Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет

«Высшая школа экономики»

Факультет компьютерных наук

Отчёт о прохождении производственной (научно-исследовательской) практики

Выполнил студент:	Пащенко Никита Станиславович
Образовательной программы:	Интеллектуальные системы и структурный анализ
Организация:	НИУ ВШЭ
Отчет проверил руководитель практики от организации:	Игнатов Д. И.
Отчет проверил руководитель практики от НИУ ВШЭ:	Макаров И. А.

Подпись студента: 

Москва, 2021

Содержание

Введение

Производственная практика проходилась в рамках НИУ ВШЭ, работа была посвящена тестированию различных классификаторов на наборах данных, в которых представлены SNP маркеры индивидуумов, склонных и несклонных к ишемическому инсульту. Основными целями являлись поиск оптимального алгоритма классификации и настройка его гиперпараметров. В процессе работы были решены следующие задачи:

- протестированы различные классификаторы на предоставленных наборах данных;
- проведен поиск оптимальной конфигурации параметров лучшего из рассмотренных алгоритмов;
- протестированы различные подходы по уменьшению эффекта переобучения.

Описание использованных наборов данных

Тестирование классификаторов было проведено на двух различных наборах данных, характеристики которых представлены в таблице ниже.

Таблица 1 – Характеристики представленных наборов данных

Набор 1				
Кол-во наблюдений	Кол-во здоровых	Кол-во больных	Кол-во признаков	Соотношение здоров./больн
1323	413	910	85142	0.69
Набор 2				
1228	305	923	67925	0.33

Как можно заметить, данные являются несбалансированными, что может затруднить работу некоторых алгоритмов.

Также стоить отметить большое количество признаков. В качестве самих признаков выступают так называемые SNP маркеры. SNP — однонуклеотидный полиморфизм, представляет собой вариацию в одной позиции последовательности ДНК среди индивидуумов, некоторые из которых связаны с определенными заболеваниями, тем самым SNP маркеры могут быть использованы в качестве фактора риска развития заболевания [1]. Каждый из признаков принимает значения: 0, 1 или 2, где 0 — полное совпадение снипа, 1 — отличие только в одном значении, 2 — отличие в двух.

Для получения оценок важности признаков, т.е. выявления релевантных SNP маркеров, в первую очередь необходимо обучить классификаторы на имеющихся данных таким образом, чтобы они хорошо различали индивидов склонных к ишемическому инсульту от несклонных.

Тестирование различных классификаторов

Для поиска лучшего классификатора протестированы следующие алгоритмы:

- логистическая регрессия (LogisticRegression);
- метод k-ближайших соседей (KneighborsClassifier);
- метод опорных векторов (SVC);
- дерево решений (DecisionTreeClassifier);
- случайный лес (RandomForestClassifier) [2];
- градиентный бустинг на решающих деревьях (XGBoost) [3].

Для оценки качества предсказаний алгоритмов использовались стандартные метрики: доля правильных ответов (Accuracy), точность (Precision), полнота (Recall) и F1-мера (F1 score). Данный набор метрик был в первую очередь необходим для отслеживания эффекта переобучения на каком-то из классов, что характерно для несбалансированных наборов данных. Оценка алгоритмов происходила по схеме скользящего контроля по 5 блокам с сохранением искомого количественного отношения между классами. Таким образом каждая метрика оценивалась и затем усреднялась по 5 блокам разбиения.

Результат работы алгоритмов на первом наборе данных

Результат работы классификаторов на первом наборе данных приведен на рисунке ниже.

	accuracy	precision	recall	F1-score	recall -	precision -
DummyClassifier	0.580463	0.693883	0.697802	0.695771	0.321951	0.326536
LogisticRegression	0.886655	0.862848	0.997802	0.924601	0.641963	0.993103
SVC	0.692361	0.691015	1	0.817264	0.0144578	0.2
DecisionTreeClassifier	0.932075	0.961051	0.948352	0.949921	0.896386	0.916439
RandomForestClassifier	0.968302	0.9625	1	0.97931	0.898795	1
KNeighborsClassifier	0.766366	0.80069	0.906593	0.845637	0.456715	0.618423
XGBClassifier	0.996978	1	0.995604	0.997787	1	0.990614

