Photometric LSST astronomical time-series classification challenge - PLAsTiCC -

Благој Христов
Факултет за електротехника и информациски технологии,
Универзитет "Св. Кирил и Методиј" во Скопје
email: blhris@gmail.com
Предмет: Роботика 2

***Апстракт*—Со развојот на технологијата и науката се развиваат и модерни, поефикасни и поточни методи за детекција и класификација на астрофизичките појави во универзумот. Еден ваков револуционерен подвиг е т.н. *Large Synoptic Survey Telescope (LSST),* со кој ќе се примени сосем нов начин за набљудување на ноќното небо. Масивното количество на податоци кое што ќе го обезбеди *LSST* невозможно е да се анализира со застарените стандардни методи, па затоа е создаден натпреварот *PLAsTiCC* чија цел е да се создаде алгоритам со користење на машинско учење, кој ќе помогне во класификацијата на детектираните објекти. Во овој труд се користат пет различни класификатори за решавање на овој проблем, а резултатите покажаа дека поради комплексноста на проблемот поуспешна класификација се добива со користење на ансамбл *(ensemble)* методите. Најдобри резултати се добија со користење на вештачка невронска мрежа.**

Клучни зборови – класификација на астрономски објекти; машинско учење; LSST; PLAsTiCC

I. Вовед

PLAsTiCC е натпревар на *Kaggle* чија цел е класификација на временска серија од астрономски податоци. Тие претставуваат симулација на вистински измерени вредности кои што ќе се добиваат со новиот *Large Synoptic Survey Telescope (LSST)*, кој ќе биде пуштен во работа во 2020 година. Телескопот ќе користи нов начин на мерење на светлина од објектите, со користење на шест различни филтри или *passbands*, со цел да добие светлински криви во различни делови од спектарот. Филтрите кои ќе се користат мерат светлина во ултравиолетовиот, видливиот и инфрацрвениот дел од светлинскиот спектар. На овој начин може со краткотрајно мерење да се открие каков тип на светлина зрачи објектот, нејзиниот интензитет, дали е од периодична или еднократна природа и други информации кои што ќе помогнат за одредување на неговата природа. Овој начин на мерење ќе овозможи огромно подобрување на количеството на податоци кои што можат да се измерат, како и брзината на нивното прибирање.

II. Опис на проблемот

За разлика од стариот начин со долготрајно мерење на истиот објект со цел да се направи негова спектроскопска анализа (за кој е потребно многу долго време), *LSST* ќе врши повеќекратно мапирање на темното небо, секоја ноќ со различен *passband*. Ова ќе овозможи да се приберат податоци со големина од 20 – 40 терабајти, секој ден. Поради масивноста на податочното множество потребен е моќен и прецизен алгоритам кој ќе се користи за класификацијата на овие податоци. Целта на натпреварот е да се создаде таков алгоритам кој ќе има улога да го одреди типот на астрономскиот објект кој се набљудува со што е можно поголема сигурност и точност, како и за откривање на нови досега невидени објекти.

Слика 1. Large Synoptic Survey Telescope (LSST)

III. Податоци

Податоците кои што се достапни за овој труд претставуваат само симулација на реалните податоци кои што ќе се добиваат од *LSST,* бидејќи тој се уште не е пуштен во работа. На располагање постојат четири множества на податоци:

- ***training_set***: Податоци кои содржат повеќекратни мерени вредности на светлинскиот флукс на објектите низ различните *passbands* и времето кога е направено мерењето, дадено во Модифициран Јулијански датум. Секоја редица претставува едно мерење на флуксот на еден објект.

- ***training_set_metadata***: Метаподатоци за секој од објектите кое содржи информација за локацијата на објектот, дали објектот има "црвено поместување" во спектарот, дали е во нашата галаксија или надворешен објект итн. Ова множество содржи и колона која одредува на која класа припаѓа објектот. Секоја редица претставува информации за еден објект.
- ***test_set***: Содржи исти податоци како *training_set,* но за нови објекти кои не се класифицирани. Ова множество е огромно, со големина од *19.3GB* и не се користи во овој труд. Точноста на алгоритмите се одредува со одделување на посебно множество за тестирање (во сооднос 1:10) од множеството за тренинг.
- ***test_set_metadata***: Исто како *test_set*, содржи метаподатоци за новите објекти, па и ова множество не се користи.

Сите податоци се претставени во табеларна форма каде првата колона го одредува идентификацискиот број на објектот, а останатите колони се неговите карактеристики или *features*. Во двете множества кои што се користат постојат различни карактеристики кои даваат различни информации за објектот. Во множеството *training_set* постојат следните *features*:

- ***object_id:*** Идентификациски број на објектот
- ***mjd:*** Единица за време, претставена со Модифициран Јулијански датум во денови. Тој се добива со одземање на *2400000.5* дена од Јулијанскиот датум, и најчесто се користи во астрономијата.
- ***passband:*** Кој филтер се користи при извршување на мерењето. Оваа карактеристика е категорична и содржи вредности од 1 до 6 за секој од поединечните филтри *(u, g, r, i, z, Y)*.
- ***flux:*** Вредноста на измерениот светлински флукс на објектот.
- ***flux_err:*** Апроксимација на грешката која постои во мерењето на флуксот.
- ***detected:*** Бинарна карактеристика која укажува дали е "детектиран" објектот, односно дали интензитетот на светлината која ја оддава е поголем од некоја претходно поставена граница. Вредноста на границата се одредува од осветленоста на самото небо, која треба да биде 3 стандардни девијации пониска од осветленоста на објектот за да се смета како детектиран.

Бидејќи постојат шест различни филтри, и за еден објект се направени повеќе мерења на флуксот низ секој од филтрите, ова множество содржи *1.42 милиони* редици.

Во множеството *training_set_metadata* постојат следните *features*:

- ***object_id:*** Идентификациски број на објектот
- ***ra:*** Ректасцензија, координата со која се дефинира положбата на некој објект на небото во екваторскиот координатен систем. Еквивалентна на географска должина на Земјата.
- ***decl:*** Деклинација, еквивалентна на географска ширина на Земјата и ја претставува оддалеченоста на небесното тело од небесниот екватор.
- ***gal_l:*** Галактичка лонгитуда (должина) претставена во степени, во галактичкиот координатен систем во чиј центар се наоѓа Сонцето.
- ***gal_b:*** Галактичка латитуда (ширина) претставена во степени.
- ***ddf:*** Бинарна карактеристика која укажува дали објектот се наоѓа во *DDF (deep drilling fields)* областа на истражување.
- ***hostgal_specz:*** Спектроскопско "црвено поместување" или *redshift* на светлинскиот извор. Оваа карактеристика претставува исклучително прецизно мерење на црвеното поместување, но за да се добие потребно е да се користи спектроскопска анализа, што е спротивно од целта на овој телескоп.
- ***hostgal_photoz:*** Фотометрички *redshift* на галаксијата во која се наоѓа објектот. Иако претставува друга форма на *hostgal_specz*, може да постојат големи отстапувања и треба да се смета како понепрецизно мерење на истата вредност. Сепак, за да се добие оваа вредност се користи фотометричка анализа (со шесте филтри), така што оваа карактеристика многу повеќе ќе ни биде од значење.
- ***hostgal_photoz_err:*** Апроксимација на грешката која постои во мерењето на *hostgal_photoz*.
- ***distmod:*** Растојанието до објектот пресметано преку *redshift* од *hostgal_photoz* користејќи ја општата теорија на релативност.
- ***mwebv:*** MW E(B-V). Оваа карактеристика го покажува "изумирањето" на светлината додека минува низ прашината во нашата галаксија Млечен Пат *(Milky Way – MW)*, од изворот на светлина до леќата на телескопот.
- ***target:*** Лабела која укажува во која класа припаѓа детектираниот објект. Постојат 14 класи во множеството за тренинг.

Секоја редица од множеството претставува информација за само еден објект, а тоа е составено од *7848* редици, што е и бројот на сите објекти во множеството за тренинг.

Бидејќи во податочното множество постојат и категорични карактеристики и континуални, потребно е да се изврши нормализација врз континуалните податоците. Нормализацијата означува делење на секој податок со сумата на сите податоци во дадена колона, со цел тие да се ограничат на вредности помеѓу 0 и 1. Исто така, голем дел од алгоритмите за машинско учење многу подобро учат кога податоците кои ги добиваат се стандардизирани. Под стандардизација се подразбира средната вредност на секоја колона да се нагоди да биде 0, додека стандардната девијација да биде 1. Бидејќи стандардизацијата исто ги ограничува податоците во граници околу нулата, за оваа цел е искористена класата *StandardScaler* од делот за препроцесирање на податоци на библиотеката *sklearn*.

