МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Машинное обучение»

Тема: «Классификация»

Студент группы 2308 Рымарь М.И.

Преподаватель Турсуков Н.О.

Санкт-Петербург, 2024

ОГЛАВЛЕНИЕ

ТЕОРИЯ

Задача классификации. Задача классификации в машинном обучении — это процесс, в котором алгоритм обучается на наборе данных с целью предсказания категориальной метки (класса) для новых, ранее невидимых данных. Классификация относится к задачам обучения с учителем, где данные предоставляются вместе с соответствующими метками.

Метод k ближайших соседей. k-ближайших соседей (k-Nearest Neighbors, kNN) — это простой и интуитивно понятный алгоритм классификации и регрессии в машинном обучении. Он основан на идее, что объекты, находящиеся близко друг к другу в пространстве признаков, скорее всего, принадлежат к одному и тому же классу.

При классификации нового объекта алгоритм ищет k ближайших соседей среди обучающей выборки. Класс нового объекта определяется по большинству голосов его соседей: класс, который встречается чаще всего среди k ближайших соседей, присваивается новому объекту.

Метод опорных векторов. SVM (Support Vector Machine, метод опорных векторов) — это мощный алгоритм машинного обучения, используемый для классификации и регрессии. Он основан на поиске гиперплоскости, которая наилучшим образом разделяет данные на классы.

В двумерном пространстве гиперплоскость представляется прямой линией, которая разделяет классы. В многомерном пространстве — это более сложная структура.SVM ищет такую гиперплоскость, которая максимизирует расстояние (зазор) между ближайшими точками разных классов, называемыми опорными векторами.

Опорные векторы — это точки данных, которые находятся ближе всего к гиперплоскости. Они определяют положение и ориентацию гиперплоскости.

Решающее дерево. Решающее дерево — это модель машинного обучения, используемая для классификации и регрессии. Оно представляет собой древовидную структуру, где каждый узел соответствует признаку (или атрибуту) данных, а каждое ветвление — условию, основанному на значении этого признака. Листовые узлы дерева содержат конечные решения или предсказания.

ПРЕДОБРАБОТКА ДАННЫХ

Перед обучением модели выполним предобработку данных. Заполним пропуски в данных, удалим ненужные переменные и приведем целевой признак к нужному виду: 0/1 – не болен/болен.

Далее проведем разведочный анализ: посмотрим основные статистики по численным переменным – рисунок 1, построим частотные графики по численным переменным – рисунок 2. Построим также для них графики «ящик с усами» (рисунок 3) и проверим наличие статистических выбросов. Построим матрицу корреляций для численных признаков – рисунок 4.

	count	mean	std	min	25%	50%	75%	max
age	32561.0	38.581647	13.640433	17.0	28.0	37.0	48.0	90.0
fnlwgt	32561.0	189778.366512	105549.977697	12285.0	117827.0	178356.0	237051.0	1484705.0
education.num	32561.0	10.080679	2.572720	1.0	9.0	10.0	12.0	16.0
capital.gain	32561.0	1077.648844	7385.292085	0.0	0.0	0.0	0.0	99999.0
capital.loss	32561.0	87.303830	402.960219	0.0	0.0	0.0	0.0	4356.0
hours.per.week	32561.0	40.437456	12.347429	1.0	40.0	40.0	45.0	99.0
income	32561.0	0.240810	0.427581	0.0	0.0	0.0	0.0	1.0

Рисунок 1 - Статистики по численным переменным.

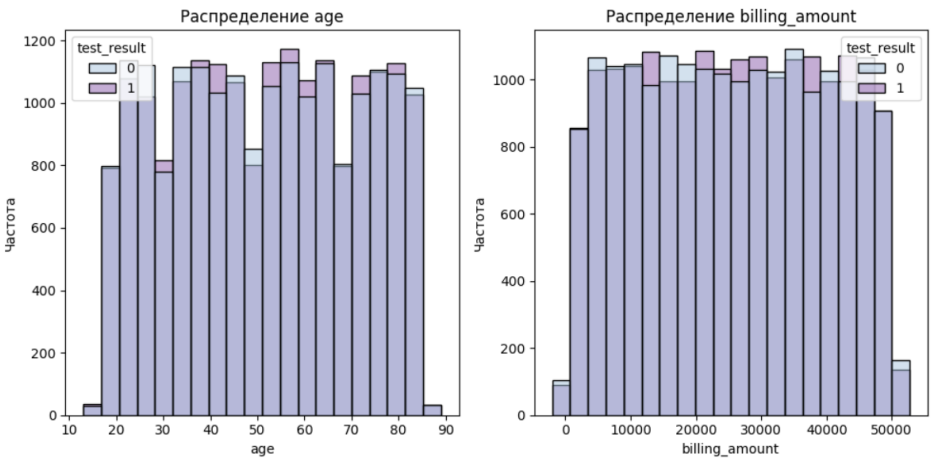

Рисунок 2 - Частотные распределения численных переменных.

