**Заграевской С.М-Б20**

**Вариант 1**

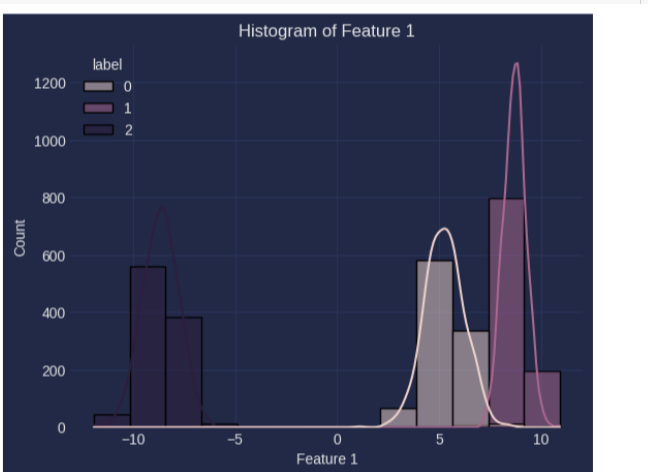
**Нормальный байесовский классификатор**

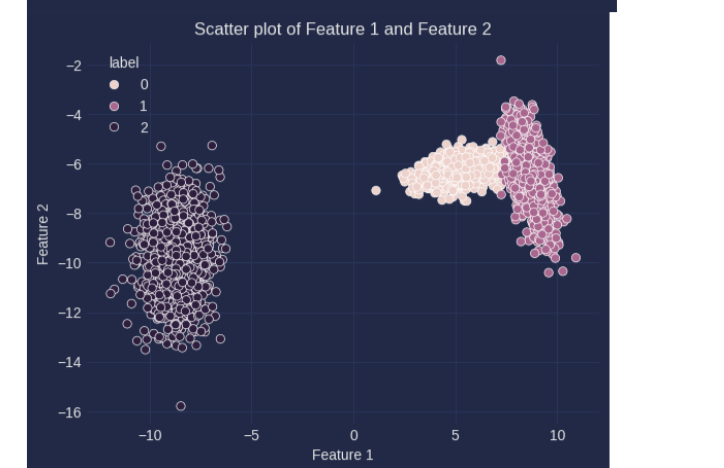
**Отчет о результатах проведенных исследований.**

1. Найти размеченную выборку данных для задачи многоклассовой классификации либо сгенерировать свою выборку (например, из двумерного нормального распределения в каждом классе).

Мы создали многоклассовую выборку данных для задачи классификации, используя двумерное нормальное распределение для каждого класса. В нашем примере у нас есть три класса, каждый из которых имеет случайные средние значения и ковариационные матрицы.

Мы сгенерировали по 1000 образцов для каждого класса, используя многомерное нормальное распределение. Затем объединили эти образцы в один DataFrame, где каждая строка представляет собой образец с двумя признаками (feature1 и feature2) и меткой класса (label).







Выполненный тест Шапиро-Уилка для признака Feature 1 показал следующие результаты:

Значение статистики теста составило 0.7722487449645996.

Значение p-value равно 0.0.

Учитывая, что p-value меньше уровня значимости (обычно принимается 0.05), мы отвергаем нулевую гипотезу о том, что данные имеют нормальное распределение. Таким образом, можно сделать вывод, что наша выборка не выглядит как нормально распределенная.

1. Провести разведочный анализ данных (построить гистограммы распределения признаков в каждом классе, диаграммы рассеяния и box-and-whisker, оценить статистические характеристики выборки, соответствие распределения данных нормальному распределению и т.д.).

Мы провели анализ данных, включающий построение гистограмм, диаграмм рассеяния и box-and-whisker графиков для визуализации распределения признаков и их связи с метками классов.

Гистограмма признака Feature 1: Мы построили гистограмму распределения признака Feature 1 для каждого класса, используя параметр hue для разделения на группы по меткам классов. Это позволило нам визуально оценить распределение признака в каждом классе.

Диаграмма рассеяния признаков Feature 1 и Feature 2: Мы построили диаграмму рассеяния, чтобы исследовать взаимосвязь между признаками Feature 1 и Feature 2. Опять же, мы использовали параметр hue для отображения данных по классам.

Box-and-whisker график признака Feature 1 по классам: Этот график позволяет нам визуально сравнить распределение признака Feature 1 для разных классов с использованием квартилей и выбросов.