IV. Feature Extraction

Очигледно е дека постои голема несогласност во големината на двете множества. За да се реши овој проблем, наместо да се користат податоците од множеството *training_set* како временска серија, извршена е нивна статистичка анализа и извлечени се нови карактеристики кои ги усогласуваат податоците без да се изгуби голем дел од информацијата. Новите карактеристики кои што се извлечени се за секој објект одделно, посебно за секој од шесте филтри:

- ***mean:*** Средна вредност на измерениот флукс.
- ***median:*** Медијана на измерениот флукс.
- ***max:*** Максимална вредност на измерениот флукс.
- ***min:*** Минимална вредност на измерениот флукс.
- ***std:*** Стандардна девијација на измерениот флукс.
- ***avg_det:*** Средна вредност на детекцијата на објектот, односно колкав процент од времето бил детектиран

По првиот обид за тренирање на класификаторите, повторно е направена анализа на податоците и извлечени се уште 3 нови карактеристики од *training_set,* меѓу кои се наоѓа и една од најзначителните:

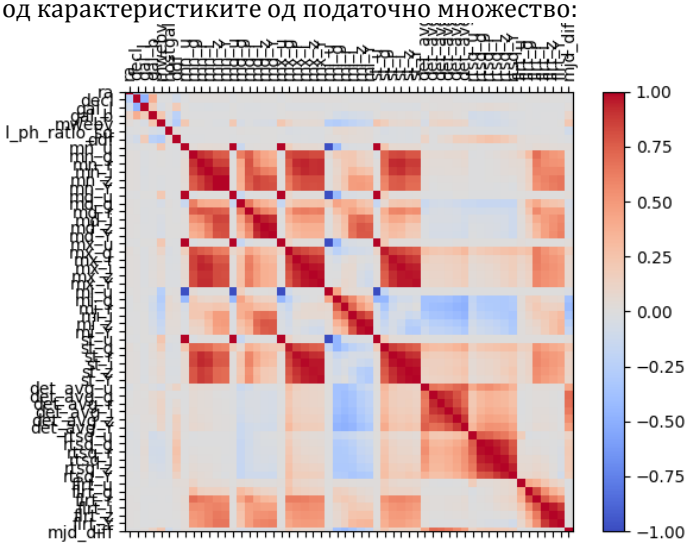
- ***ratio_sq_sum:*** Сума на квадрираниот однос помеѓу измерениот флукс и грешката *(flux ratio squared).*
- ***flux_by_ratio_sq_sum:*** Сума на односот помеѓу измерениот флукс и *flux ratio squared*.
- ***mjd_diff:*** Разлика помеѓу максималната и минималната вредност на денот на мерењето, кога знамето за детекција на објектот е 1 (објектот е детектиран). Оваа карактеристика е од огромно значење при тренирање на класификаторите, бидејќи вредноста која што се добива укажува дали објектот има циклична природа. Со додавање на овој *feature* забележано е значително подобрување на резултатите.

Од множеството *training_set_metadata* исто е извршена екстракција на една нова карактеристика:

- ***hostgal_ph_ratio_sq:*** Квадриран однос помеѓу измерениот фотометрички *redshift* и грешката.

V. Feature Selection

На слика 2 е прикажана корелацијата помеѓу секоја од карактеристиките од податочно множество:



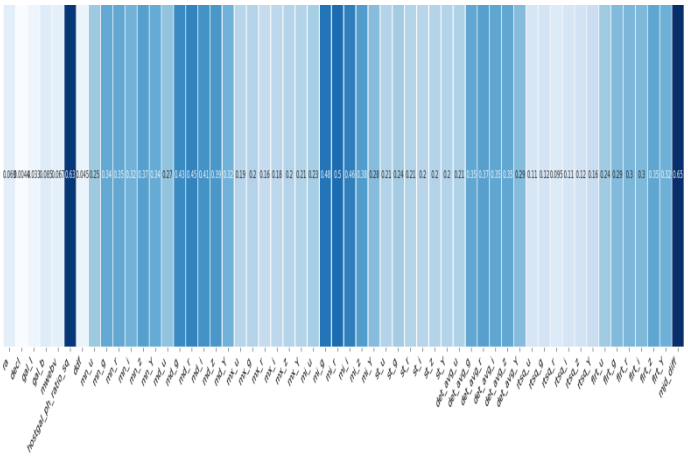
Слика 2. Корелација на карактеристиките на податочното множество

