Отчет по заданию в Kaggle

Гончарова Наталия, группа БПМ-17-2 Декабрь 2020 г.

1 Постановка задачи

Решение задачи с Kaggle с использованием нейронной сети. Ссылка на задачу - https://www.kaggle.com/nasa/kepler-exoplanet-search-results

2 Входные переменные

Ознакомимся со структурой файла kepler-exoplanet-search-results.csv

	rowid	kepid	kepoi_name	kepler_name	koi_disposition	koi_pdisposition	koi_score	koi_fpflag_nt	koi_fpflag_ss	koi_fpflag_co	 koi_steff_err2	koi_
0	1	10797460	K00752.01	Kepler-227 b	CONFIRMED	CANDIDATE	1.000	0	0	0	 -81.0	
1	2	10797460	K00752.02	Kepler-227 c	CONFIRMED	CANDIDATE	0.969	0	0	0	 -81.0	
2	3	10811496	K00753.01	NaN	FALSE POSITIVE	FALSE POSITIVE	0.000	0	1	0	 -176.0	
3	4	10848459	K00754.01	NaN	FALSE POSITIVE	FALSE POSITIVE	0.000	0	1	0	 -174.0	
4	5	10854555	K00755.01	Kepler-664 b	CONFIRMED	CANDIDATE	1.000	0	0	0	 -211.0	

Рис. 1: Структура файла

3 Решение

Посмотрим на распределение параметров koi-pdisposition и koi-disposi.



Рис. 2: Распределение параметров

Перекодируем категориальные признаки для дальнейшего анализа. Реализуем функцию, которая принимает на вход DataFrame, кодирует числовыми значениями категориальные признаки и возвращает обновленный DataFrame и сами кодировщики.

r	owid	kepid	kepoi_name	koi_disposition	koi_pdisposition	koi_score	koi_fpflag_nt	koi_fpflag_ss	koi_fpflag_co	koi_fpflag_ec	 koi_steff_err2	koi_
0	1	10797460	1080	1	0	1.000	0	0	0	0	 -81.0	
1	2	10797460	1081	1	0	0.969	0	0	0	0	 -81.0	
2	3	10811496	1082	2	1	0.000	0	1	0	0	 -176.0	
3	4	10848459	1083	2	1	0.000	0	1	0	0	 -174.0	
4	5	10854555	1084	1	0	1.000	0	0	0	0	 -211.0	

Рис. 3: Категориальные признаки

Построим матрицу корреляций:

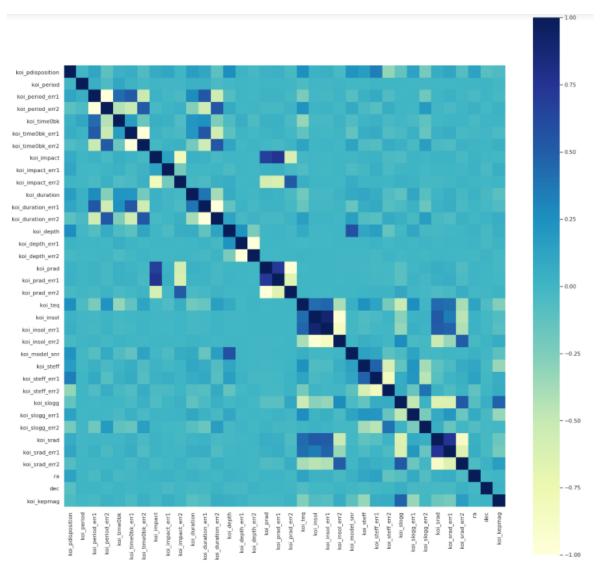


Рис. 4: Матрица корреляций

Посмотрим на распределение величин по признакам в данных:

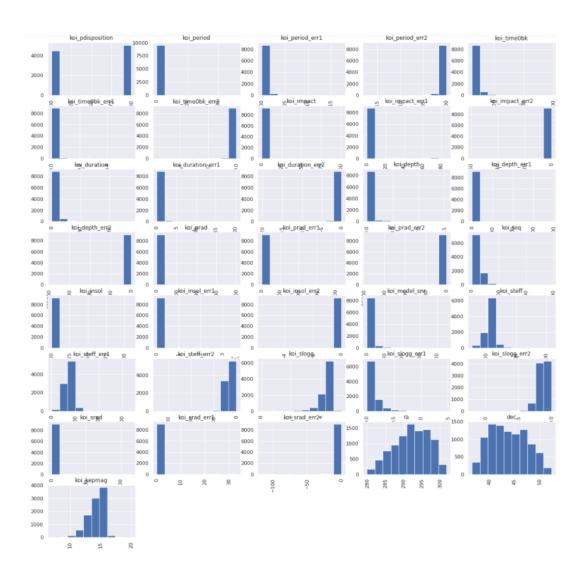


Рис. 5: Распределение величин по признакам

Для заполнения пропущенных данных используем Imputation. Imputation заполняет недостающее значение некоторым числом.

	0	1	2	3	4	5	6	7	8	9	 26	27	28	29	30	31	32	3
0	0.0	9.488036	2.775000e- 05	-2.775000e- 05	170.538750	0.002160	-0.002160	0.146	0.318	-0.146	 -81.0	4.467	0.064	-0.096	0.927	0.105	-0.061	291.9342
1	0.0	54.418383	2.479000e- 04	-2.479000e- 04	162.513840	0.003520	-0.003520	0.586	0.059	-0.443	 -81.0	4.467	0.064	-0.096	0.927	0.105	-0.061	291.9342
2	1.0	19.899140	1.494000e- 05	-1.494000e- 05	175.850252	0.000581	-0.000581	0.969	5.126	-0.077	 -176.0	4.544	0.044	-0.176	0.868	0.233	-0.078	297.0048
3	1.0	1.736952	2.630000e- 07	-2.630000e- 07	170.307565	0.000115	-0.000115	1.276	0.115	-0.092	 -174.0	4.564	0.053	-0.168	0.791	0.201	-0.067	285.5346
4	0.0	2.525592	3.761000e- 06	-3.761000e- 06	171.595550	0.001130	-0.001130	0.701	0.235	-0.478	 -211.0	4.438	0.070	-0.210	1.046	0.334	-0.133	288.7548

Создадим обучающую выборку и разделим ее на тестовую 0.15 и обучающую 0.85.

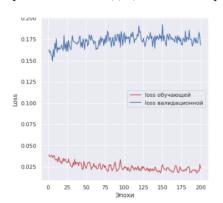
В качестве библиотеки для создания нейронных сетей используется keras.

Задаются следующие характеристики:

- 1. на первом слое содержится 128 нейронов с relu функцией активации;
- 2. на втором слое содержится 64 нейронов с relu функцией активации;
- 3. на третьем слое содержится 64 нейрона с relu функцией активации;
- 4. на третьем слое содержится 64 нейрона с relu функцией активации;
- 5. на выходное слое содержится 1 нейрон с сигмоидной функцией;

В качестве метода оптимизации был выбран adam. Было проведено обучение на 100 эпохах с 32 батчами

Посмотрим на графики изменения ошибки и точности по эпохе тренировочной и валидационных выборках.



0.950
0.925
0.925
0.875
0.850
0.825
0.800
0.25 50 75 100 125 150 175 200

Рис. 6: Графики изменения ошиб-ки

Рис. 7: Графики изменения точност.

Получим пример предсказания:

	precision	recall	f1-score	support
0	0.84	0.80	0.82	679
1	0.81	0.84	0.82	670
accuracy			0.82	1349
macro avg	0.82	0.82	0.82	1349
weighted avg	0.82	0.82	0.82	1349

Рис. 8: Пример предсказания

4 Вывод

В результате данной работы была написана и обучена нейронная сеть с использованием библиотеки keras для предсказывания, а также построены графики обучения.