SUNS – Zadanie 3:

Súborové učenie

Petra Kirschová

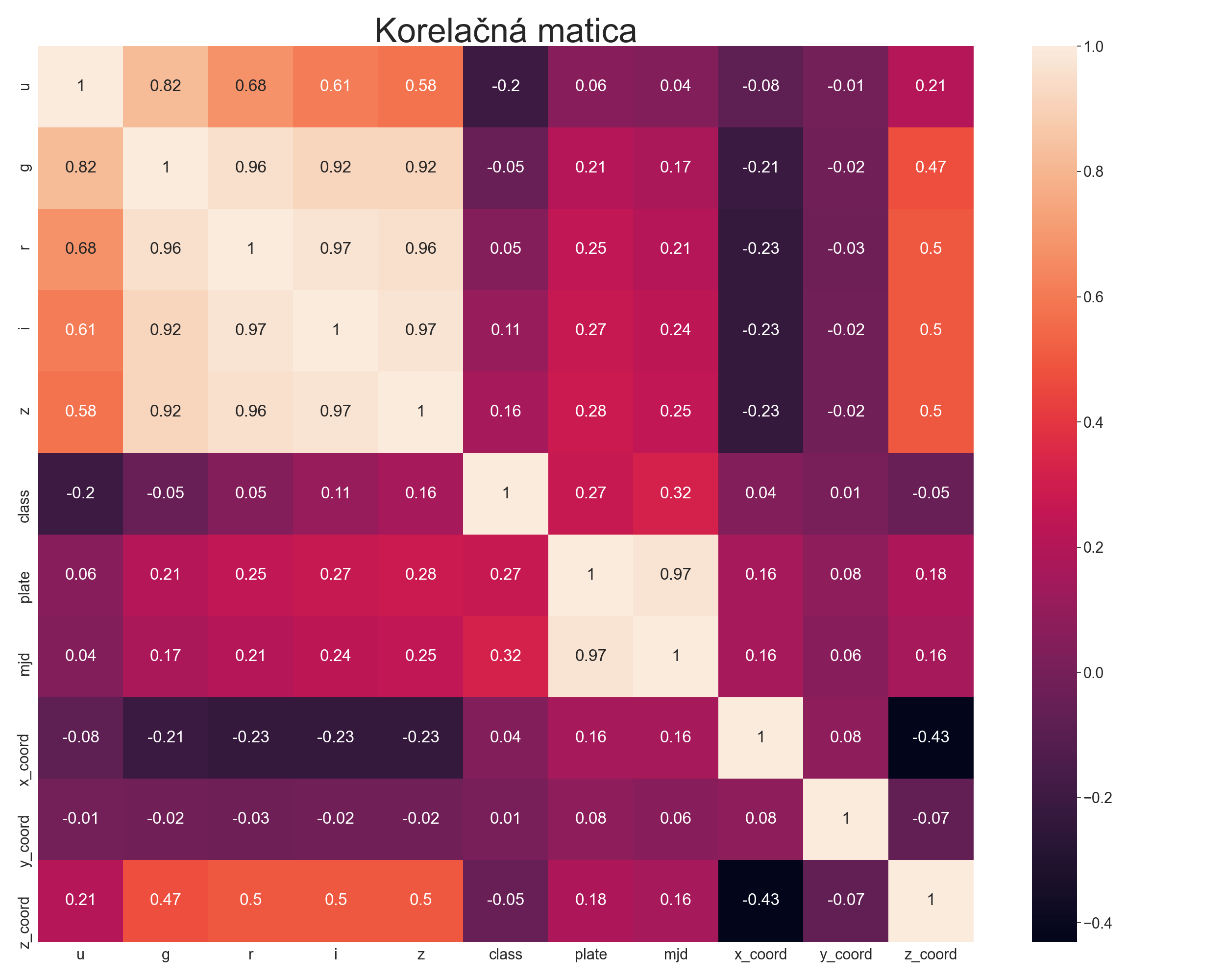
# Analýza a spracovanie dát

Na načítanie a prípravu dát som použila knižnicu pandas. Boli k dispozícii zvlášť trénovacie (11740 záznamov) a testovacie dáta (2935 záznamov).