Иако од графикот се гледа дека постои висока позитивна и негативна корелација помеѓу некои од карактеристиките, користењето на ***PCA (Principal Component Analysis)*** не доведе до значително подобрување на конечните резултати. Единствени карактеристики кои беа отстранети се:

- ***decl:*** Незначителна, речиси нулева, корелација со излезот (слика 3). Истата информација се содржи во *gal_b.*
- ***hostgal_specz:*** Како што беше наведено и погоре, оваа карактеристика е иста со *hostgal_photoz,* но бидејќи целта на овој телескоп е да користи фотометрија наместо спектроскопија (и за новодобиените мерења нема да постои спектроскопска анализа), решено е да се отстрани.

- ***distmod:*** Не е значително растојанието до објектот бидејќи е изведено од неговиот *redshift*, а таа информација веќе се содржи во *hostgal_photoz.* За некои од новодобиените податоци нема да постои оваа карактеристика, па решено е уште сега да се отстрани и да не се користи.
- ***hostgal_photoz u hostgal_err:*** Двете карактеристики се втиснати во една *(hostgal_ph_ratio_sq)* со која што се корелирани, така што нема потреба да продолжат да се користат.

На слика 3 е прикажана корелацијата помеѓу секоја од карактеристиките од податочно множество со класната лабела:



Слика 3.Корелација на карактеристиките на податочното множество со излезната класна лабела

VI. Алгоритми

Како што беше наведено погоре, во овој труд се користат пет различни алгоритми за класификација за решавање на овој проблем. Сите алгоритми припаѓаат во групата на надгледувано учење, а три од нив претставуваат ансамбл *(ensemble)* методи – излезот од еден класификатор се користи како влез на нареден итн. Алгоритмите кои што се користат се: *K-Nearest Neighbor (kNN), Gradient Boosting Classifier, Extra Trees Classifier, AdaBoost Classifier* со основен *base* естиматор *Random Forest* и *Multilayer Perceptron Neural Network.*

За одредување на оптималните вредности за хиперпараметрите на алгоритмите се користи *grid search* пристап со 10 кратна *cross-validation*. Барањето на хиперпараметрите е изведено со користење на класата *GridSearchCV* од *sklearn* која овозможува да се тренираат повеќе класификатори истовремено, на секое процесорско јадро поединечно. Ова значително го намалува времето потребно за целосен *grid search* на параметрите.

Поделбата на податочното множество во множество за тренирање и множество за тестирање е извршено со користење на *Stratified K-Fold cross validation.* Стандардниот *K-Fold* го поделува множеството на податоци на *K* еднакви делови од кои *K-1* се користат за тренирање а останатиот дел за тестирање. Процесот се повторува *K* пати, се додека секој од деловите не е искористен за тестирање барем еднаш. Надополнување на овој метод е користење на *stratified* метода. Идејата на поделбата е иста, но сега дополнително на алгоритамот се испраќаат и класните лабели, со цел секој од деловите да содржи ист процент од класите. Поради големата нерамнотежа на класите во даденото податочно множество (претежно се детектирани објекти од класата 90), неопходно е да се користи овој метод за да се обезбеди поголема точност од учењето.

VI-1. Baseline Classifier

Како основен *baseline* класификатор се користи наједноставен *K-Nearest Neighbor* или *kNN*. Тој работи на принцип што одредува во која класа припаѓаат најблиските *K* соседи на новиот податок и според мнозинско гласање ја одредува неговата класа. Значителни хиперпараметри кои се оптимизираа се бројот на соседи *K* и тежината (значењето) на секој од соседите при пресметување на конечното предвидување. За оптимален број на соседи се доби *K = 370*, додека тежините *weights = 'distance'*, што означува дека оние соседи кои се поблиску до новиот примерок имаат поголемо значење отколку оние кои што се подалеку.