Рисунок 1 – Результат работы классификаторов на первом наборе данных

Как можно заметить, лучше всего показала себя модель XGBoost. Стоить также отметить, что и более простые модели показали достаточно высокие значения метрик, лучшими из которых являются алгоритмы, основанные на решающих деревьях. Метод опорных векторов имеет наихудший результат среди представленных моделей, более того он имеет наибольшую вычислительную сложность.

Результат работы алгоритмов на втором наборе данных

Для второго набора был использован аналогичный список моделей. Результат работы классификаторов представлен на рисунке ниже:

	accuracy	precision	recall	F1-score	recall -	precision -
DummyClassifier	0.635182	0.755443	0.760564	0.757538	0.255738	0.26358
LogisticRegression	0.702778	0.764414	0.874371	0.815481	0.183607	0.321612
SVC	0.751628	0.751628	1	0.858205	0	0
DecisionTreeClassifier	0.642535	0.763299	0.760652	0.761256	0.285246	0.282338
RandomForestClassifier	0.751628	0.751628	1	0.858205	0	0
KNeighborsClassifier	0.702014	0.751402	0.901475	0.819552	0.0983607	0.261227
XGBClassifier	0.674797	0.751196	0.848649	0.796954	0.147541	0.243243

Рисунок 2 – Результат работы классификаторов на втором наборе данных

Как видно по результатам, метрики на втором наборе данных намного хуже, чем в первом. Более того явно заметен эффект переобучения на мажоритарном классе, что может быть связано с тем, что данный набор данных является ещё более несбалансированным. Тем не менее, лучше всего себя показали модели: XGBoost и LogisticRegression.

Для избавления от эффекта переобучения на большем классе были отрегулированы веса обратно пропорционально частотам классов во входных данных, за счет этого накладывались большие штрафы за ошибку в слабо представленном классе в функции ошибок. К сожалению, данных подход не смог полностью решить проблему, так как теперь переобучение происходит и в сторону меньшего класса.

Поиск оптимальной конфигурации параметров для модели XGBoost и LogisticRegression

Для моделей XGBoost и LogisticRegression, показавших наилучший результат, был проведен подбор оптимальных параметров. Поиск был осуществлен с помощью прямого перебора по сетке значений гиперпараметров. Для модели XGBoost были выбраны следующие параметры:

- глубина дерева (*max_depth*);
- мин. сумма весов, необходимая для создания листа (*min_child_weight*);
- веса, используемые для балансировки классов (*scale_pos_weight*);
- соотношение подвыборок (*subsample*).

При подборе оптимальных гиперпараметров для LogisticRegression учитывались следующие:

- тип регуляризации (*penalty*);
- значение константы регуляризации (*C*);
- веса, используемые для балансировки классов (*class_weight*);

Окончательные показатели метрик после отбора оптимальных гиперпараметров представлены на рисунке ниже.

	accuracy	precision	recall	F1-score	recall -	precision -
LogisticRegression	0.75326	0.752859	1	0.859006	0.00655738	0.4

Рисунок 3 – Результат работы LogisticRegression на втором наборе данных

	accuracy	precision	recall	F1-score	recall -	precision -
XGBClassifier	0.704423	0.761594	0.883026	0.817807	0.163934	0.322101

Рисунок 4 – Результаты работы XGBoost на втором наборе данных

Для сравнения лучших моделей XGBoost и LogisticRegression представим на рисунках ниже ROC-кривые обучения в паре с площадью под кривой (AUC), т.е. ROC-AUC показатели.