Datasety neobsahovali žiadne null hodnoty a tiež nebolo potrebné mazať outliery, keďže strojové učenie nebolo ovplyvnené prítomnosťou outlierov.

Z datasetov som odstránila stĺpce, ktoré predstavujú identifikáciu záznamov: *objid*, *specobjid*, *run*, *rerun, camcol* a *field.*

Po odstránení stĺpcov vyzerala korelačná matica takto:



Na korelačnej matici som sledovala stĺpec *class* a súradnice *x\_coord*, *y\_coord* a *z\_coord*.

Stĺpec *class* má pomerne nízke korelácie s ostatnými ukazovateľmi, no napriek tomu klasifikátor dosahoval veľmi dobré výsledky, keď som na trénovanie použila celý dataset (okrem *class*).

Zo súradníc *x\_coord, y\_coord* a*z\_coord* majú *z\_coord* vysoké korelácie takmer so všetkými ostatnými parametrami, hlavne s meraniami *u*, *g*, *r*, *i*, *z.*

# Klasifikácia

Zo súborových klasifikátorov som si vybrala random forest klasifikátor a ako ďalší som zvolila MLP klasifikátor. Na ich trénovanie som použila knižnicu sklearn*,* na vykreslenie grafov seaborn a matplotlib a na vykreslenie a exportovanie stromu graphviz.

## Random forest klasifikátor

Pri random foreste nebolo potrebné dáta normalizovať ani konvertovať na číselné reprezentácie.

Vstupné parametre trénovania sú stĺpce: *u, g, r, i, z, plate, mjd, x\_coord, y\_coord* a *z\_coord*.

Výstupným parametrom je stĺpec *class*, ktorý obsahuje 3 rôzne triedy objektov: STAR, GALAXY a QSO.

Pre klasifikátor som nastavila:

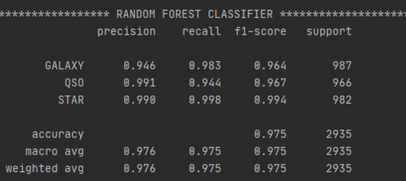
classifier = **RandomForestClassifier**(**criterion**='entropy',

**max\_leaf\_nodes**=40,**min\_samples\_leaf**=10)

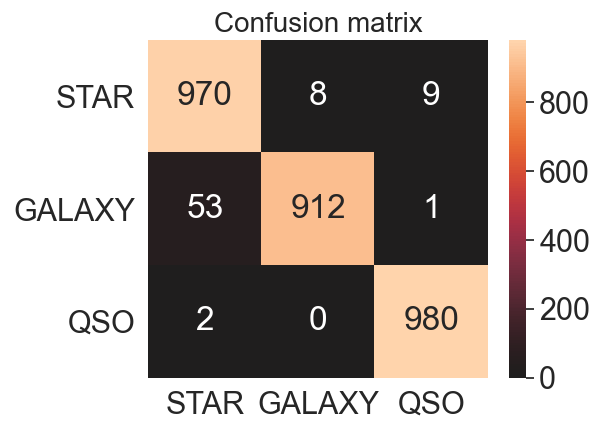
* **criterion=‘entropy’** – miera správnosti/úspešnosti rozdeľovania dát
* **max\_leaf\_nodes=40** - maximálny počet listov v stromoch
* **min\_samples\_leaf=10** minimálny počet vzoriek v liste

### Výsledky

Pri trénovaní random forest klasifikátora som dosiahla veľmi dobré výsledky, ktoré sa pohybovali medzi 97-98%. Na classification report sú uvedené úspešnosti utriedenia do jednotlivých kategórií a tiež celková presnosť klasifikácie.



Na confusion matrix je tiež vidno, že trénovanie prebehlo veľmi úspešne. Na diagonále sa nachádzajú vzorky, ktoré boli správne zatriedené do jednotlivých kategórií. Nesprávne zatriedených vzoriek je v porovnaní s veľkosťou datasetu zanedbateľné množstvo.



## Slabý klasifikátor

Slabým klasifikátorom v random forest modeli je 1 strom. V modeli som nechala defaultný počet slabých klasifikátorov – 100, t.j. celkovo sa vyhodnocuje 100 rozhodovacích stromov.

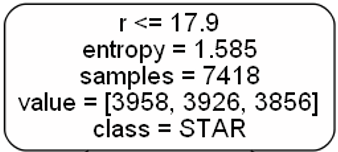
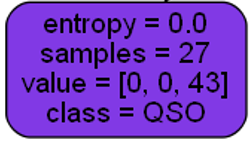
Jeden uzol stromu obsahuje:

* *Rozhodovaciu podmienku*
* *Entropy* – miera rozdielnosti dát
* *Samples* - počet vzoriek
* *Value* – koľko vzoriek prislúcha jednotlivým triedam
* *Class* – výsledok zatriedenia v danom uzli stromu

Strom na základe rozhodovacej podmienky rozdelí dáta na 2 časti – vľavo pôjdu tie, pre ktoré podmienka platí a vpravo ostatné. Tento krok sa opakuje, kým nie sú splnené zastavovacie podmienky trénovania.

Entropia vyjadruje, ako veľmi sú dáta v danej vzorke navzájom rozdielne. V roote stromu je entropia vysoká, pretože všetky triedy sú takmer rovnako zastúpené. Klesaním nižšie v strome sa entropia znižuje a v niektorých listoch dosiahla aj nulu.

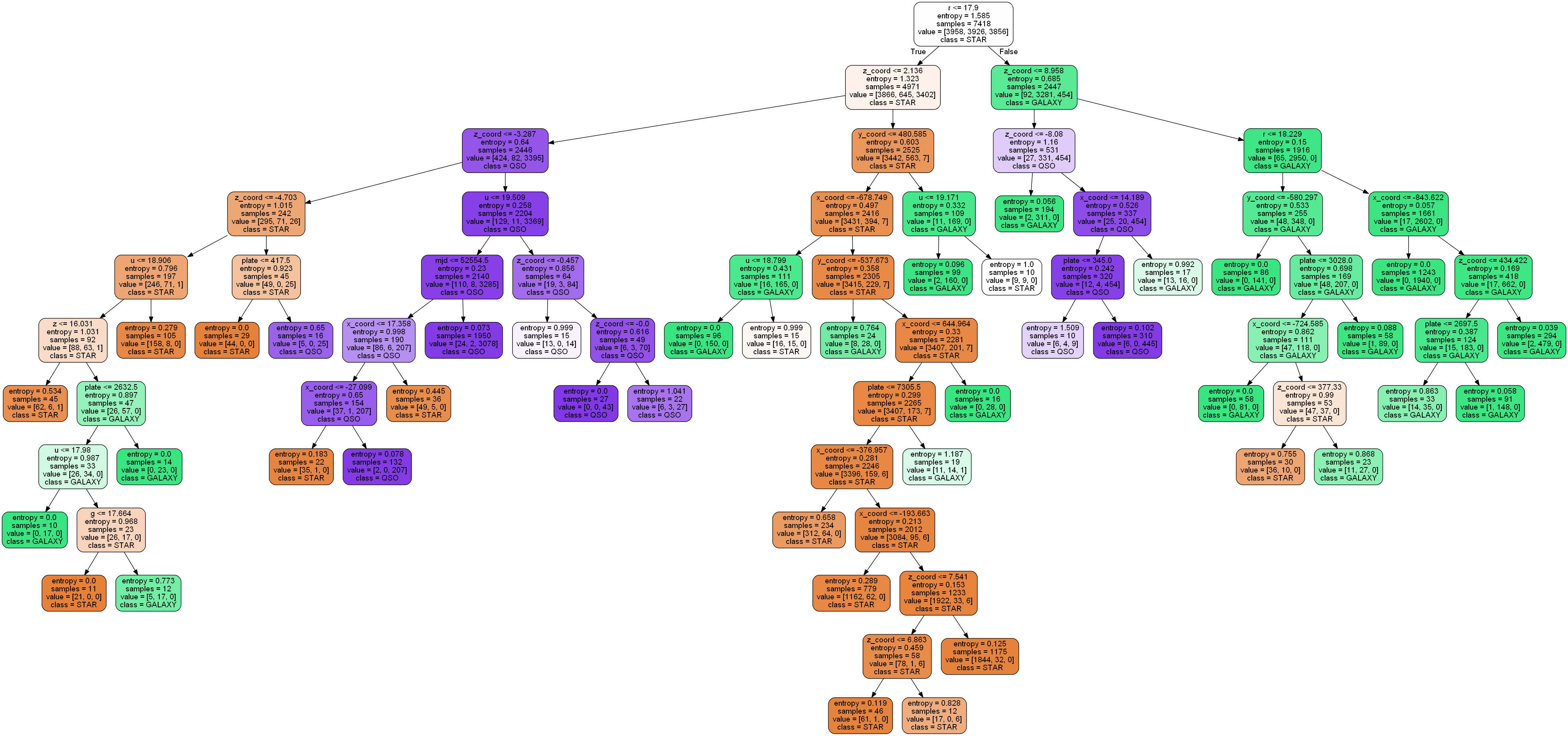
**Root stromu List stromu s nulovou entropiou**

