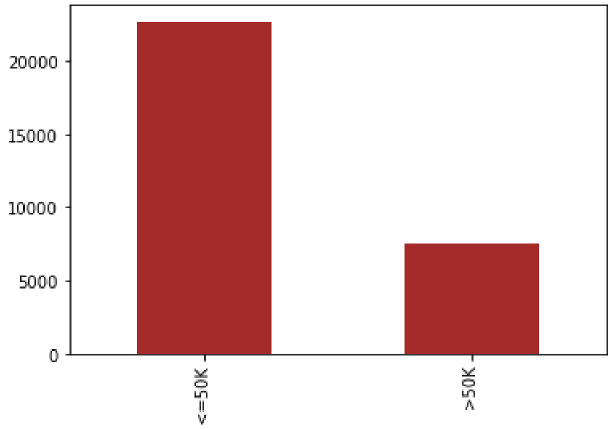
In [15]:

```
train.sex.value_counts().plot.bar(color='pink');
```



In [16]:

```
train.Salary.value_counts().plot.bar(color='brown');
```



In [17]:

```
train1 = train.iloc[:,0:13]
train1 = pd.get_dummies(train1)
train1
```

Out[17]:

	age	educationno	capitalgain	capitalloss	hoursperweek	workclass_Federal-gov	workclass_Local-gov	workclass_Private	workclass_Self-emp-inc	workclass_Self-emp-not-inc	...	native_Portugal	native_Puerto-Ric
0	39	13	2174	0	40	0	0	0	0	0	...	0	
1	50	13	0	0	13	0	0	0	0	1	...	0	
2	38	9	0	0	40	0	0	1	0	0	...	0	
3	53	7	0	0	40	0	0	1	0	0	...	0	
4	28	13	0	0	40	0	0	1	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
30156	27	12	0	0	38	0	0	1	0	0	...	0	
30157	40	9	0	0	40	0	0	1	0	0	...	0	
30158	58	9	0	0	40	0	0	1	0	0	...	0	
30159	22	9	0	0	20	0	0	1	0	0	...	0	
30160	52	9	15024	0	40	0	0	0	1	0	...	0	

30161 rows × 102 columns

In [18]:

```
finaltrain = pd.concat([train1, train['Salary']],axis=1)
finaltrain
```

Out[18]:

	age	educationno	capitalgain	capitalloss	hoursperweek	workclass_ Federal-gov	workclass_ Local-gov	workclass_ Private	workclass_ Self-emp-inc	workclass_ Self-emp-not-inc	...	native_ Puerto-Rico	native Scotlar
0	39	13	2174	0	40	0	0	0	0	0	...	0	
1	50	13	0	0	13	0	0	0	0	1	...	0	
2	38	9	0	0	40	0	0	1	0	0	...	0	
3	53	7	0	0	40	0	0	1	0	0	...	0	
4	28	13	0	0	40	0	0	1	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
30156	27	12	0	0	38	0	0	1	0	0	...	0	
30157	40	9	0	0	40	0	0	1	0	0	...	0	
30158	58	9	0	0	40	0	0	1	0	0	...	0	
30159	22	9	0	0	20	0	0	1	0	0	...	0	
30160	52	9	15024	0	40	0	0	0	1	0	...	0	

30161 rows × 103 columns

In [19]:

```
test1 = test.iloc[:,0:13]
test1 = pd.get_dummies(test1)
test1
```

Out[19]:

	age	educationno	capitalgain	capitalloss	hoursperweek	workclass_ Federal-gov	workclass_ Local-gov	workclass_ Private	workclass_ Self-emp-inc	workclass_ Self-emp-not-inc	...	native_ Portugal	native Puertc Ric
0	25	7	0	0	40	0	0	1	0	0	...	0	
1	38	9	0	0	50	0	0	1	0	0	...	0	
2	28	12	0	0	40	0	1	0	0	0	...	0	
3	44	10	7688	0	40	0	0	1	0	0	...	0	
4	34	6	0	0	30	0	0	1	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
15055	33	13	0	0	40	0	0	1	0	0	...	0	
15056	39	13	0	0	36	0	0	1	0	0	...	0	
15057	38	13	0	0	50	0	0	1	0	0	...	0	
15058	44	13	5455	0	40	0	0	1	0	0	...	0	
15059	35	13	0	0	60	0	0	0	1	0	...	0	

15060 rows × 102 columns

In [20]:

```
finaltest = pd.concat([test1, test['Salary']],axis=1)
finaltest
```

Out[20]:

	age	educationno	capitalgain	capitalloss	hoursperweek	workclass_Federal-gov	workclass_Local-gov	workclass_Private	workclass_Self-emp-inc	workclass_Self-emp-not-inc	...	native_Puerto-Rico	native Scotlar
0	25	7	0	0	40	0	0	1	0	0	...	0	
1	38	9	0	0	50	0	0	1	0	0	...	0	
2	28	12	0	0	40	0	1	0	0	0	...	0	
3	44	10	7688	0	40	0	0	1	0	0	...	0	
4	34	6	0	0	30	0	0	1	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
15055	33	13	0	0	40	0	0	1	0	0	...	0	
15056	39	13	0	0	36	0	0	1	0	0	...	0	
15057	38	13	0	0	50	0	0	1	0	0	...	0	
15058	44	13	5455	0	40	0	0	1	0	0	...	0	
15059	35	13	0	0	60	0	0	0	1	0	...	0	

15060 rows × 103 columns

In [21]:

```
X = finaltrain.values[:,0:102]
Y = finaltrain.values[:,102]
x = finaltest.values[:,0:102]
y = finaltest.values[:,102]
```

In [22]:

```
# Preparing a naive bayes model
from sklearn.naive_bayes import MultinomialNB as MB
from sklearn.naive_bayes import GaussianNB as GB
# Multinomial Naive Bayes
classifier_mb = MB()
classifier_mb.fit(X,Y)
train_pred_m = classifier_mb.predict(X)
accuracy_train_m = np.mean(train_pred_m==Y)
test_pred_m = classifier_mb.predict(x)
accuracy_test_m = np.mean(test_pred_m==y)
print('Training accuracy is:',accuracy_train_m,'\n','Testing accuracy is:',accuracy_test_m)
```

```
Training accuracy is: 0.7729186698053778
 Testing accuracy is: 0.7749667994687915
```

In [23]:

```
# Gaussian Naive Bayes
classifier_gb = GB()
classifier_gb.fit(X,Y) # we need to convert tfidf into array format which is compatible for gaussian naiv
train_pred_g = classifier_gb.predict(X)
accuracy_train_g = np.mean(train_pred_g==Y)
test_pred_g = classifier_gb.predict(x)
accuracy_test_g = np.mean(test_pred_g==y)
print('Training accuracy is:',accuracy_train_g,'\n','Testing accuracy is:',accuracy_test_g)
```

```
Training accuracy is: 0.8031563940187659
 Testing accuracy is: 0.8029216467463479
```

In [ ]: