Протестированные модели:
- Naive Bayes
- SVM

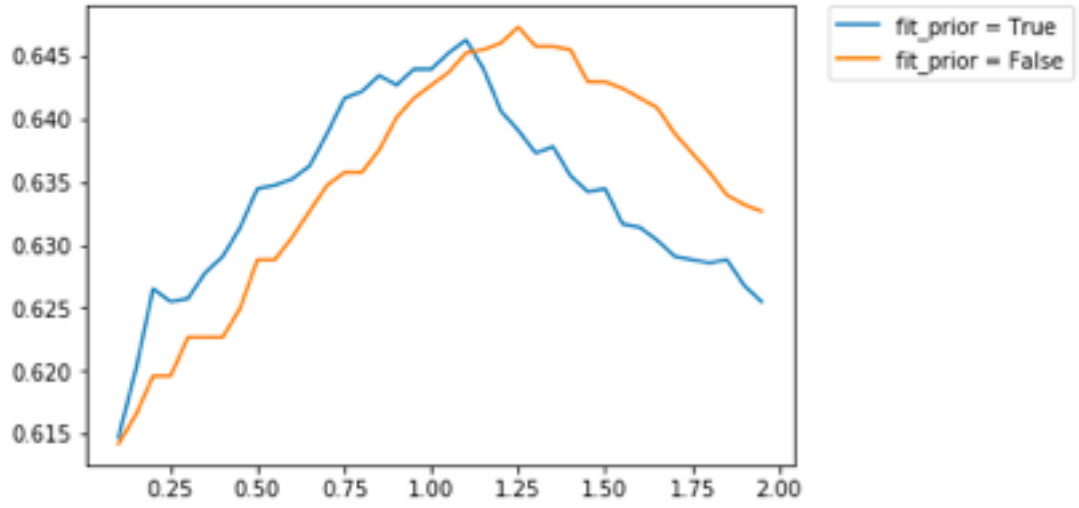
Рассмотренные вектора:
- обычный, количественный
- tf-idf
- нормализованный l1. вероятность встретить слово в тексте

Результаты:

- Naive Bayes

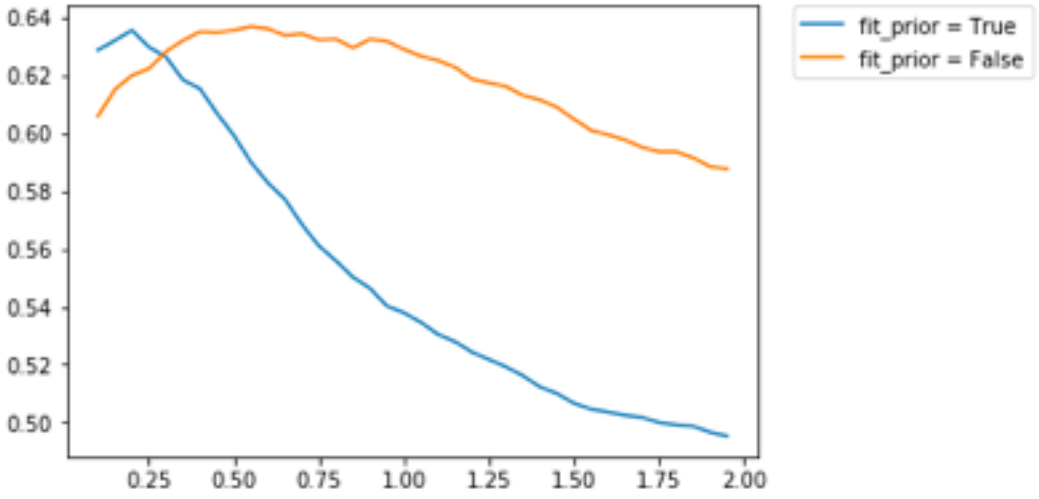
  параметры: {'fit_prior': (True, False), 'alpha': np.arange(0.1, 2, 0.05)}
  Cross-validation = 3
  по оси OX - alpha (параметр сглаживания)
  OY - f1-score (усредненное по Cross-validation)

  обычный вектор:



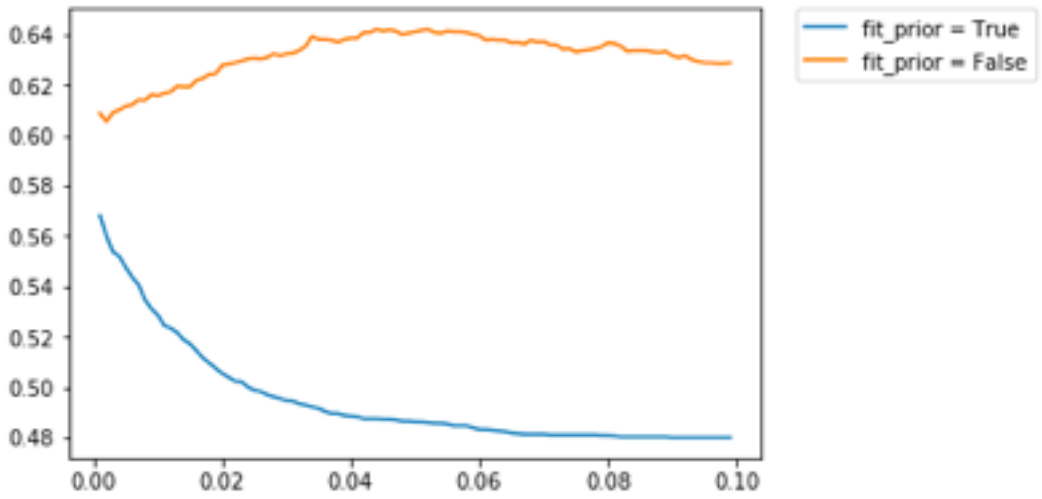

```
Train score: 0.904443873619
Test score: 0.633282309206
```

  tf-idf:




```
Train score: 0.931158489597
Test score: 0.627596763613
```

нормализованный:




```
Train score: 0.930901618289
Test score: 0.635687732342
```

- SVM:
  параметры:{'kernel': [‘rbf’, ‘linear’, ‘poly’], 'C': [0.1, 1, 10, 100, 1000]}
  Cross-validation = 3
  по оси OX - alpha (параметр сглаживания)
  OY - f1-score (усредненное по Cross-validation)

  Результаты примерно такие же.

Лучший результат среди всех моделей на тестовой выборке показал **Naive Bayes** с нормализованными векторами, **alpha = 0.044**, не обучая вероятности классов **(fit_prior = False**)