Výslednou triedou v danom uzle stromu je tá trieda, ktorá má vo vzorke najväčšie zastúpenie (najväčšie číslo v poli *value*).

Vykreslený strom je výsledkom pruningu, kvôli tomu, že nenastavenie zastavovacej podmienky viedlo k pretrénovaniu modelu. Pôvodný strom bol tiež príliš veľký a koncové uzly obsahovali málo dát na to, aby boli smerodajné. Pri veľkom strome vo väčšine vetiev nastala situácia, kedy sa ďalším delením stromu nezískala žiadna výpovedná hodnota, keďže sa vzorky v oboch častiach datasetu zaradili do rovnakej triedy.

Po pruningu sa v strome nachádza tiež zopár takých vetiev, ktorých rozdelením dostaneme rovnaký výsledok, no vzhľadom na dosiahnutú úspešnosť je tento model relatívne dobre rozložený. Sprísňovanie zastavovacích podmienok sa potom odzrkadľovalo na celkovej úspešnosti klasifikátora.

******

## MLP klasifikátor

Vstup a výstup MLP klasifikátora bol rovnaký, ako pri random foreste, vstupné dáta sa na rozdiel od random forestu museli normalizovať. Výstupnú množinu, t.j. stĺpec *class* som nechala bez úprav v pôvodnom tvare – trénovanie fungovalo, aj keď som vo výstupnej množine nechala triedy vo formáte string.

Nastavenie klasifikátora:

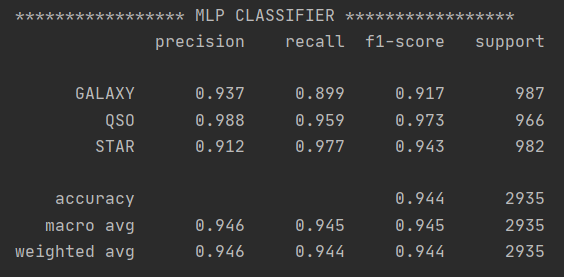
classifier = **MLPClassifier**(**max\_iter**=600, **alpha**=0.01,

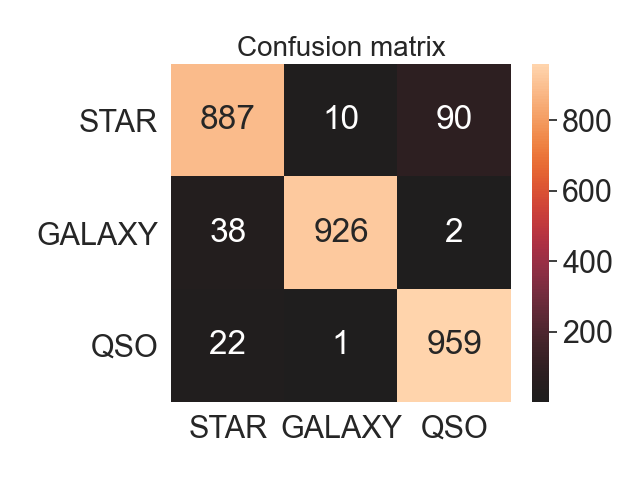
**hidden\_layer\_sizes**=(20,))

* aktivačná funkcia=**relu** (default)
* solver=**adam** (default)
* **max\_iter=600** - maximálny počet iterácii
* **alpha= 0.01** – regularizačná penalta
* **hidden\_layer\_sizes=(20,)** - 1 skrytá vrstva s 20 neurónmi

### Výsledky

MLP klasifikátor dosiahol trochu nižšie úspešnosti, ako random forest, ale stále sú veľmi vysoké. Celkovo sa presnosti MLP klasifikátora pohybovali okolo 93-94%.