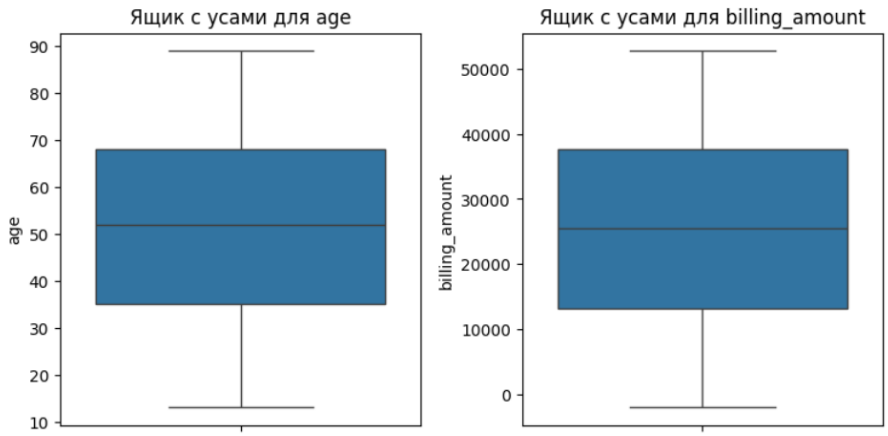

Рисунок 3 – Ящики с усами.

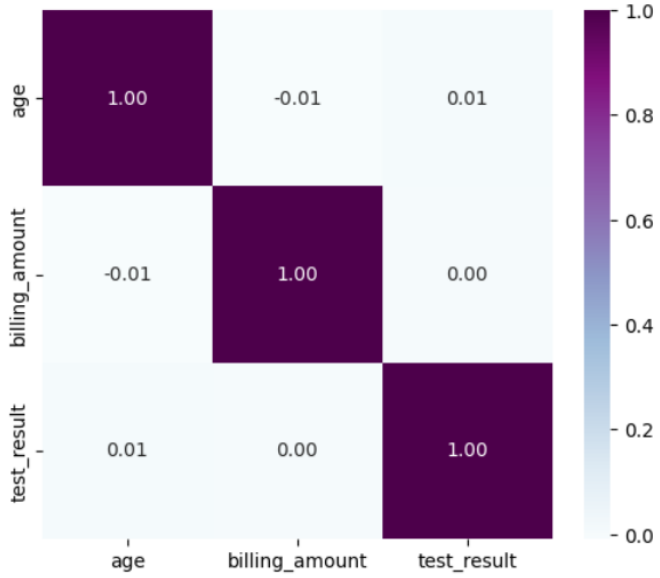

Рисунок 4 – Матрица корреляций.

Далее рассмотрим частотные распределения категориальных переменных и отношения целевых классов внутри них – рисунок 5.

Также проверим баланс классов по всей выборке – рисунок 6.

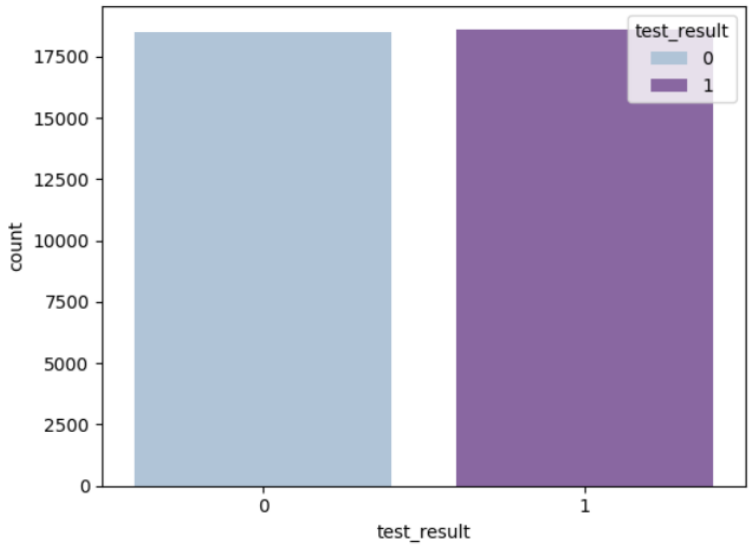

Рисунок 6 – Баланс классов по выборке.