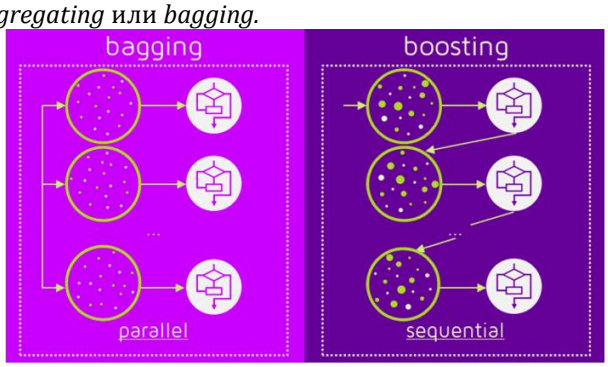
VI-2. Ensemble Classifiers

1) Gradient Boosting

Првиот ансамбл алгоритам кој се користи е *Gradient Boosting*. Тој претставува *boosting* метода која користи ансамбл од повеќе слаби предвидувачки модели, најчесто дрва на одлучување, со цел да се подобри конечниот резултат. *Boosting* означува инкрементално градење на ансамблот, тренирајќи ја секоја нова итерација на моделот така што се дава посебно внимание на оние примероци кои претходниот модел ги класифицирал погрешно. *Gradient Boosting* алгоритамот работи на овој принцип, каде што главна цел е оптимизација на произволна, арбитрарна и диференцијабилна функција на загуба, така што со додавање на слабите предвидувачи итеративно се стреми кон нејзиното минимизирање. За оптимален број на предвидувачи се доби *n_estimators = 500*, додека за стапката на учење *learning_rate = 0.01*.

2) AdaBoost

Наредниот алгоритам исто така претставува *boosting* ансамбл метода, а тој е *AdaBoost* или *Adaptive Boosting.* Кај овој алгоритам се тренираат повеќе слаби предвидувачи истовремено, а конечниот резултат се одредува со тежинска сума на сите естиматори. Тежината на секој од класификаторите се одредува од тоа колку точно ги предвиделе примероците, така што ако нивното предвидување е поточно отколку на другите им се доделува поголема тежина и обратно. Адаптивноса на алгоритамот доаѓа во тоа што сите наредни естиматори во понатамошните итерации се ажурираат, со цел да се подобри предвидувањето на погрешно класифицираните примероци. Оптимални резултати се добија кога за основен естиматор се користеше *Random Forest* со *n_estimators = 1200*. *Random Forest* исто претставува ансамбл алгоритам сам во себе, но тој наместо *boosting* користи *bootstrap aggregating* или *bagging.*



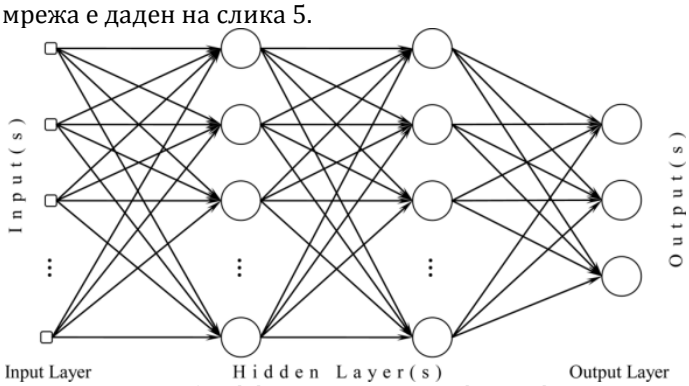
Слика 4. Разлика помеѓу bagging и boosting