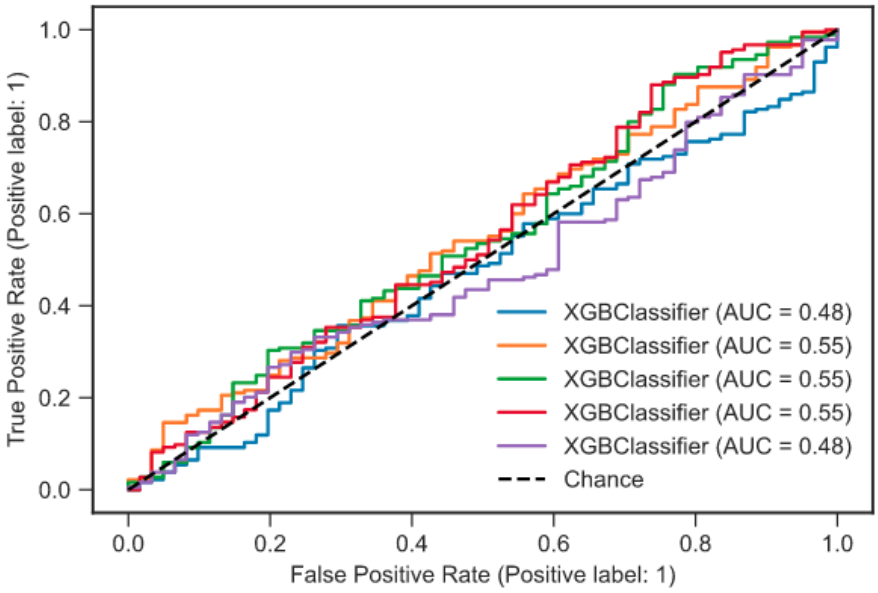

Рисунок 5 – ROC-кривые модели XGBoost на втором наборе данных (KFold=5)

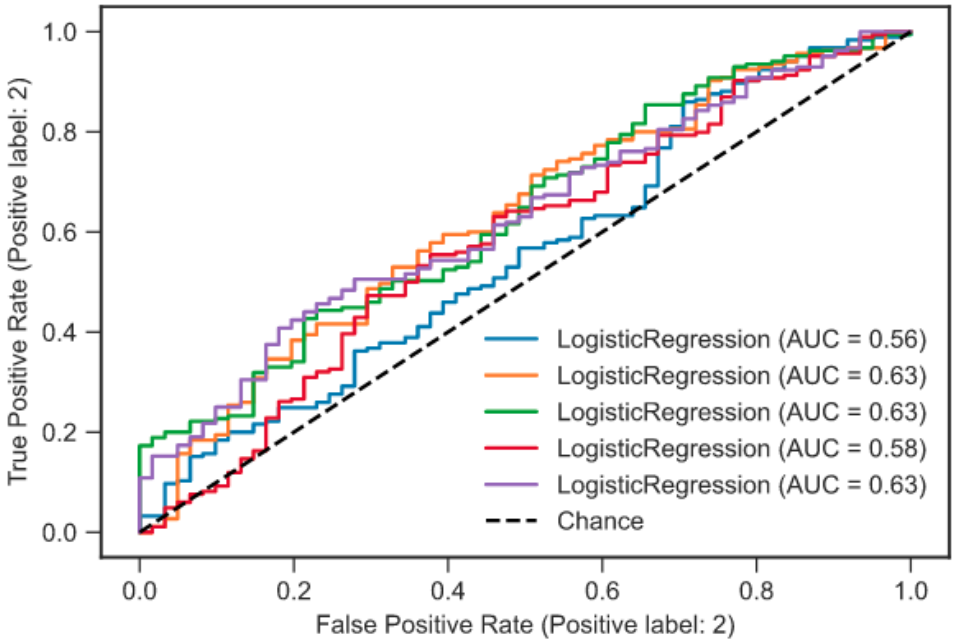

Рисунок 6 – ROC-кривые LogisticRegression на втором наборе данных (KFold=5)

Тестирование различных подходов для уменьшения эффекта переобучения

Для уменьшения эффекта переобучения на втором наборе данных были протестированы несколько общепринятый подходов, одним из которых является уменьшение размерности. В качестве алгоритма уменьшения размерности был выбран метод главных компонент (PCA). Тестирование проводилось с использованием модели LogisticRegression, показавшей наилучший результат на втором наборе данных.