# Regresia

Spomedzi súborových klasifikátorov som na regresiu použila random forest regresor a ako ďalší regresor som si vybrala k-nearest neighbors regresor. Regresory som implementovala pomocou knižnice sklearn a na vykreslenie grafov som použila seaborn*,* matplotliba plotly.

## Random forest regresor

Pri random forest regresore nebolo potrebné dáta normalizovať, ale vstupný stĺpec class bolo treba konvertovať na číselné reprezentácie tried.

Vstupné parametre trénovania sú stĺpce: *u, g, r, i, z, plate, mjd* a*class*.

Výstupnými parametrami sú stĺpce so súradnicami *x\_coord, y\_coord* a *z\_coord*, ktoré určujú polohu sledovaného objektu vo vesmíre.

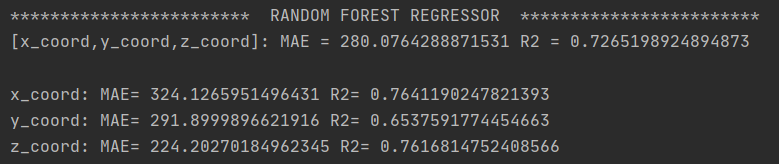
Nastavenie regresora:

regressor = **RandomForestRegressor**(**criterion**="mse", **max\_depth**=15)

* **criterion="mse"** – miera správnosti rozdelenia dát = mean squared error
* **max\_depth=15** – maximálna hĺbka stromu, zastavovacia podmienka

