Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

“Методы машинного обучения”

РУБЕЖНЫЙ КОНТРОЛЬ № 2. **«Методы построения моделей машинного обучения»**

Студент группы ИУ5-24М

Петропавлов Д.М.

_____ Дата

_____ Подпись

Москва 2020

Задача

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета (кроме примера, который рассматривался в лекции). Классификация может быть бинарной или многоклассовой. Целевой признак из выбранного Вами датасета может иметь любой физический смысл, примером является задача анализа тональности текста.

Необходимо сформировать признаки на основе CountVectorizer или TfidfVectorizer.

В качестве классификаторов необходимо использовать два классификатора, не относящихся к наивным Байесовским методам (например, LogisticRegression, LinearSVC), а также Multinomial Naive Bayes (MNB), Complement Naive Bayes (CNB), Bernoulli Naive Bayes.

Для каждого метода необходимо оценить качество классификации с помощью хотя бы двух метрик качества классификации (например, Accuracy, ROC-AUC).

Сделате выводы о том, какой классификатор осуществляет более качественную классификацию на Вашем наборе данных.

In [3]:
```
import numpy as np
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix, plot_confusion_matrix
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB, ComplementNB, BernoulliNB
import matplotlib.pyplot as plt
import pandas as pd
```

Подключаем данные

In [4]:
```
data = pd.read_csv('C:/Users/wonde/virtualenvs/tensorflow/Scripts/MyFolderForMMOLabs/googleplaystore.csv', sep=",")
```

In [5]:
```
data.head()
```

Out[5]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

In [9]:
```
data.shape
```

Out[9]:
```
(10841, 13)
```

Используем TfidVectorizer

In [10]:
```
newsgroups_train = fetch_20newsgroups(subset='train', remove=('headers', 'footers'))
newsgroups_test = fetch_20newsgroups(subset='test', remove=('headers', 'footers'))
```

In [11]:
```
vectorizer = TfidfVectorizer()
vectorizer.fit(newsgroups_train.data + newsgroups_test.data)
```

Out[11]:
```
TfidfVectorizer()
```

In [12]:
```
X_train = vectorizer.transform(newsgroups_train.data)
X_test = vectorizer.transform(newsgroups_test.data)

y_train = newsgroups_train.target
y_test = newsgroups_test.target
```

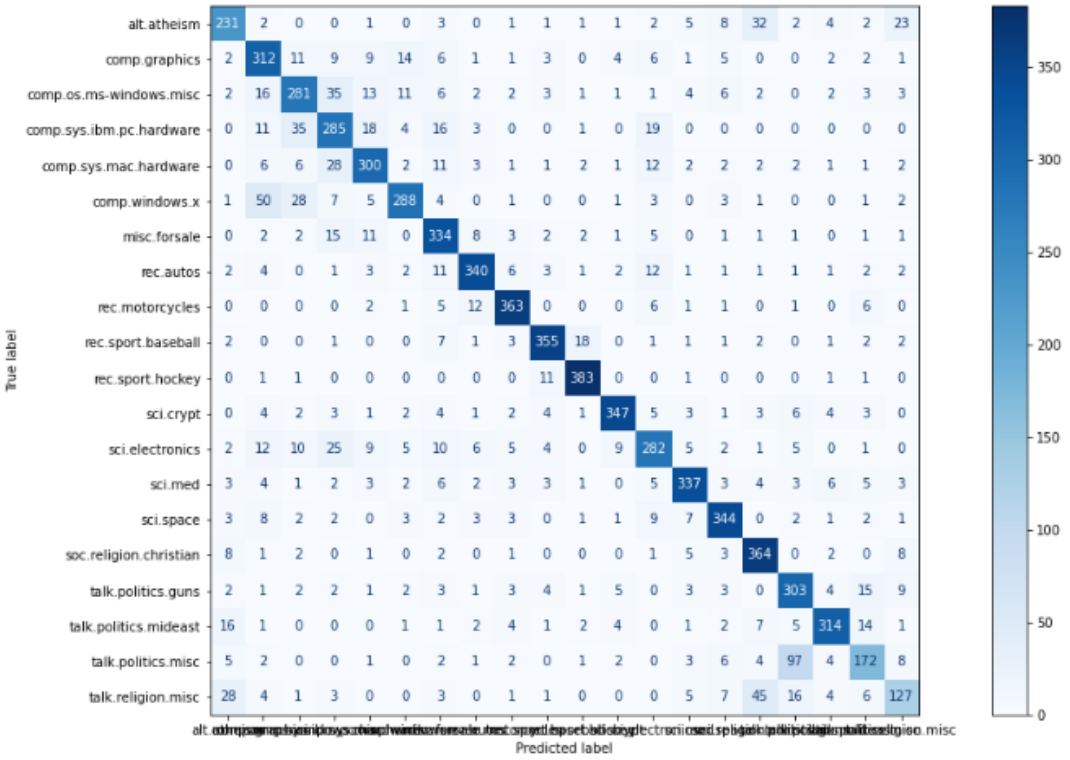
Создадим функцию для оценки каждого классификатара, а в качестве метрик оценки точности возьмём Accuracy и Confusion_matrix:

In [13]:
```
def test(model,ax):
    print(model)
    model.fit(X_train, y_train)
    print("Accuracy:", accuracy_score(y_test, model.predict(X_test)))
    #print("Confusion_matrix:", confusion_matrix(y_test, model.predict(X_test), labels = np.unique(model.predict(X_test))))
    plot_confusion_matrix(model, X_test, y_test,
                          display_labels=newsgroups_test.target_names,
                          cmap=plt.cm.Blues, ax=ax)
```

LogisticRegression:

In [14]:
```
fig, ax = plt.subplots(figsize=(20,10))
test(LinearSVC(), ax)
```

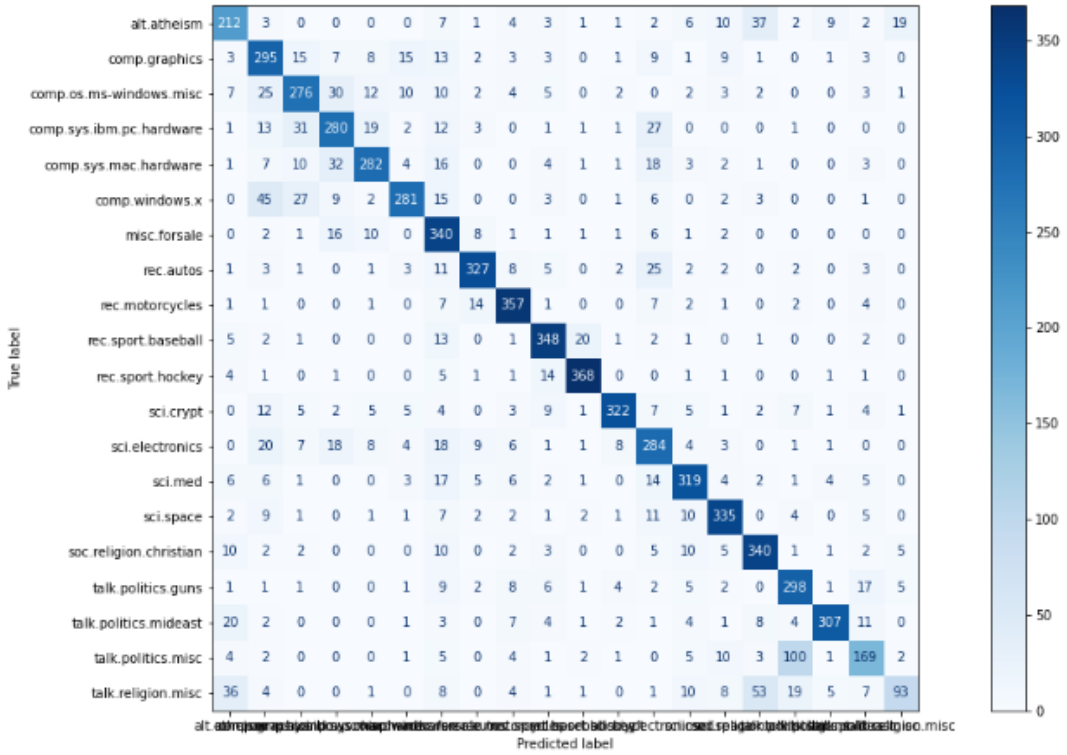
```
LinearSVC()
Accuracy: 0.8048327137546468
```



LinearSVC:

In [15]:
```
fig, ax = plt.subplots(figsize=(20,10))
test(LogisticRegression(), ax)
```

