Светашева Юлия ИУ5-64Б

17 вариант РК-2

Задание. Для заданного набора данных (по Вашему варианту) постройте модели классификации или регрессии (в зависимости от конкретной задачи, рассматриваемой в наборе данных). Для построения моделей используйте методы 1 и 2 (по варианту для Вашей группы). Оцените качество моделей на основе подходящих метрик качества (не менее двух метрик). Какие метрики качества Вы использовали и почему? Какие выводы Вы можете сделать о качестве построенных моделей? Для построения моделей необходимо выполнить требуемую предобработку данных: заполнение пропусков, кодирование категориальных признаков, и т.д.

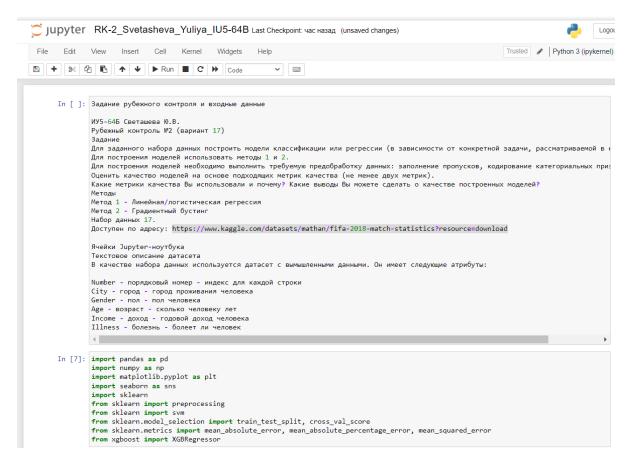
• Для студентов групп ИУ5-61Б, ИУ5-62Б, ИУ5-63Б, ИУ5-64Б, ИУ5-65Б, РТ5-61Б номер варианта = номер в списке группы.

 Группа
 Метод №1
 Метод №2

 ИУ5-64Б, ИУ5Ц-84Б
 Линейная/логистическая регрессия
 Градиентный бустинг

Используемый набор данных: Predict FIFA 2018 Man of the Match | Kaggle

Результат:



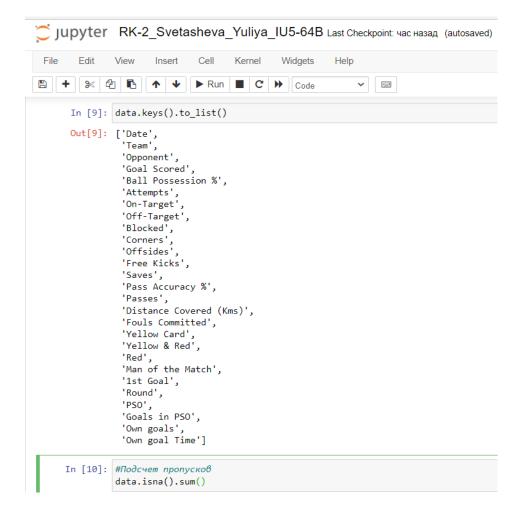


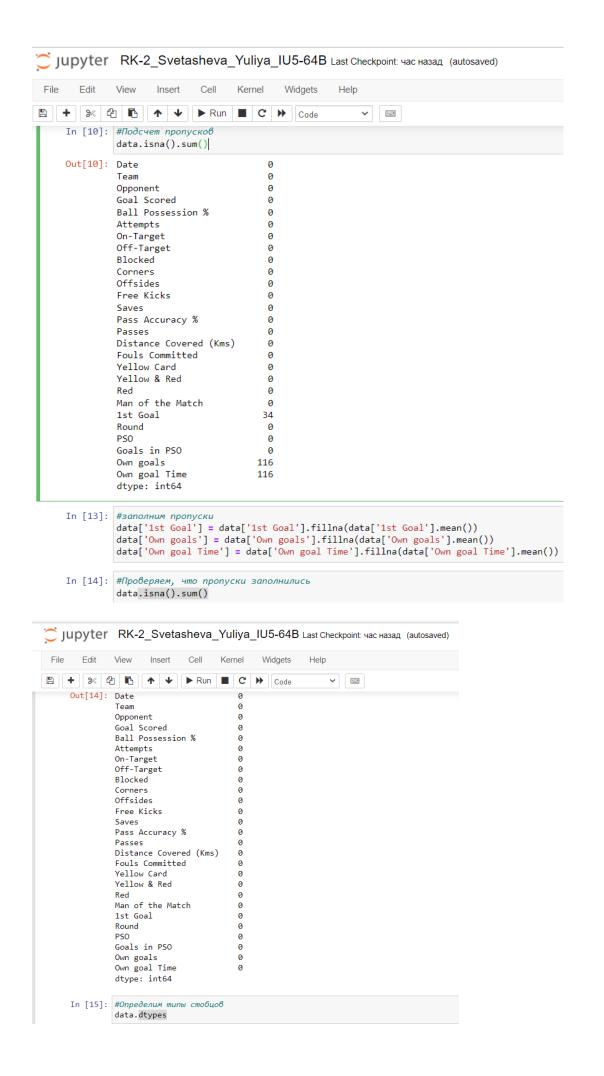


Out[8]:

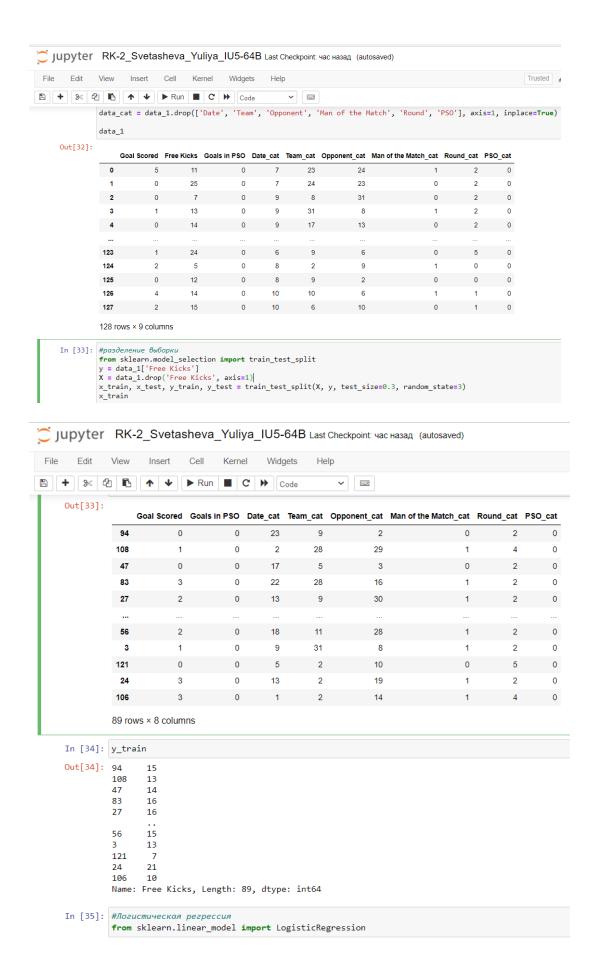
	Date	Team	Opponent	Goal Scored	Ball Possession %	Attempts	On- Target	Off- Target	Blocked	Corners	 Yellow Card	Yellow & Red	Red	Man of the Match	1st Goal	Round	PSO	Goals in PSO	٤
	14- 0 06- 2018	Russia	Saudi Arabia	5	40	13	7	3	3	6	 0	0	0	Yes	12.0	Group Stage	No	0	
	14- 1 06- 2018	Saudi Arabia	Russia	0	60	6	0	3	3	2	 0	0	0	No	NaN	Group Stage	No	0	
	15- 2 06- 2018	Egypt	Uruguay	0	43	8	3	3	2	0	 2	0	0	No	NaN	Group Stage	No	0	
	15- 3 06- 2018	Uruguay	Egypt	1	57	14	4	6	4	5	 0	0	0	Yes	89.0	Group Stage	No	0	
	15- 4 06- 2018	Morocco	Iran	0	64	13	3	6	4	5	 1	0	0	No	NaN	Group Stage	No	0	
12	11- 3 07- 2018	England	Croatia	1	46	11	1	6	4	4	 1	0	0	No	5.0	Semi- Finals	No	0	
12	14- 4 07- 2018	Belgium	England	2	43	12	4	3	5	4	 1	0	0	Yes	4.0	3rd Place	No	0	
12	14- 5 07- 2018	England	Belgium	0	57	15	5	7	3	5	 2	0	0	No	NaN	3rd Place	No	0	
12	15- 07- 2018	France	Croatia	4	39	8	6	1	1	2	 2	0	0	Yes	18.0	Final	No	0	
12	15- 7 07- 2018	Croatia	France	2	61	15	3	8	4	6	 1	0	0	No	28.0	Final	No	0	