3) Extra Trees

Подобра верзија на *Random Forest* претставува *Extra Trees* алгоритамот, чија единствена разлика е во начинот на кој што се прави поделба или *split* на податоците кој ќе го добие секое од поединечните дрва на одлучување. Додека кај обичен *Random Forest split* се прави со пресметување на локално оптимална *feature/split* комбинација која ја зема секоја карактеристика во предвид, кај *Extra Trees* се доделува потполно случајна вредност за *split.* Ова доведува до поразновидни дрва и помалку *splitters* за евалуација при тренирањето на алгоритамот. Како што беше наменето погоре, ансамбл методата која ја користи *Extra Trees* е користи *bootstrap aggregating*. *Bagging,* исто како и *boosting*, користи повеќе слаби предвидувачи за да со нивна агрегација се добие еден силен класификатор. Разликата меѓу двете методи лежи во тоа што кај *bagging* секој од естиматорите користи случајно избрано подмножество од целосното податочно множество за тренинг, додека градењето на моделот не е итеративно туку паралелно. Тежините во тежинската сума која го одредува крајниот резултат се еднакви, односно се користи обична средна вредност од предвидувањето на секој од класификаторите. За оптимален број на естиматори на се доби *n_estimators = 1200,* користејќи ги сите *features* на секој *split*.

VI-3. Multilayer Perceptron Neural Network

Повеќеслојниот перцептрон претставува *feedforward* вештачка невронска мрежа. *Feedforward* означува дека податоците на мрежата се предаваат од влезниот слој, низ скриените слоеви, кон излезниот, додека невроните кои што се активираат зависи од функцијата на активирање и тежините на врските. Типичен *MLP* се состои од најмалку три слоеви, влезен, скриен и излезен слој. Сите неврони освен влезните користат нелинеарна функција на активирање која одредува дали тој ќе се "вклучи" или активира. Можноста за повеќе слоеви и нелинеарноста на функциите за активирање го одделуваат *MLP* од обичен линеарен перцептрон и овозможуваат тој да се користи за податоци кои што не се линеарно раздвојливи. Изгледот на една типична невронска мрежа е даден на слика 5.



Слика 5. Multilayer Perceptron Neural Network

Начинот на учење на мрежата се сведува на метода на надгледувано учење наречена пропагација наназад или *backpropagation.* При оваа постапка, по минување на податоците низ мрежата и добивање на конечното предвидување, тоа се споредува со вистинското од тренинг множеството, а разликата се користи за ажурирање на тежините на врските помеѓу невроните, одејќи од излезот наназад кон влезот.

Градбата на добиената оптимална мрежа се состои од влезен слој со 55 неврони (број на карактеристики од податочното множество, еден излезен слој со 14 неврони (број на класни лабели во кои што треба да се класифицираат податоците) и три скриени слоеви со по 50 неврони. Бројот на неврони во скриените слоеви е добиен со формулата:

$$HLN = \frac{2}{3} * ILN + OLN$$

каде *HLN* е бројот на неврони во скриениот слој, *ILN* е бројот на неврони во влезниот слој, додека *OLN* е бројот на неврони во излезниот слој. Бројот на скриени слоеви е одреден така што вкупно постојат приближно 3 пати повеќе неврони во скриените слоеви отколку во влезниот.

Во сите три скриени слоеви се користи *tanh* функција на активирање, што претставува издолжена сигомоидална функција по *y*-оската така што се наоѓа помеѓу *-1* и *1* и минува низ нулата. Излезниот слој има *softmax* функција на активирање, бидејќи предвидувањето на податоците не треба да биде категорично туку треба да се добијат веројатностите на припадност на примерокот во секоја од класите. За иницијализација на почетните тежини на врските се користи *Xavier* или *Glorot* иницијализација, која ги поставува тежините на случајна вредност од униформна дистрибуција која е претставена со следната релација:

$$\pm \frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}$$

каде n_i е бројот на врски кои влегуваат или *"fan-in"*, додека n_{i+1} е бројот на врски кои излегуваат од дадениот слој или *"fan-out"*.

Со цел да се избегне *over-fitting*, во мрежата се користат *dropout* слоеви за регуларизација. Овие слоеви имаат улога случајно да отстранат одреден процент од невроните од скриениот слој за кои се поврзани. Во конкретната мрежа стапката на остранување е *0.2* за првите два скриени слоеви, додека за последниот е *0.1*. Отстранувањето на неврони ќе доведе до значително зголемување на вредностите на тежините во мрежата, па затоа се користи и ограничување кое гласи дека збирот на сите тежини во секој од слоевите не смее да надмине одредена претходно зададена вредност, во конкретниот случај *max_norm = 3.*

Уште една постапка која што се користи е *batch normalization*. Како што се изврши нормализација на основните податоци во делот *III*, *batch normalization* претставува нормализација на секој *batch* поединечно, при секое негово поминување од еден во друг слој.