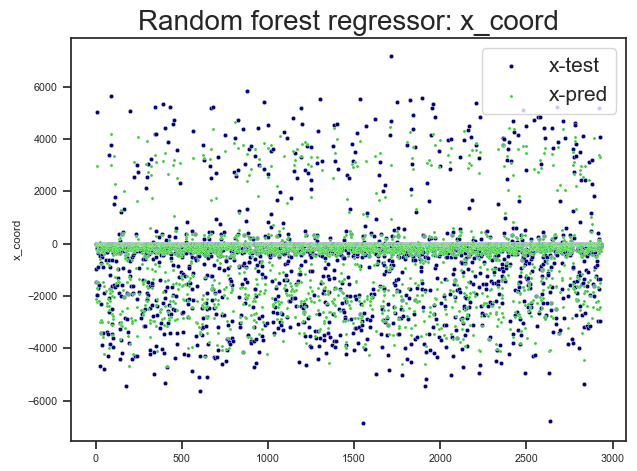
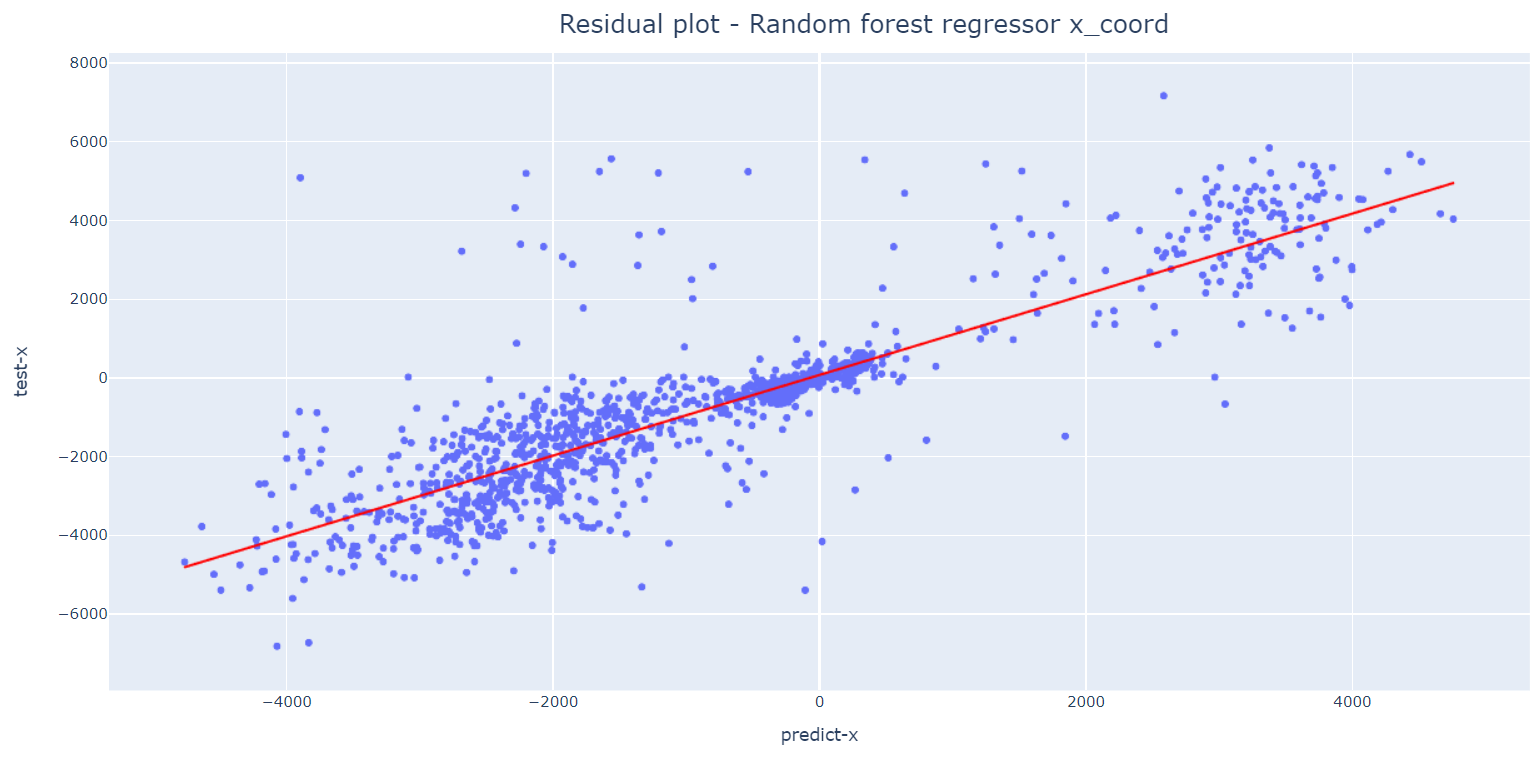
### Výsledky

Random forest regresor dosahoval dosť dobré výsledky, dosiahol R2 score pre celý výstup okolo 73% a mean absolute error cca 280. Pre jednotlivé súradnice bola pre *x\_coord* najvyššia chyba a pre *z\_coord* najnižšia. R2 score bolo najvyššie pre *x\_coord* a najnižšie pre *y\_coord*.

