Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №1 по курсу «Искусственный интеллект»

Тема: KNN, Naive Bayes и линейные модели

Студент: А.С. Федоров

Преподаватель: Самир Ахмед

Группа: М8О-307Б-19

Дата:

Оценка:

Подпись:

Москва, 2022

Задача:

1) реализовать следующие алгоритмы машинного обучения: Linear/ Logistic Regression, SVM, KNN, Naive Bayes. Подобрать оптимальные гиперпараметры с помощью кросс-валидации. Сравнить с готовыми решениями из библиотеки sklearn. Проанализировать метрики качества моделей и полученные результаты. Сюрелизовать получившиеся модели в файлы.

Описание

Создавать классификаторы буду в отдельных классах. Для совместимости с sklearn буду наследоваться от BaseEstimator и ClassifierMixin, что понадобиться в процессе работы. Для подбора гиперпараметорв воспользуюсь функциями GridSearchCV и RandomSearchCV из sklearn. Метрики качества моделей также буду получать, с помощью готовых функций. Использую accuracy_score, recall_score, precision_score, roc_curve, roc_auc_score. Сравнивать получившиеся модели буду с готовыми решениями также из библиотеки sklearn.

Ход работы

В предыдущей лабораторной работе был проведен анализ данных, на которых требуется реализовать решение поставленной задачи. Напомню задачу: на основе косвенных данных (активное время абонента, количество проговоренных минут, количество звонков в тех поддержку и т.д.) предсказать, откажется ли абонент от услуг компании.

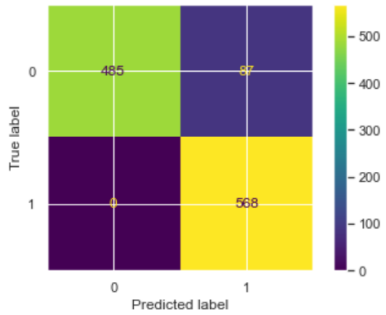
KNN

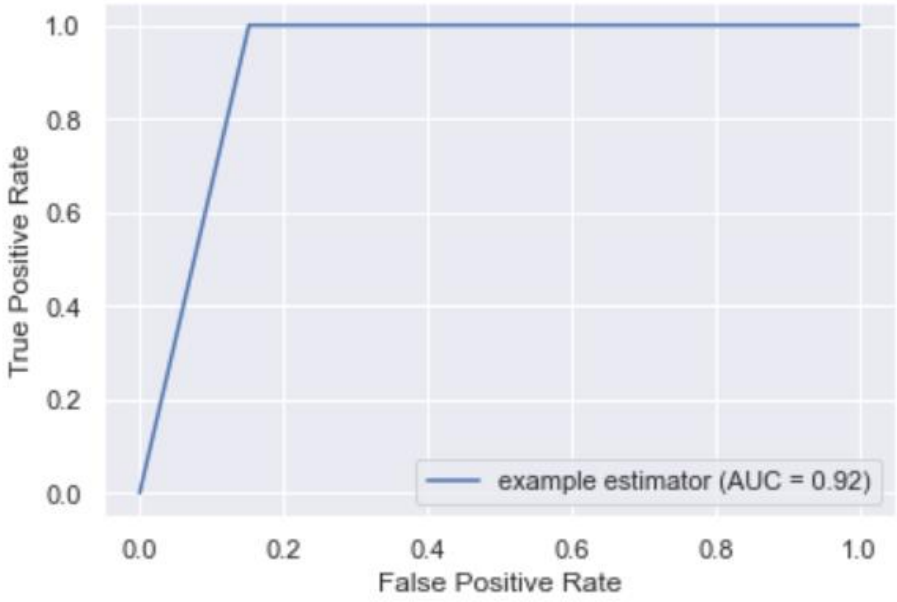
Был реализован классификатор k-ближайших соседей. «Расстояние» между объектами выборки подсчитывается как евклидово. Был произведен подбор гиперпараметра, выбор делался из нескольких кандидатов: 3, 5, 10, 50, 100. Что GridSearchCV, что RandomSearchCV сделали одинаковый выбор – 3. Точность предсказаний составила 92%. Коробочное решение мало того, что в процессе подбора гиперпараметра выбрала 3, но и точность модели оказалась такой же.