Тренирањето на мрежата се одвива во 1000 епохи со големина на еден *batch* од 256. Големината на *batch* одредува колку примероци се процесираат пред моделот да се ажурира. За оптимизација на мрежата се користи *Adam* оптимизациски алгоритам со стапка на учење *learning_rate = 0.001*, за метрика на евалуација се користи категорична прецизност или *categorical accuracy,* додека за функција на загуба се користи средна квадратна логаритамска грешка (MSLE):

$$MSLE = \frac{1}{n}\sum_1^n [\log(Y_{(n)\ estimated} + 1) - \log(Y_{(n)\ true} + 1)]^2$$

VII. Метрика на оценување

Во овој труд се користат две метрики за одредување на точноста на конечниот резултат. Првата е метриката која што се користи во самиот натпревар на *Kaggle – multiweighted log-loss*, додека втората, со цел да се добие поразбирлива вредност за точноста на алгоритмите, е т.н. *F1* метрика.

VII-1. Multiweighted log-loss metric

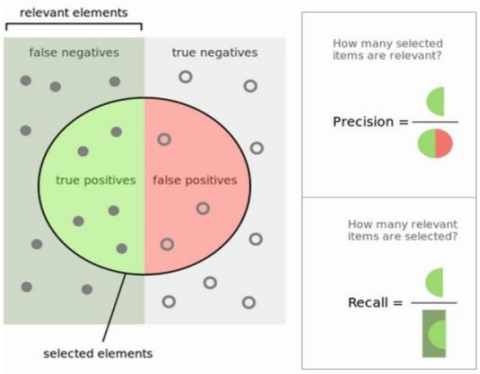
Основната метрика која што се користи во овој труд е тежинска, логаритамска функција на загуба која го има следниот облик:

$$L = -\frac{\sum_{j=1}^{M} w_j \cdot \sum_{i=1}^{N} \frac{1}{N_j} \tau_{i,j} \ln(P_{ij})}{\sum_{j=1}^{M} w_j}$$

каде $\tau_{i,j} = 1$ ако *i*-тиот објект доаѓа од *j*-тата класа и 0 кога важи спротивното, N_j е бројот на објекти во секоја дадена класа *j*, додека w_j се индивидуалните тежини за секоја класа која го рефлектира нејзиниот релативен придонес во целокупната метрика (која класа е побитно да се класифицира поточно во однос на другите). Овие тежини не беа дадени од страна на организаторите на натпреварот, но учесниците ги имаа пресметано така што се користат во конечното решение во овој труд.

VII-2. F1 – score metric

Определување на точноста на алгоритмите според обична прецизност или *accuracy* дава само резултати во однос на бројот на точно класифицирани примероци во однос на вкупниот број. Ова може да доведе до добивање на наизглед поголема точност на алгоритамот од вистинската поради постоењето на лажно позитивни и лажно негативни примероци. За таа цел во овој труд е искористена метриката F1 (слика 5).



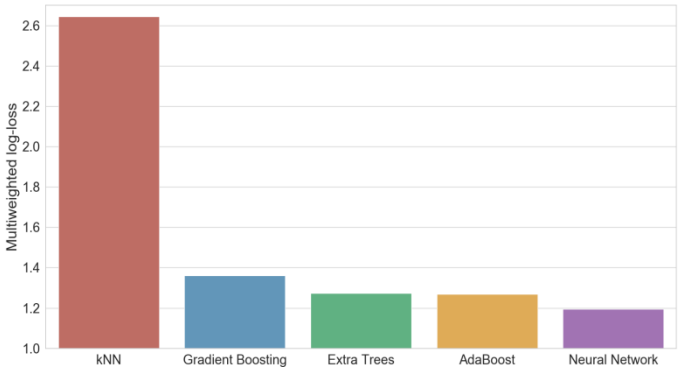
Слика 6. Графички приказ на F1 score

Принципот на оваа метрика е одредувањето на бројот на примероци кои се вистински позитивни, вистински негативни, лажно позитивни и лажно негативни и потоа одредување на *precision* и *recall*. *Precision* претставува колку од примероците класифицирани како позитивни се релевантни, односно, точно предвидени. *Recall* претставува колку од релевантните примероци, односно, вистински позитивните примероци се класифицирани како позитивни. F1 метриката е значително подобра во однос на други метрики за прецизност и таа претставува хармониска средина на *precision* и *recall*:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

