Софийски университет „Св. Кл. Охридски”

Факултет по математика и информатика

Курсов Проект

на тема: „Прогнозиране успеха на онлайн дарителски кампании”

Студент: Симеон Емилов Христов, Ф.Н. **6MI3400191**

Курс: „първи“, Учебна година: 2022/23

Преподаватели: **проф. Иван Койчев, ас. Борис Величков**

==========

Декларация за липса плагиатство:

- Плагиатство е да използваш, идеи, мнение или работа на друг, като претендираш, че са твои. Това е форма на преписване.
- Тази курсова работа е моя, като всички изречения, илюстрации и програми от други хора са изрично цитирани.
- Тази курсова работа или нейна версия не са представени в друг университет или друга учебна институция.
- Разбирам, че ако се установи плагиатство в работата ми ще получа оценка “Слаб”.

27.6.23 г. Подпис на студента:

Съдържание

1 Увод

Онлайн дарителските кампании стават все по-популярен начин за създаването на (допълнителни) доходи за предприемачи и творци. По-точното прогнозиране на успеха на даден проект още в самото му начало може да помогне както на страната, която предлага продукта, така и на страната, която го финансира, т.е. ще доведе до по-добро разпределяне на ресурси.

Този курсов проект разглежда начини за създаване на различни модели, които предсказват дали (и/или доколко) онлайн дарителска кампания ще успее да събере нужните на производителя ресурси. За решаването на тази задача се преминава също и през задачата за предвиждане на броят хора, които ще дарят средства в рамките на кампанията.

2 Преглед на областта за онлайн дарителски кампании

Проучванията показват, че за прогнозиране на успеха с голяма точност трябва да се използват както характеристики на самата кампания (колко хора са дарители, какво количество пари е необходимо и др), така и анализи на текста, който описва кампанията.

Въпреки това в [2] Винсент и др показват, че дори и само с информацията за изискащите се средства (количество, валута) и брой хора, които даряват може да се постигне точност около 97%. В допълнение на това те анализират и текстовете на постове, свързани с кампанията, в социалната мрежа Twitter. Те използват SVM, който обединява резултатите от алгоритмите KNN и Наивен Бейсов класификатор.

В [4] Шоуни и др показват, че времето от началото на кампанията до нейния край също оказва влияние и може да корелира с вероятността за успех. Отделно не също се фокусират върху обработката на текстовите характеристики и чрез Наивен Бейсов класификатор обучен само върху униграми, успяват да постигнат точност от 65%. Експериментите им показват, че ако вместо това се обучи SVM на същите тези униграми, но се добави и информация за броят дарители, то точността се вдига до 92%.

Чен и др също анализират чрез SVM характеристики, които идват от кампаниите, които създателите на дарителските сметки създават в различни социални мрежи – Youtube и Twitter. Те обаче заключват, че такива външни данни нямат по-голяма предвиждаща сила, т.е. не са по-добри за използване в модели от характеристиките на самите данни [3]. Тяхните експерименти показват, че най-много информация се постига от средствата, които се изискват, и по-скоро промяната на тези средства с течение на времето (тук се приема, че създалият кампанията може да променя парите, които цели да събере). С други

думи тази задача се променя от класификационен проблем в проблем, свързан с анализиране на различни видове времеви редове.

Горните статии разглеждат главно линейни класификационни модели. В [1] Ченчен и др разглеждат как проблемът може да бъде решен чрез дълбоки невронни мрежи. Те създават дълбока рекурентна невронна мрежа базирана на архитектурата LSTM и чрез нея постигат точност от 72%. Трябва обаче да се отбележи, че докладваната точност е постигната върху дисбалансирани данни. В частност вероятността за провал на кампанията, в данните, които те използват е 68%, т.ч. постигане на голяма точност не е толкова неочаквано. Все пак тази статия се отличава от останалите с неконвенционалния си подход.

3 Проектиране

Използваните данни са същите, които са използвани в [1][5]. Това предразполага към изграждане на модел, които да може да сравни с други вече съществуващи решения. Данните са структурирани и предварително разделени на данни за трениране на модел и на такива за неговото тестване. Моделът на данните е следният:

- project_id – уникален идентификатор за кампанията, поставен от Kickstarter в нейното начало.
- name – име на кампанията.
- desc – описание на кампанията, поставено от нейния създател.
- goal – количество средства, които трябва да се съберат (пари).
- keywords – ключови думи за кампанията.
- disable_communication – булев флаг, който посочва дали създателят би искал да използва функцията чат с неговите дарители. Ако стойността му е „лъжа“, то няма комуникация между двете страни.
- country – държава, в която се провежда кампанията. Забележка: човек може да дари на кампанията дори и да е от друга държава.
- currency – валута, в която се изискват паричните средства.
- deadline – краен срок за събиране на изисканите парични средства.
- state_changed_at – дата и час на промяна на поне една характеристика от страна на създателя на кампанията.
- created_at - дата и час на създаване на кампанията.
- launched_at - дата и час на създаване на започване на кампанията. Забележка: възможно е да се създаде кампания за напред във времето, т.е. дарители няма да могат да се присъединят докато не настъпи даден ден и час. Времето между датата на създаване и датата на започване може да се използва за реклама на (целите на) кампанията.

- backers_count – брой дарители. Забележка: това поле би имало стойност 0 при самото създаване. С течение на времето то се променя и именно тези промени са анализирани от статиите и имат добра предсказваща сила. В използваните данните то заема стойността на броя дарители в момента на приключване на кампанията.
- final_status – стойност 0 или 1. Стойностите означават съответно дали кампание не е била успешна и дали е била успешна.

Целевата характеристика е final_status.

4 Реализация, тестване/експерименти

4.1 Използвани технологии, платформи и библиотеки

Проектът е реализиран на Python с множество интерактивни тетрадки (ipynb файлове), както и скриптове. Като допълнителни библиотеки се използват:

- numpy, pandas, matplotlib;
- inflect – за преобразуване на числата към думи;
- tqdm – за визуализация на прогреса при обхождания;
- tensorflow - за токенизиране на текстовите характеристики;
- nltk – за работа с текстовите характеристики;
- pycaret – за автоматично изграждане и тестване на голямо количество модели за машинно самообучение;
- pytorch – за работа с невронни мрежи.

За реализация, трениране и валидиране на различни невронни мрежи с помощта на графични карти бе използвана среда, предоставена от Kaggle.

4.2 Реализация и провеждане на експерименти

За анализ на входните данни бе разработен клас DataAudit, който изчислява различни статистически характеристики за всеки атрибут, включително взаймна информация, хи-квадрат p-value, и kruskal-wallis p-value.

Понеже стойностите на колоната backers_count се променят във времето и в частност са 0 в началото на кампанията, в тестовата разбивка на данните липсват колоните backers_count и final_status. Провеждите експерименти и проучвания показват, че именно backers_count често има най-голямо влияние за правилно предвиждане на final_status. С други думи не е желателно да се създаде модел, който не е бил обучаван със стойностите на backers_count в тренировъчната разбивка. Това довежда до нуждата от създаването на два модела - регресионен и класификационен. Първият от тях предвижда очаквания

броя дарителите в края на кампанията и след това чрез втория тази информация да бъде взета предвид при изчисляването на вероятността за успех.

Използваната архитектура за новопостъпили данни е показана на Фигура 1:

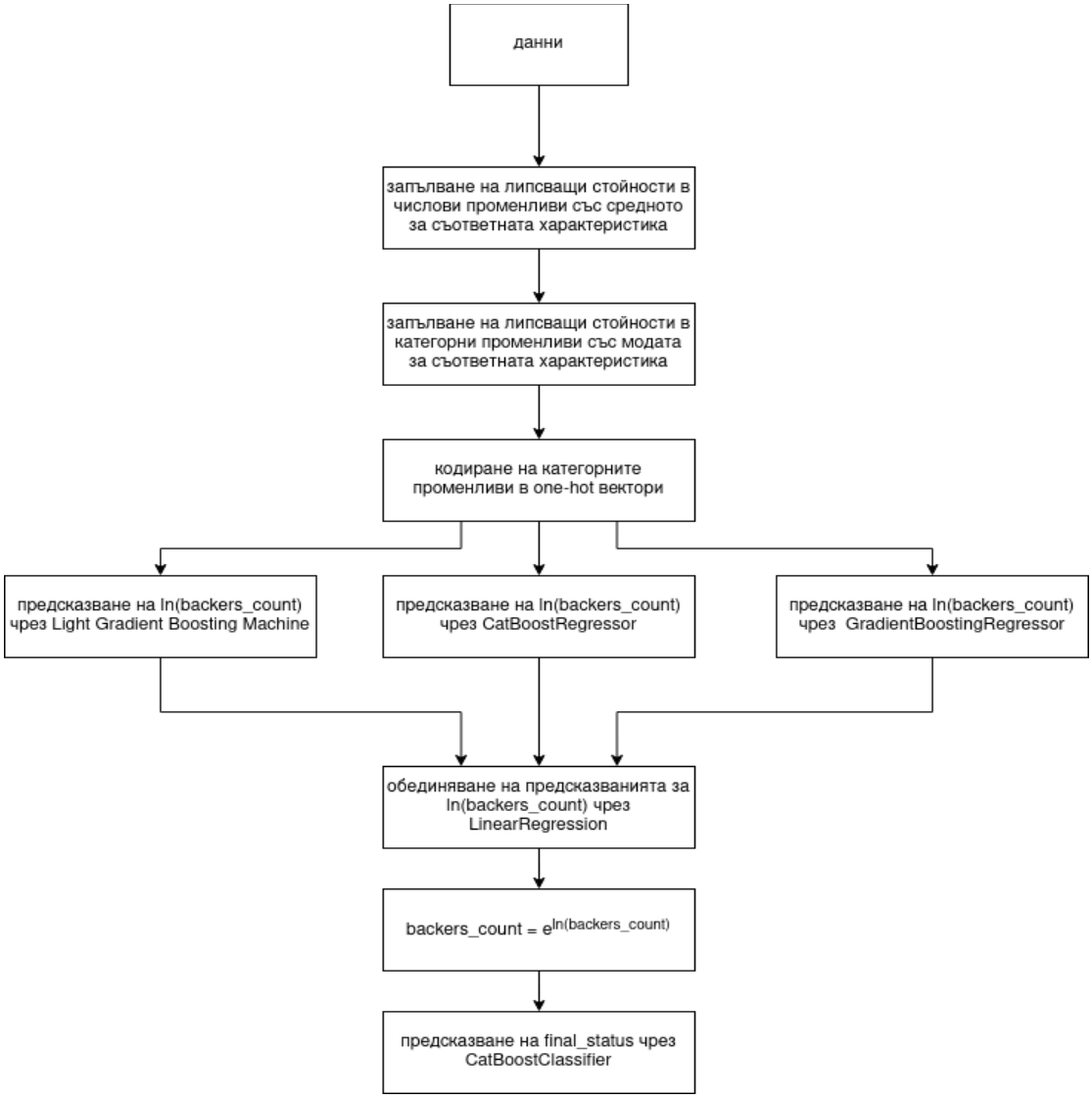

Figure 1: Архитектура на финално приложение

За трениране на регресионните модели бяха използвани колоните: goal, disable_communication, country, currency, create_launch_hours, create_launch_hours_log, create_deadline_hours, create_deadline_hours_log, и launched_deadline_hours. Последните 5 колони бяха създадени допълнително от налични данните за датите. Постигнатите резултати на най-добрия регресионен модел са обобщени в таблица 1.

Table 1: Постигнати резултати от StackingRegressor

Име на модел	**MAE**	**MSE**	**RMSE**	**R2**	**RMSLE**	**MAPE**	**R2 Adjusted**
StackingRegressor	1.3903	2.9069	1.7050	0.1839	0.5567	0.6087	0.1839

Постигнатите резултати на най-добрия класификационен модел са обобщени в таблица 2.

Table 2: Постигнати резултати от CatBoostClassifier

Име на модел	AUC	Обхват	Прецизност	F1-оценка	Коефициент на Матю
CatBoostClassifier	0.95	0.85	0.78	0.81	0.72

Полученият регресионен модел не се представя добре. Това се дължи главно на малкия брой характеристики и това, че са нямат добра предвиждаща сила. За решаване на този проблем могат да се потърсят нови характеристики (например от текстовите колони).

С цел сравняване с класификационните резултати с тези постигнати в [1] бе изчислена и точността на модела: 87,63%. Тя подобрява тази, постигната в [1] – 72,42%, с 15,21%, като при това не използва дълбоко машинно самообучение.

Експерименти бяха направени с различни видове плитки невронни мрежи. Целта бе да се провери дали може да се научи такова вграждане, което да помогне за подобряване на класификационната точност. Използваната архитектура е представена на картина 2.