Большинство алгоритмов машинного обучения не умеет работать с категориальными переменными, поэтому необходимо закодировать их. Сделаем это с помощью ручного кодирования для пола, one-hot кодирования для категорий с числом уникальных значений не больше 10 и target-кодированием – для остальных. Теперь все переменные представлены в численном виде – рисунок 7.

	age	gender	date_of_admission	doctor	hospital	billing_amount	room_number	discharge_date	test_result	medication_Aspirin	...	blood_type_A-	blood_type_AB
0	30	1	0.538462	0.214286	0.5	18856.281306	0.600000	0.470588	0	0	...	0	
1	76	0	0.481481	0.000000	0.0	27955.096079	0.452830	0.428571	0	1	...	1	
2	28	0	0.619048	1.000000	1.0	37909.782410	0.549020	0.633333	1	0	...	0	
3	43	0	0.555556	1.000000	1.0	14238.317814	0.513274	0.666667	1	0	...	0	
4	36	1	0.565217	0.000000	0.0	48145.110951	0.413462	0.266667	0	0	...	0	

Рисунок 7 – Закодированные категориальные переменные.

Теперь, когда все переменные представлены в численном виде, можно приступать к обучению моделей.

МОДЕЛЬ KNN

Для обучения модели KNN воспользуемся классом KNeighborsClassifier библиотеки sklearn. С помощью RandomizedSearchCV подберем гиперпараметры модели на кросс-валидации. Обучим лучшую модель. Оценим качество модели на тестовой выборке: выведем метрики Precision, Recall, F1-score, ROC-AUC, а также матрицу ошибок бинарной классификации – рисунок 8.


```
Численные метрики качества:
Precision=0.62, recall=0.63, f1-score_macro=0.62, roc_auc=0.62
```

Рисунок 8 - Метрики обученного классификатора KNN.

МОДЕЛЬ SVM

Для обучения модели SVM воспользуемся классом SVC библиотеки sklearn.

Предварительно необходимо стандартизировать данные для нормальной работы модели. Для этого воспользуемся SimpleScaler из sklearn. Получим стандартизированные данные – рисунок 9.

	age	gender	date_of_admission	doctor	hospital	billing_amount	room_number	discharge_date	medication_Aspirin	medication_Ibuprofen	...	insurance
0	-0.845810	0.993941	-1.294496	-0.366267	1.090498	0.488806	0.186244	0.636699	-0.500120	-0.501682	...	
1	-1.507680	-1.006096	-0.740461	-1.086772	-1.100462	0.396939	3.088219	0.719854	1.999519	-0.501682	...	
2	-0.336679	0.993941	0.308201	1.074741	1.090498	-0.650775	-0.022482	0.351597	-0.500120	-0.501682	...	
3	0.121539	-1.006096	1.950187	1.074741	1.090498	-0.161771	0.156426	-0.523014	-0.500120	-0.501682	...	
4	-0.438505	0.993941	0.581068	1.074741	1.090498	0.678279	-0.639843	-0.016660	-0.500120	-0.501682	...	

Рисунок 9 – Данные после стандартизации.

Подберем гиперпарметры (значение регуляризции) на кросс-валидации с помощью GridSearchCV. Оценим качество модели на тестовой выборке – рисунок 10.


```
Численные метрики качества:
Precision=0.99, recall=0.99, f1-score_macro=0.99, roc_auc=0.99
```

Рисунок 10 - Метрики обученного классификатора SVM.

МОДЕЛЬ РЕШАЮЩЕЕ ДЕРЕВО

Для обучения модели решающего дерева воспользуемся классом DecisionTreeClassifier библиотеки sklearn.

С помощью GridSearch подберем гиперпараметры модели (глубину дерева, критерий информативности, минимальное число объектов для сплита и листа) на кросс-валидации. Обучим лучшую модель. Оценим качество модели на тестовой выборке – рисунок 11.


```
Численные метрики качества:
Precision=0.99, recall=0.99, f1-score_macro=0.99, roc_auc=0.99
```

Рисунок 11 - Метрики обученного классификатора Random Forest.