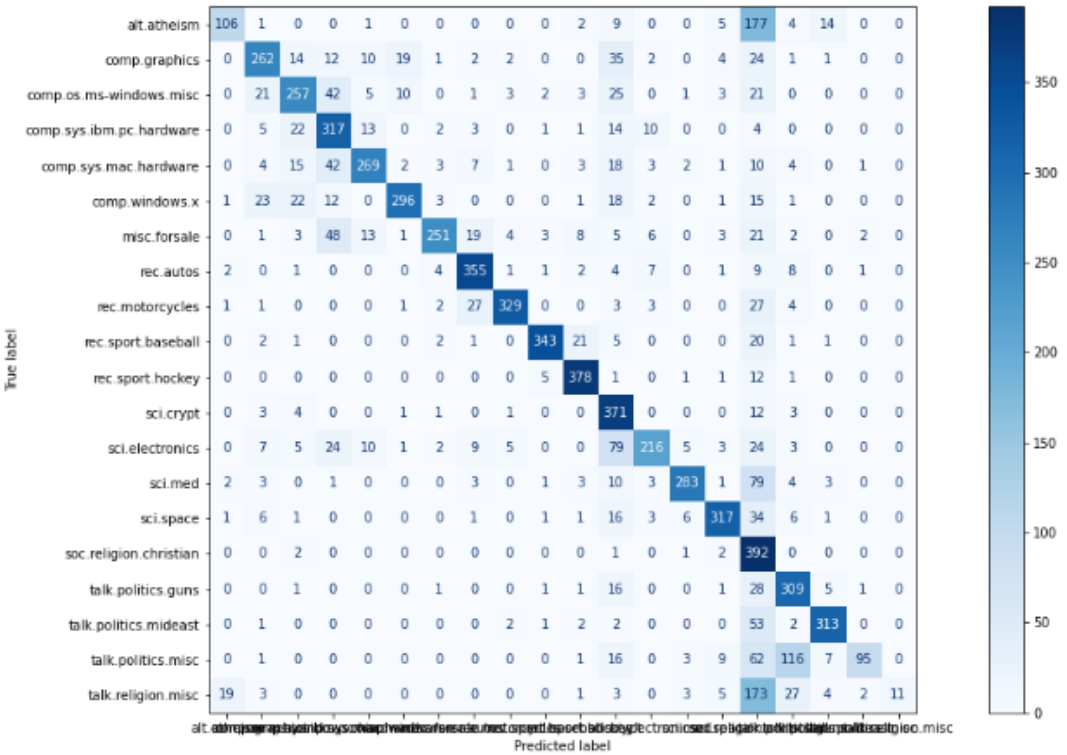
```
LogisticRegression()
Accuracy: 0.774429102496017
```



Multinomial Naive Bayes:

In [16]:
```
fig, ax = plt.subplots(figsize=(20,10))
test(MultinomialNB(),ax)
```

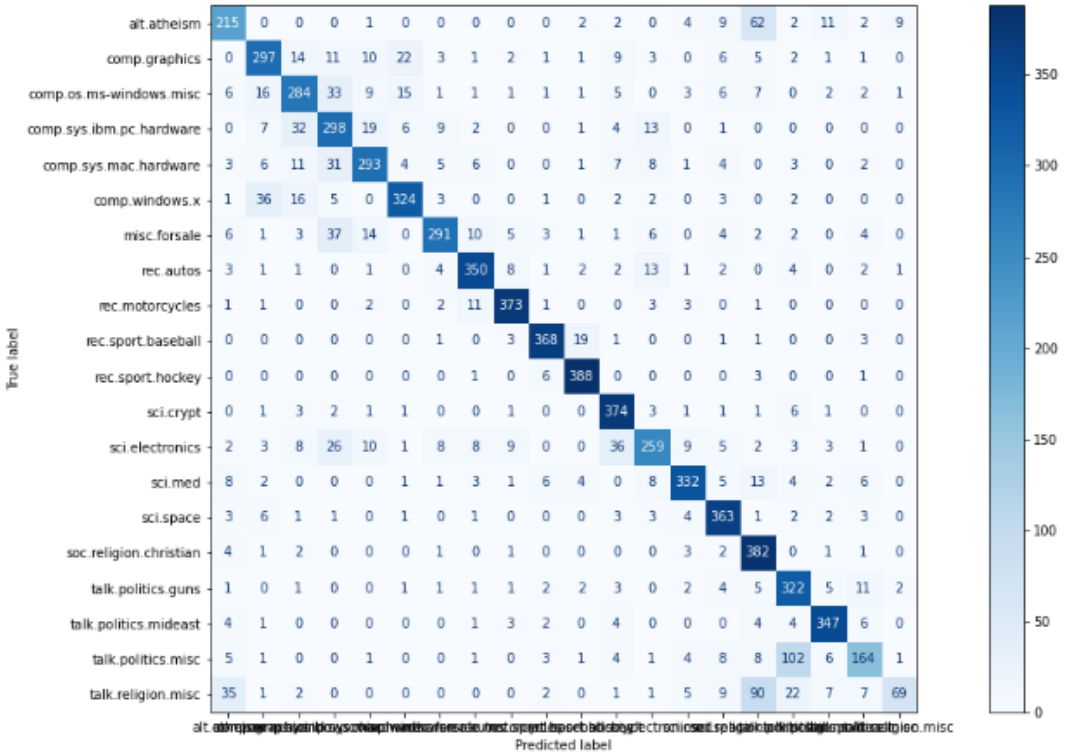
```
MultinomialNB()
Accuracy: 0.72623473181094
```