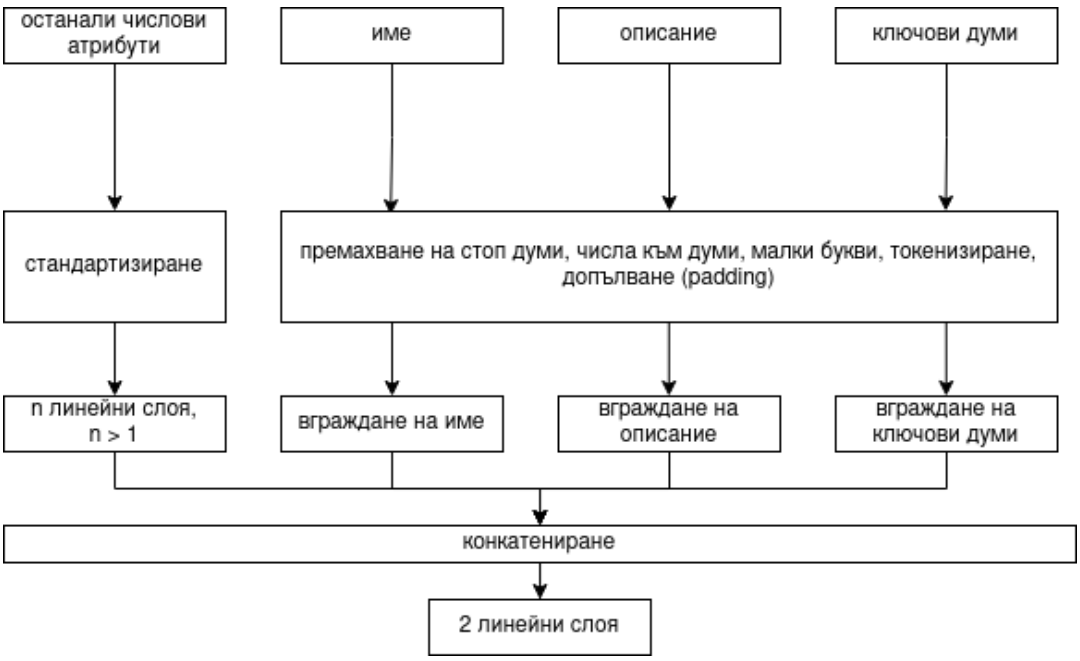

Figure 2: Архитектура на плитка невронна мрежа

Бяха направени експерименти с различени хиперпараметри, регулиращи пространството на вграждане и размерът на скритите линейни слоеве. Резултатите не са добри – 50% F1-оценка. Една причина за това е неизползването на рекурентна невронна мрежа. Важно предимство на този вид невронни мрежи е тяхната памет и използването им при текстова обработка би спомогнало за подобряване на резултатите. От друга страна, както в [1] посочват вместо да се научават влаганията на думите, могат да се използват наготово научените в GloVe.

5 Заключение

Курсовият проект демонстрира как може да се използват различни номинални, числови и текстови характеристики, за да се предвиди успехът на онлайн дарителска кампания. За целта бяха решени две задачи – предвиждане на броят на дадена кампания и, взимайки това под внимание, предвиждане на вероятността за успех на кампания още от нейното създаване. Резултатите на финалния модел са обобщени в следващата таблица.

Table 3: Постигнати резултати от CatBoostClassifier

Име на модел	AUC	Обхват	Прецизност	F1-оценка	Коефициент на Матю
CatBoostClassifier	0.95	0.85	0.78	0.81	0.72

С цел сравняване с резултати от предишни разработки по същите данни бе изчислена и точността на модела: 87,63%, която подобрява тази, постигната в [1] – 72,42%, с 15,21%, като при това не използва дълбоко машинно самообучение. Метриката не е отбелязана в таблицата, т.к. при дисбалансирани класове не е подходящо да се използва.

При по-нататъшно развитие могат да бъдат направени експерименти с обработката на текста. Вместо научаване на вграждания на думи, могат да се използват готовите такива от GloVe. Друг възможен подход за преобразуването на думите в числови вектори е tf-idf. В тази връзка е възможна и по-добра подготовка на текстовите данни. Възможно е и използването на архитектурата Transformer като по-комплексна невронна мрежа.

За подобряване на резултатите на регресионния модел могат да се потърсят и нови данни с повече характеристики. Като последствие от подобрението би следвало да се наблюдава и по-надеждна класификация.

6 Използвана литература

[1] Chenchen Pan, et al. Predicting The Success of Crowdfunding 2017, https://cs230.stanford.edu/projects_spring_2018/reports/8289614.pdf.

[2] Etter, Vincent, et al. Launch Hard or Go Home! 2013, https://doi.org/10.1145/2512938.2512957.

[3] Kevin Chen, et al. KickPredict: Predicting Kickstarter Success 2015, https://nlp.stanford.edu/courses/cs224n/2015/reports/15.pdf.

[4] Kartik Sawhney, et al. Using Language to Predict Kickstarter Success, https://stanford.edu/~kartiks2/kickstarter.pdf.

[5] Kickstarter data 2018, https://www.kaggle.com/datasets/iamsajanbhagat/kickstarter.