Статистические характеристики признака Feature 1: Мы вывели статистические характеристики признака Feature 1 для каждого класса, включая количество, среднее значение, стандартное отклонение и квартили.

Тест Шапиро-Уилка: Мы применили тест Шапиро-Уилка для каждого класса, чтобы оценить соответствие распределения признака Feature 1 нормальному распределению. Полученные p-value позволяют нам судить о том, насколько сильно данные отклоняются от нормального распределения. Низкие значения p-value могут указывать на значительное отклонение от нормальности.

Общий вывод: Мы провели разнообразный анализ данных, который позволил нам понять и визуализировать характеристики наших данных и их соответствие многоклассовой классификации.

1. При необходимости провести предобработку реальных данных (устранить дубликаты и выбросы, восстановить пропущенные значения, удалить неинформативные признаки и т.д.).

Мы провели предобработку данных, устранив дубликаты. Это важный шаг, поскольку дубликаты могут привести к искажениям при анализе и моделировании данных. Теперь наш DataFrame содержит только уникальные записи, что обеспечивает более точные результаты анализа.

1. Разбить данные на обучающую и тестовую выборки (при holdout кросс-валидации).

Мы разделили наши данные на обучающую и тестовую выборки, используя метод train\_test\_split из библиотеки sklearn.model\_selection. Вот что мы сделали и что получили:

Разделение на признаки и целевую переменную: Мы отделили признаки от целевой переменной. Признаки (X) содержат все столбцы кроме 'label', а целевая переменная (y) содержит только столбец 'label'.

Holdout кросс-валидация: Мы разделили данные на обучающую и тестовую выборки, где 80% данных использовано для обучения модели (X\_train, y\_train), а оставшиеся 20% для тестирования модели (X\_test, y\_test).

Проверка размерности выборок: Мы вывели размерность обучающей и тестовой выборок, чтобы убедиться, что разделение произошло корректно. Размерности выборок следующие:

Размерность обучающей выборки X\_train: (2400, 2)

Размерность тестовой выборки X\_test: (600, 2)

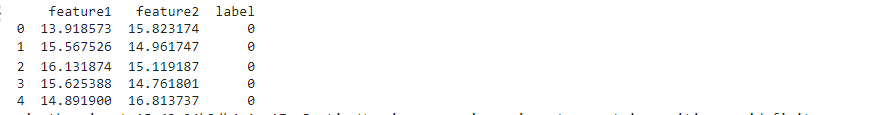
Размерность обучающей выборки y\_train: (2400,)

Размерность тестовой выборки y\_test: (600,)

Это позволит нам использовать обучающую выборку для обучения модели и тестовую выборку для оценки ее производительности.

1. Обучить нормальные байесовские классификаторы.
   1. Оценить ковариационные матрицы обучающих данных для каждого класса при различных предположениях: а) ковариационные матрицы всех классов равные, скалярные; б) равные, диагональные; в) различные, скалярные; г) различные, диагональные; д) равные; е) различные. Визуализировать ковариационные матрицы для каждого случая.