Алгоритм PCA не решил проблему переобучения на мажоритарном классе, более того при небольшом количестве главных компонент предсказания ещё более близки к предсказаниям случайного классификатора. Результирующие метрики LogisticRegression классификатора для разного количества главных компонент представлены на рисунке ниже.

Number of components	accuracy	precision	recall	F1-score	recall -	precision -
DummyClassifier	0.621377	0.755631	0.73245	0.743645	0.285246	0.265511
2	0.537438	0.775802	0.540564	0.636649	0.527869	0.275724
10	0.496768	0.734602	0.517926	0.60697	0.432787	0.228177
50	0.56435	0.77613	0.595893	0.672128	0.468852	0.272034
100	0.548824	0.75317	0.594736	0.664309	0.409836	0.250287
1000	0.683222	0.766569	0.832027	0.797706	0.232787	0.314855
full	0.75326	0.752859	1	0.859006	0.00655738	0.4

Рисунок 7 – Результат работы LogisticRegression на втором наборе данных при различном количестве главных компонент

Далее были протестированы алгоритмы сэмплирования: SMOTE для разного количества ближайших соседей (k_neighbors) и RandomUnderSampler [4]. Основная цель данного подхода заключаются в корректировке обучающей выборки с целью балансировки распределения классов в исходном наборе данных. Результирующие метрики LogisticRegression классификатора для разных подходов сэмплирования представлены на рисунке ниже.

	accuracy	precision	recall	F1-score	recall -	precision -
DummyClassifier	0.621377	0.755631	0.73245	0.743645	0.285246	0.265511
RandomUnderSampler	0.52623	0.520202	0.590164	0.551719	0.462295	0.536037
SMOTE k=1, 2, 3, 5, 10	0.5	0.5	1	0.666667	0	0
LogisticRegression	0.75326	0.752859	1	0.859006	0.00655738	0.4

Рисунок 8 – Результат работы LogisticRegression на втором наборе данных при различных подходах к сэмплированию

Как видно по результатам выше, алгоритмы сэмплирования также не решили проблему переобучения, более того при использовании SMOTE алгоритма происходит полное переобучение на мажоритарном классе.

Заключение

Практика была посвящена поиску лучшего классификатора на заданном наборе данных. Рассмотренные методы классификации показали достаточно хорошие значения метрик на первом наборе данных, что может говорить о релевантности признаков, тем самым в дальнейшем может быть проведена идентификация аллелей, ассоциированных с инсультом.

На втором наборе данных полученные метрики имеют намного меньшие значения. Также при обучении классификаторов большой проблемой стало переобучение на мажоритарным классе. Данную проблему не удалось полностью решить ни балансировкой весов, ни уменьшением размерности, ни сэмплированием. Из полученных результатов можно предположить, что второй набор SNP маркеров является неинформативным для определения фактора риска развития ишемического инсульта.

Код, используемый для реализации моделей классификации представлен в Git репозитории.

Список использованных источников

1. Sherry ST, Ward M, Sirotkin K. dbSNP-database for single nucleotide polymorphisms and other classes of minor genetic variation. Genome Res. 1999 Aug; 9(8):677-9. PMID: 10447503.

2. Scikit-learn (Machine Learning in Python) [Электронный ресурс] / Режим доступа: https://scikit-learn.org/stable/modules/classes.html, свободный. (дата обращения: 20.10.21)

3. XGBoost (Gradient boosting library) [Электронный ресурс] / Режим доступа: https://xgboost.readthedocs.io, свободный. (дата обращения: 20.10.21)

4. Imbalanced-learn (MIT-licensed library relying on scikit-learn tools when dealing with classification with imbalanced classes) [Электронный ресурс] / Режим доступа: https://imbalanced-learn.org/stable/references/index.html#api, свободный. (дата обращения: 20.10.21)