Complement Naive Bayes:

In [17]:
```
fig, ax = plt.subplots(figsize=(20,10))
test(ComplementNB(),ax)####
```

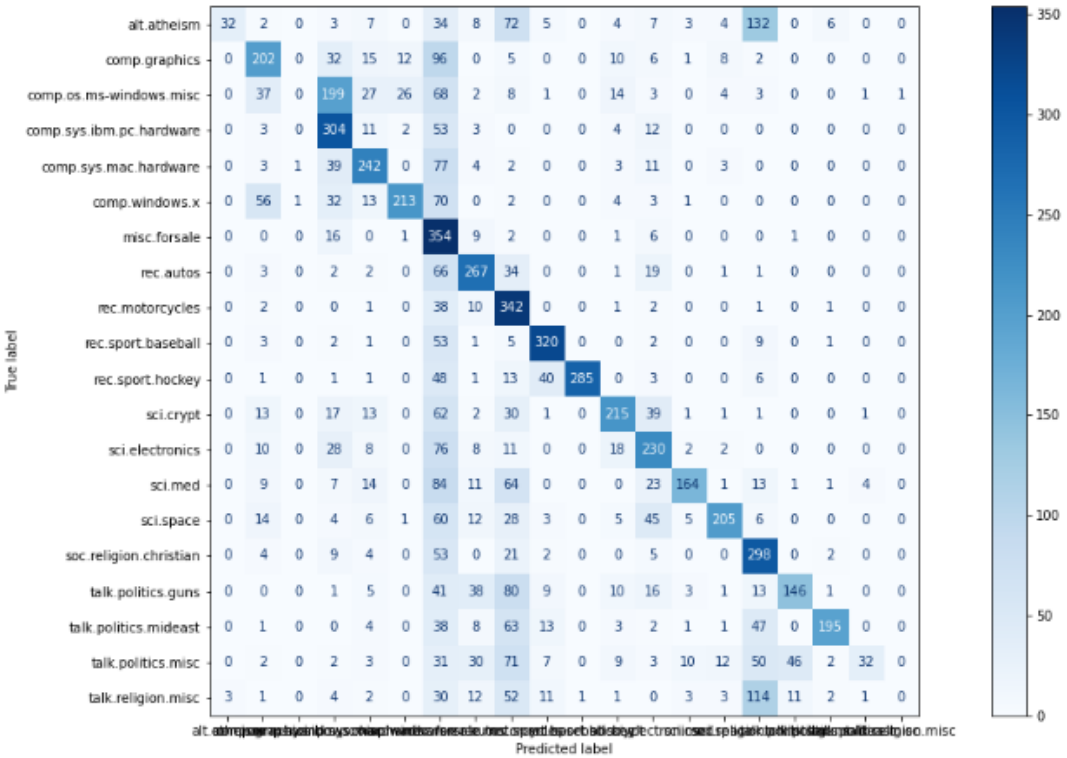
```
ComplementNB()
Accuracy: 0.8089484864577802
```



Bernoulli Naive Bayes:

In [18]:
```
fig, ax = plt.subplots(figsize=(20,10))
test(BernoulliNB(),ax)
```

```
BernoulliNB()
Accuracy: 0.5371747211895911
```



Вывод ¶

Complement Naive Bayes дал более качественную классификацию для данного набора данных.