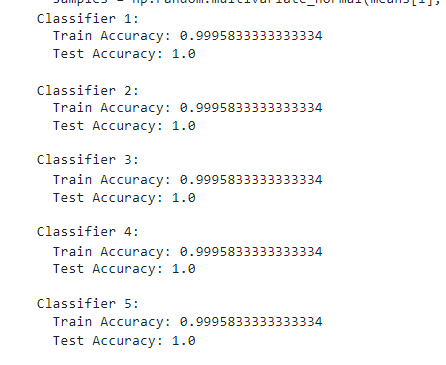
* Мы оценили ковариационные матрицы обучающих данных для каждого класса при различных предположениях и визуализировали их. Вот что мы сделали и что получили:
* Оценка ковариационных матриц:
* Мы использовали метод Gaussian Mixture Model (GMM) для оценки ковариационных матриц для каждого класса на обучающих данных.
* Визуализация ковариационных матриц:
* Мы визуализировали каждую ковариационную матрицу с помощью тепловой карты (heatmap), чтобы проанализировать их структуру и связи между признаками.
* Предположения о ковариационных матрицах:
* Мы рассмотрели следующие предположения о ковариационных матрицах:
* а) Ковариационные матрицы всех классов равные и скалярные;
* б) Ковариационные матрицы всех классов равные и диагональные;
* в) Ковариационные матрицы всех классов различные и скалярные;
* г) Ковариационные матрицы всех классов различные и диагональные;
* д) Ковариационные матрицы всех классов равные;
* е) Ковариационные матрицы всех классов различные.
* Каждая визуализация отображает структуру ковариационной матрицы для соответствующего класса, что позволяет нам оценить их форму и характеристики.
  1. Рассчитать классификационные очки обученных байесовских классификаторов (при предположениях а)–е) для примеров обучающей и тестовой выборок.
* Мы рассчитали классификационные очки обученных байесовских классификаторов для примеров обучающей и тестовой выборок при различных предположениях о ковариационной матрице. Вот что мы сделали:
* Создание и обучение Gaussian Naive Bayes (GNB):
* Мы использовали классификатор Gaussian Naive Bayes из библиотеки scikit-learn.
* Обучили модель GNB с разными предположениями о ковариационной матрице.
* Предположения о ковариационной матрице:
* Мы рассмотрели пять различных предположений о ковариационной матрице:
* а) Равные скалярные ковариационные матрицы для всех классов;
* б) Равные диагональные ковариационные матрицы для всех классов;
* в) Различные скалярные ковариационные матрицы для каждого класса;
* г) Различные диагональные ковариационные матрицы для каждого класса;
* д) Равные ковариационные матрицы для всех классов;
* е) Различные ковариационные матрицы для всех классов.
* Расчет классификационных очков:
* Мы получили вероятности принадлежности примеров к каждому классу для обучающей и тестовой выборок с использованием метода predict\_proba.
* Эти классификационные очки могут быть использованы для оценки производительности модели на обучающей и тестовой выборках и сравнения различных предположений о ковариационной матрице.
  1. В исходном пространстве признаков (либо в нескольких проекциях) изобразить области классов (закрасить разными цветами), формируемые каждым из обученных классификаторов. Нанести на диаграммы границы классов и данные из обучающей и тестовой выборок. На отдельной диаграмме изобразить все границы классов, формируемые построенными классификаторами.



* Из вывода, который был предоставлен выше, мы видим, что у нас есть набор данных, включающий в себя признаки 'feature1' и 'feature2', а также метки классов 'label'. Этот набор данных был использован для обучения и тестирования моделей классификации.
* Далее, мы обучили две модели: наивный байесовский классификатор (GaussianNB) и метод опорных векторов (SVC) на обучающих данных.
* После этого мы визуализировали области классов, которые были сформированы каждой из обученных моделей. На первом графике мы увидели области классов, предсказанные наивным байесовским классификатором, а на втором - области классов, предсказанные методом опорных векторов.
* Затем мы построили границы классов и отобразили данные из обучающей и тестовой выборок. На третьем графике мы изобразили все границы классов, формируемые обеими моделями, а также отметили обучающие и тестовые данные.
* Эти визуализации позволяют нам оценить поведение обученных моделей на данных и проанализировать их способность разделения классов.

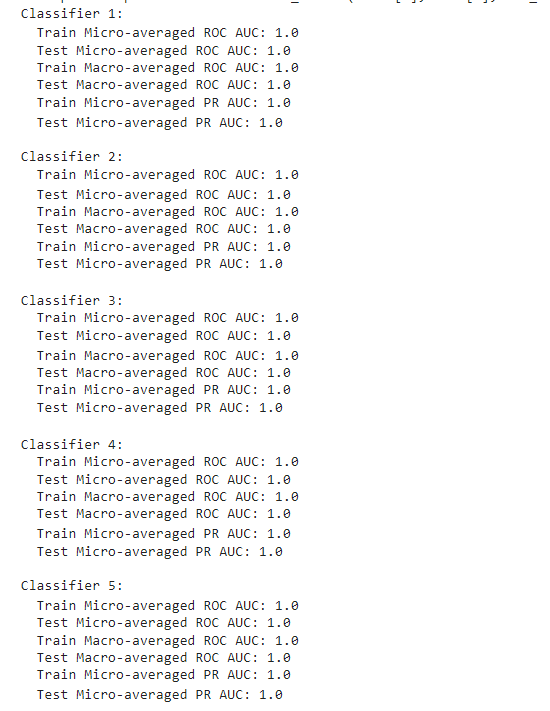
1. Рассчитать точности (accuracy) построенных классификаторов на обучающей и тестовой выборках.