Визуализируем полученное дерево решений – рисунок 12.

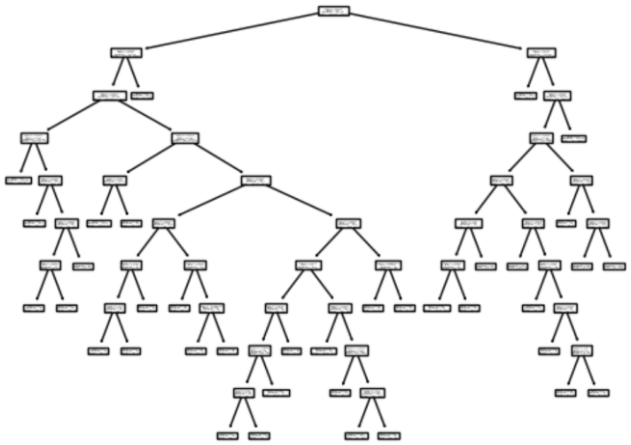

Рисунок 12 – Обученное дерево решений.

Визуализируем feature importance – важность переменных для предсказания дерева. Feature importance представлен на рисунке 13.

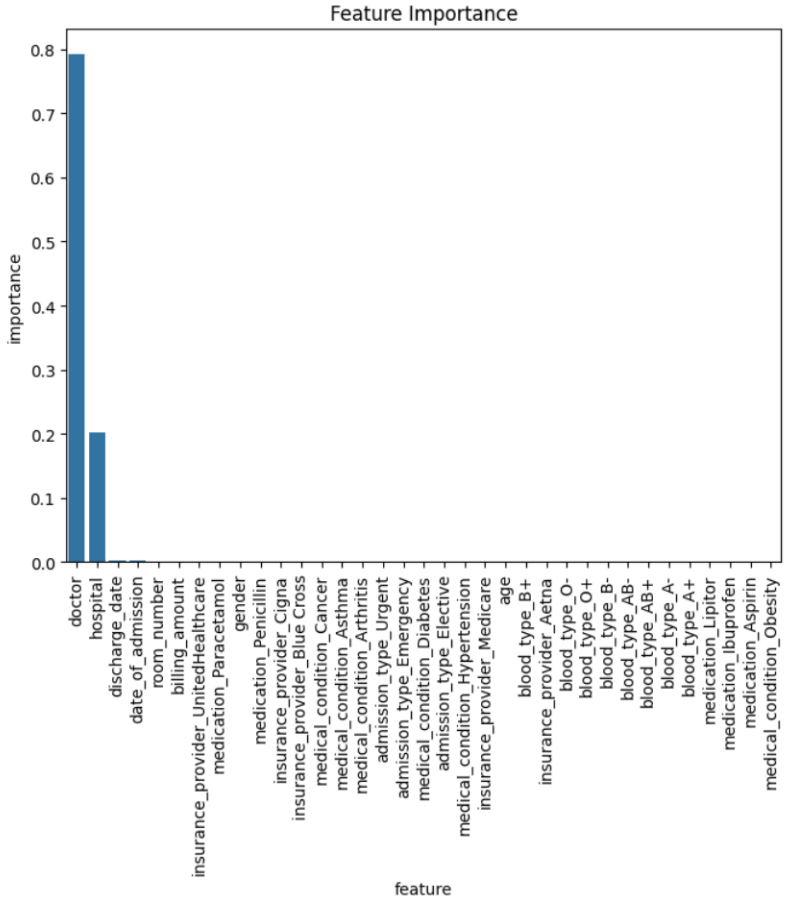

Рисунок 13 - Важность переменных для предсказания дерева.

Оказалось, что самыми важными признаками для предсказания модели оказались доктор и больница.

ВЫВОД

В ходе лабораторной работы была проведена оценка трех популярных алгоритмов классификации: K ближайших соседей (KNN), метода опорных векторов (SVM) и решающего дерева (Decision Tree). Каждый из методов был применен к одному и тому же набору данных с целью выявления их эффективности и сравнительных характеристик.

Качество классификаторов можно сравнить, например, по метрике F1-score. Самый слабый результат показала модель KNN. Метод опорных векторов и решающее дерево дали очень хороший результат.