In [9]:
```
import numpy as np
import pandas as pd
from typing import Dict, Tuple
from scipy import stats
from sklearn.naive_bayes import GaussianNB, MultinomialNB, ComplementNB, BernoulliNB
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score, classification_report
from sklearn.pipeline import Pipeline
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")

import warnings
```

In [10]:
```
warnings.filterwarnings('ignore') # Отключаем предупреждения
```

In [3]:
```
data = pd.read_csv('adult.data.csv')
data.head()
```

Out[3]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

In [42]:
```
x_train, x_test, y_train, y_test = train_test_split(data['race'], data['sex'], test_size=0.5, random_state=1)
```

In [43]:
```
def accuracy_score_for_classes(
    y_true: np.ndarray,
    y_pred: np.ndarray) -> Dict[int, float]:
    """
    Вычисление метрики accuracy для каждого класса
    y_true - истинные значения классов
    y_pred - предсказанные значения классов
    Возвращает словарь: ключ - метка класса,
    значение - Accuracy для данного класса
    """
    # Для удобства фильтрации сформируем Pandas DataFrame
    d = {'t': y_true, 'p': y_pred}
    df = pd.DataFrame(data=d)
    # Метки классов
    classes = np.unique(y_true)
    # Результирующий словарь
    res = dict()
    # Перебор меток классов
    for c in classes:
        # отфильтруем данные, которые соответствуют
        # текущей метке класса в истинных значениях
        temp_data_flt = df[df['t']==c]
        # расчет accuracy для заданной метки класса
        temp_acc = accuracy_score(
            temp_data_flt['t'].values,
            temp_data_flt['p'].values)
        # сохранение результата в словарь
        res[c] = temp_acc
    return res

def print_accuracy_score_for_classes(
    y_true: np.ndarray,
    y_pred: np.ndarray):
    """
    Вывод метрики accuracy для каждого класса
    """
    accs = accuracy_score_for_classes(y_true, y_pred)
    if len(accs)>0:
        print('Метка \t Accuracy')
    for i in accs:
        print('{} \t {}'.format(i, accs[i]))
```

In [44]:
```
def sentiment(v, c):
    model = Pipeline(
        [("vectorizer", v),
         ("classifier", c)])
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print_accuracy_score_for_classes(y_test, y_pred)
```

Классификация с использованием логистической регресии

In [45]:
```
sentiment(TfidfVectorizer(), LogisticRegression(C=5.0, solver='lbfgs'))
```
```
Метка    Accuracy
Female   0.14269870609981516
Male     0.9268696532057769
```

In [46]:
```
sentiment(CountVectorizer(), MultinomialNB())
```
```
Метка    Accuracy
Female   0.14269870609981516
Male     0.9268696532057769
```

In [47]:
```
sentiment(TfidfVectorizer(), MultinomialNB())
```
```
Метка    Accuracy
Female   0.14269870609981516
Male     0.9268696532057769
```

In [48]:
```
sentiment(CountVectorizer(), ComplementNB())
```
```
Метка    Accuracy
Female   0.16506469500924215
Male     0.9105878024100819
```

In [49]:
```
sentiment(TfidfVectorizer(), ComplementNB())
```
```
Метка    Accuracy
Female   0.16506469500924215
Male     0.9105878024100819
```

In [50]:
```
sentiment(CountVectorizer(binary=True), BernoulliNB())
```
```
Метка    Accuracy
Female   0.15397412199630314
Male     0.9184067703063196
```