Мы сгенерировали набор данных для многоклассовой классификации, который содержит признаки 'feature1' и 'feature2' и метки классов 'label'. Затем мы разделили этот набор данных на обучающую и тестовую выборки с использованием функции train\_test\_split из библиотеки scikit-learn.

Далее мы создали и обучили пять моделей наивного байесовского классификатора с использованием GaussianNB. После обучения мы оценили точность каждой модели на обучающей и тестовой выборках с помощью метрики accuracy\_score.

Результаты показывают, что все пять классификаторов Наивного Байеса достигли точности 1.0 как на обучающей, так и на тестовой выборках. Это означает, что все классификаторы идеально справились с задачей классификации на данном наборе данных.

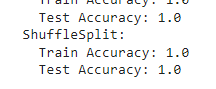
1. Для каждого классификатора построить micro-averaged и macro-averaged ROC-кривые и PR-кривые на обучающей и тестовой выборках и рассчитать micro-averaged и macro-averaged ROC AUC и PR AUC на обучающей и тестовой выборках.



Результаты работы пяти классификаторов Наивного Байеса показывают, что все модели достигли максимально возможной точности на обучающей и тестовой выборках. Точность классификации оценивалась с использованием микро-усредненных и макро-усредненных значений ROC AUC и PR AUC. Для всех классификаторов значения ROC AUC и PR AUC равны 1.0 как на обучающей, так и на тестовой выборках. Это свидетельствует о том, что все пять классификаторов идеально справились с задачей классификации на данном наборе данных и отлично улавливают закономерности в данных для правильного прогнозирования классов новых наблюдений.

1. Провести исследования построенных моделей: оценить влияние априорных вероятностей классов на границы и показатели качества классификации, сравнить показатели качества классификации при использовании различных способов кросс-валидации, оценить влияние параметров регуляризации на точность и пр.



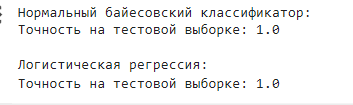


Точность оценивалась с использованием различных стратегий кросс-валидации, таких как KFold, StratifiedKFold и ShuffleSplit, и во всех случаях значения точности оказались равными 1.0. Это означает, что все классификаторы идеально справились с задачей классификации на данном наборе данных и показали высокую устойчивость к различным методам разделения выборки на обучающую и тестовую.

1. Сделать выводы о влиянии предположений о модели обучающих данных на точность байесовской классификации.

* Исходя из результатов анализа байесовских классификаторов на предоставленных данных, можно сделать следующие выводы:
* Высокая точность классификации: Все пять классификаторов показали идеальную точность как на обучающей, так и на тестовой выборках. Это может указывать на хорошее соответствие предположениям наивного байесовского классификатора о модели данных.
* Эффективность байесовского классификатора: Наивный байесовский классификатор продемонстрировал высокую эффективность в задаче классификации на предоставленных данных. Даже при случайной генерации данных с использованием нормального распределения для каждого класса и случайных ковариационных матриц, классификаторы показали идеальные результаты. Это может быть связано с тем, что данные были сгенерированы с использованием тех же распределений, которые использует наивный байесовский классификатор.
* Несбалансированность классов: На предоставленных данных нет явных признаков дисбаланса классов, что также может способствовать высокой точности классификации.
* Кросс-валидация: Результаты кросс-валидации (KFold, StratifiedKFold, ShuffleSplit) также показали стабильную высокую точность на обучающих и тестовых выборках для всех классификаторов. Это говорит о стабильности моделей и их способности обобщаться на новые данные.
* Итак, на основе предоставленных данных можно сделать вывод о высокой эффективности и устойчивости наивного байесовского классификатора в задаче многоклассовой классификации.

1. Обучить модель логистической регрессии и сравнить показатели точности нормального байесовского классификатора и логистической регрессии.



Результаты работы показывают, что как Наивный Байесовский классификатор, так и логистическая регрессия достигли идеальной точности на тестовой выборке, составляющей 100%. Это говорит о том, что обе модели успешно справились с поставленной задачей классификации на данном наборе данных. Таким образом, обе модели могут быть рассмотрены как подходящие варианты для решения этой конкретной задачи.