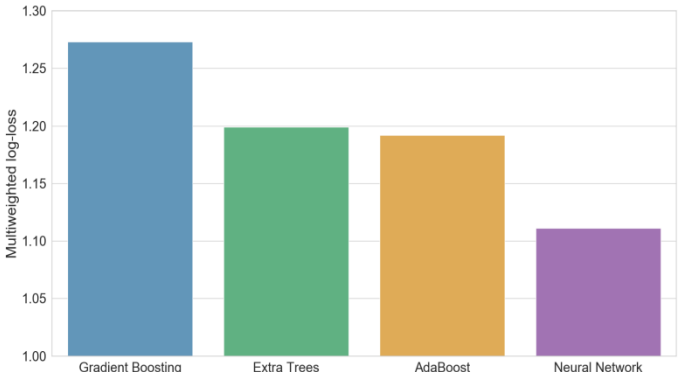
VIII. Резултати

Конечните резултати од сите алгоритми пред да се изврши *feature extraction* од дадените карактеристики се дадени на графиците претставени на слика 7:

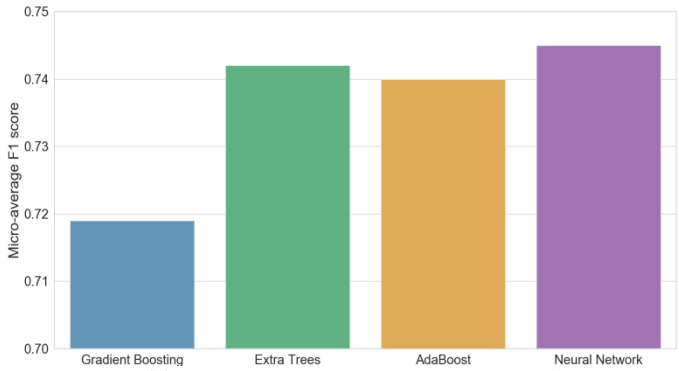


Слика 7. Споредба на multiweighted log-loss на алгоритмите со користење на основите податоци

Со извлекување на новите карактеристики забележано е значително подобрување во минимизацијата на *log-loss* функцијата. Новите резултати се дадени на графикот на слика 8, додека резултатите од *F1* метриката се претставени на графикот на слика 9. Во споредбата не е вклучен *kNN,* со цел подобро да се прикаже разликата помеѓу останатите четири алгоритми.



Слика 8. Споредба на multiweighted log-loss на алгоритмите со користење на новите податоци



Слика 9. Споредба на F1 score со micro average на секоја од класите со користење на новите податоци

Од резултатите може да се заклучи дека најдобро решение на проблемот се доби со *MLP* невронската мрежа, со *log-loss score* од *1.111* и *micro F1 score* од *0.745.* Може да се забележи дека иако *Extra Trees* даде речиси ист *F1 score* со мрежата, таа подобро ја минимизира *log-loss* функцијата што се должи на тежините на класите кои се позначајни да бидат точно одредени во крајното класифицирање на податоците.

IX. Заклучок и идна работа

Во овој труд се разгледаа пет можни пристапи за решавање на проблемот на класификација на астрофизичките појави во нашиот универзум во нерамнотежно распределено множество од податоци. Најдобриот модел беше *MLP* невронската мрежа која успеа релативно добро да ги распредели објектите во нивните вистински класи, со мал дел на лажно позитивни примероци. Поголемиот дел од ансамбл алгоритмите дадоа добри резултати за разлика од стандардните методи, што се должи на комплексноста на самиот проблем и начинот на кој се приложени податоците. За во иднина, би можело повторно да се наврати на податоците и да се изврши нова анализа, со цел да се добијат нови врски помеѓу карактеристиките што сега не постојат. Еден метод кој што може да се проба е претворање на временската низа во фазен домен со користење на т.н. *wavelet* трансформација, која што не беше применета во рамките на овој труд, за да се провери дали ќе дојде до подобрување на конечните резултати.