128 rows × 27 columns









```
Jupyter RK-2 Svetasheva Yuliya IU5-64B Last Checkpoint: час назад (autosaved)
File
       Edit
              View
                     Insert
                             Cell
                                    Kernel
                                           Widgets
                                                       Help
                                                                                                                    Trusted
A code
A code
A code
A code
                                                           ~
     In [47]: def print_metrics(y_test, y_pred):
                  print(f"R^2: {r2_score(y_test, y_pred)}")
                  print(f"MSE: {mean_squared_error(y_test, y_pred)}")
                  print(f"MAE: {mean_absolute_error(y_test, y_pred)}")
     In [41]: import warnings
              warnings.filterwarnings('ignore')
              model_logistic = LogisticRegression()
              model_logistic.fit(x_train, y_train)
    Out[41]: LogisticRegression()
     In [43]: targ logistic = model logistic.predict(x test)
     In [44]: | mae = mean_absolute_error(y_test,targ_logistic)
              mape = mean_absolute_percentage_error(y_test,targ_logistic)
              mse = mean_squared_error(y_test,targ_logistic)
              print('MAE:' + str(round(mae,3)) + ' MAPE:' + str(round(mape,3)) + ' MSE:' + str(round(mse,3)))
              MAE:5.564 MAPE:0.43 MSE:45.513
     In [49]: #Градиентный брустинг
              XGB_model = XGBRegressor()
              mape = -cross_val_score(XGB_model,x_train,y_train,cv=4,scoring='neg_mean_absolute_percentage_error').mean()
              mae = -cross_val_score(XGB_model,x_train,y_train,cv=4,scoring='neg_mean_absolute_error').mean()
              mse = -cross_val_score(XGB_model,x_train,y_train,cv=4,scoring='neg_mean_squared_error').mean()
              print('SVM Errors')
              print('MAE:' + str(round(mae,3)) + ' MAPE:' + str(round(mape,3)) + ' MSE:' + str(round(mse,3)))
              SVM Errors
              MAE:4.361 MAPE:0.355 MSE:30.424
     In [50]: XGB_model.fit(x_train,y_train)
              mae = mean_absolute_error(y_test,XGB_model.predict(x_test))
              mape = mean_absolute_percentage_error(y_test,XGB_model.predict(x_test))
              mse = mean_squared_error(y_test,XGB_model.predict(x_test))
              print('MAE:' + str(round(mae,3)) + ' MAPE:' + str(round(mape,3)) + ' MSE:' + str(round(mse,3)))
              MAE:5.732 MAPE:0.416 MSE:45.084
 In [62]: #Сравнение моделей
```

Видим, что модель Градиентный брустинг показал себя лучше, чем логистическая регрессия