Следует отметить катастрофически больше время для классификации. Данный недостаток недопустим на практике.

```
accuracy_score: 0.9236842105263158
recall_score: 0.9236842105263158
precision_score: 0.933820811570912
roc_auc_score: 0.923951048951049
roc_auc_curve:
```

confusion matrix:





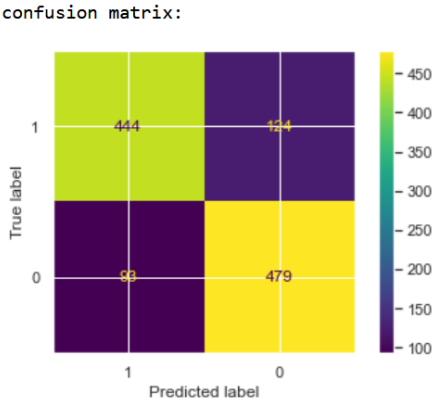
Naive Bayes

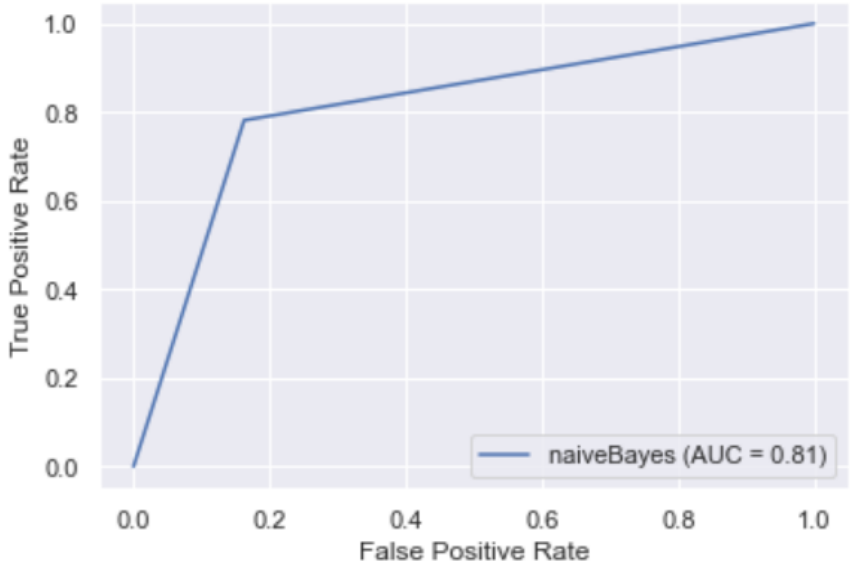
Исходя из предположения, что признаки объектов независимы и распределены нормально, что является довольно сильным утверждением. Могу применить теорему Байеса и, как следствие, предсказывать принадлежность объекта к классу. Реализация метода подсчитывает матожидания и дисперсии для всех классов по всем признакам и вычисляет плотности вероятности исходя из предположения, что распределение имеет нормальный характер. Наиболее правдоподобный класс будет иметь наибольшее произведение по всем плотностям вероятности по всем признакам.

Приятно отметить, что модель не имеет никаких гиперпараметров. И хоть предположения о признаках довольно сильные, точность предсказаний оказалась выше 80%, что довольно неплохо. Коробочное решение дает результат еще лучше – 92%. Возможно, это связанно с тем, что там не отталкиваются от предположения, что все признаки распределены нормально, а пытаются аппроксимировать распределения.

```
accuracy_score: 0.8096491228070175
recall_score: 0.8096491228070175
precision_score: 0.8105316470345193
roc_auc_score: 0.809551364128829
roc_auc_curve:
```

confusion matrix:





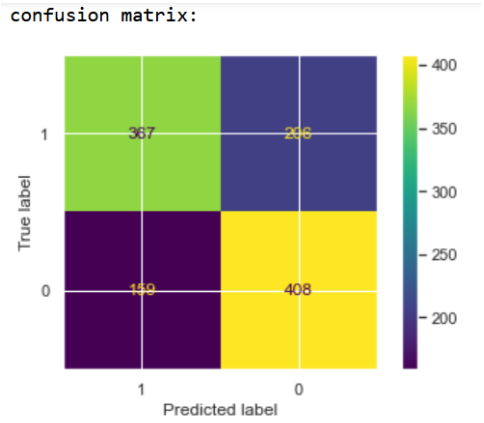
Logit Model

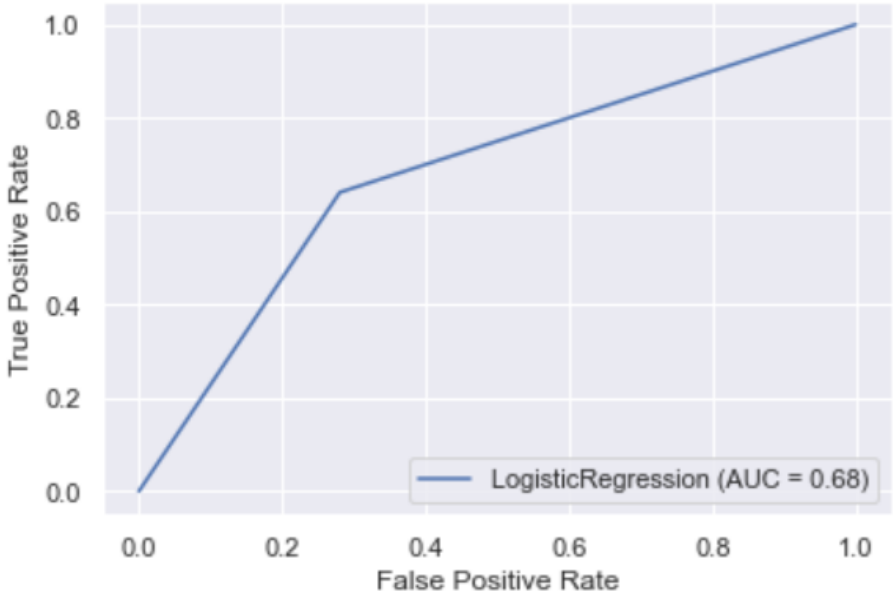
Реализовал логистическую регрессию. Обучение весов происходит методом градиентного спуска. Для предсказаний класс имеет два метода: predict для предсказания лейблов и predictProba. Гиперпараметры модели: число эпох, learning rate, сила регуляризации. Также для обучения был реализован специальный класс для генерации батчей по выборке. Размер батча также можно настраивать, как гиперпараметр модели. Конкретные значения для перебора приведены ниже.

```
parameters = {'classifier__alpha': [0.01, 0.1, 0.3],
              'classifier__batchSize': [256, 512, 1024],
              'classifier__epoches': [100, 300, 1000],
              'classifier__lr': [0.00001, 0.0001, 0.001, 0.01]}
```

В результате подбора оптимальных параметров была получена точность в 67%. Коробочное решение явно обучается не методом градиентного спуска, а также не использует батчи. Поэтому ее результат несколько лучше 74%.

```
accuracy_score: 0.6798245614035088
recall_score: 0.6798245614035088
precision_score: 0.6811943029009588
roc_auc_score: 0.680032687886091
```





SVM

Реализация аналогична логистической регрессии. Изменен только метод обучения. согласно формуле.

$$L(w, x, y) = \lambda \|w\|_2^2 + \sum_i \max(0, 1 - y_i \langle w, x_i \rangle)$$

Что обеспечивает более оптимальное расположение разделяющей плоскости, как можно более равноудаленной от крайних объектов классов. Гиперпараметры такие же, как и у логистической регрессии. Подбор происходит из аналогичного набора.

Сериализация

Лучшие модели всех классов были стерилизованы с помощью pickle в бинарные файлы. Данные файлы можно считать программным продуктом и результатом работы.

Анализ результатов

Хоть результаты KNN оказались лучшими, ввиду огромного времени, затрачиваемого на вычисление результата, данная модель неприменима. Поэтому лучшей моделью можно считать **Naive Bayes**, так как ее точность оказалась наивысшей. Хоть остальные метрики были проанализированы и приняты к сведенью, но в рамках задачи не считаю их исчерпывающими.

Вывод

В ходе выполнения лабораторной работы был проведен анализ работы четырех классификаторов: KNN, Naive Bayas, Logit Regression, SVM. Из всех моделей лучше всего себя показала Naive Bayas. Ее точность составила свыше 80%. Остальные модели оказались несколько хуже (примерно 70-75%). Однако, KNN показал гораздо более хороший результат – 92%. Но она неприменима, по указанным в отчете причинам. Возможно можно уменьшить каким-то образом выборку с которой KNN сравнивает, оставив наиболее репрезентативных, но это требует более тщательного анализа.

Для задачи предсказания оттока абонентов, признано приемлемым использовать Naive Bayas. Интересный результат, учитывая, что для применения такой модели необходимо сделать несколько сильный предположений.