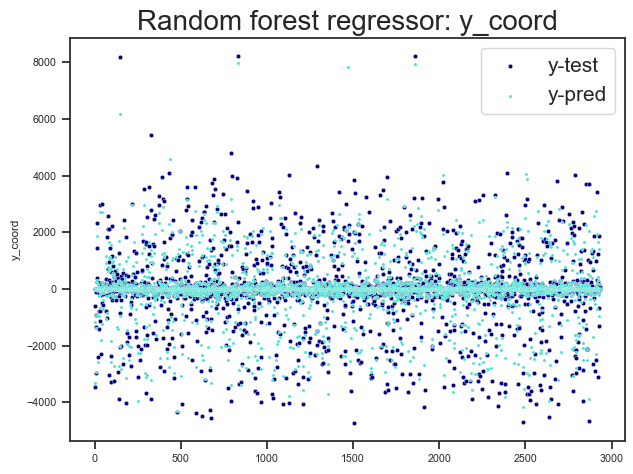
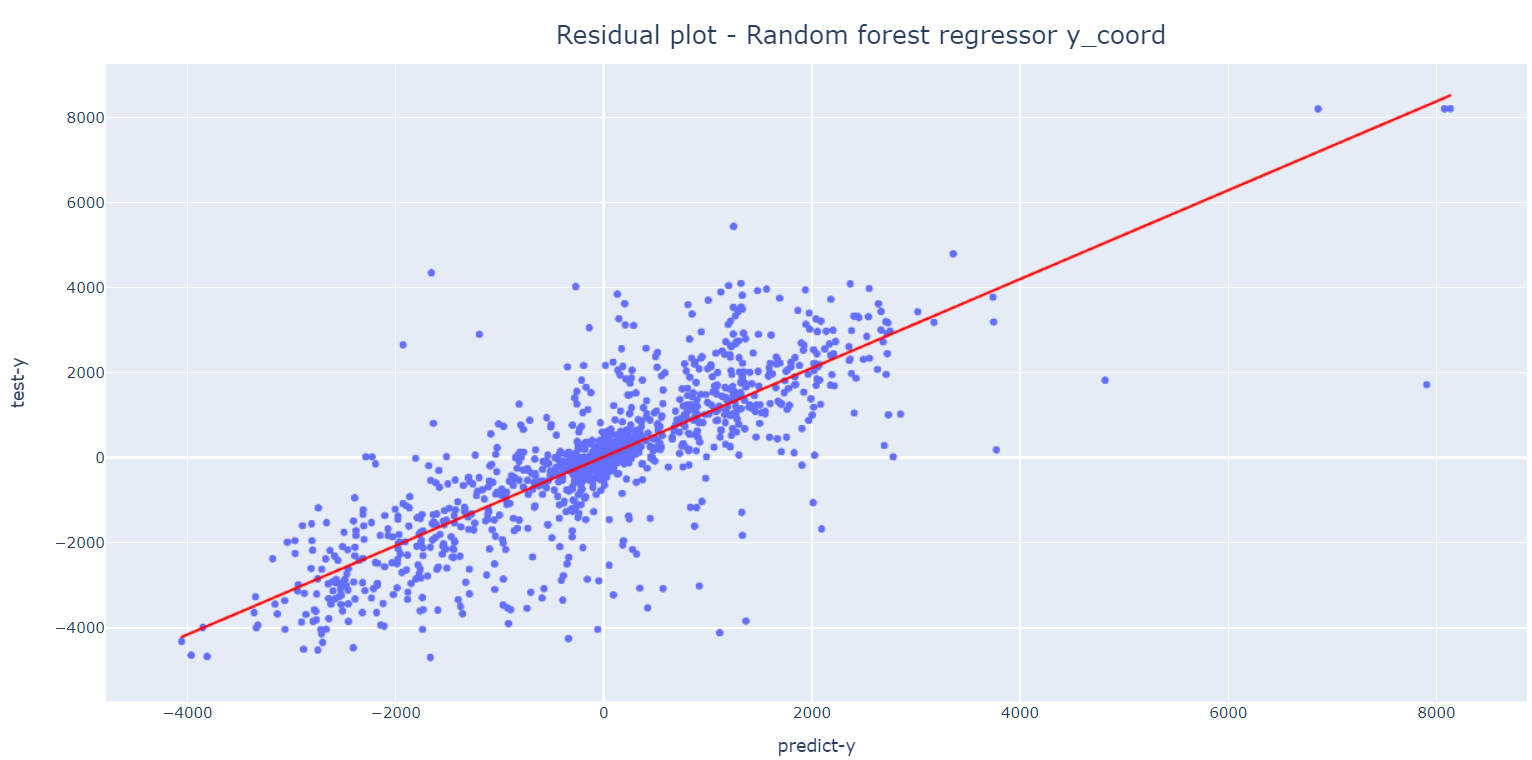


V 2D a 3D grafoch som porovnala výstup testovacích a predikovaných dát. Na grafoch je vidno, že regresor priblížil výstupné hodnoty relatívne dobre, pozície predpovedaných bodov sú veľmi blízko k reálnym bodom. Na 2D grafoch sú porovnané predpovedané a očakávané výstupy (v závislosti od id alebo pri residual plote priamo závislosť medzi predikovaným a očakávaným výstupom) pre každú súradnicu zvlášť. 3D graf obsahuje body *[x\_coord, y\_coord, z\_coord]* v trojrozmernom priestore.

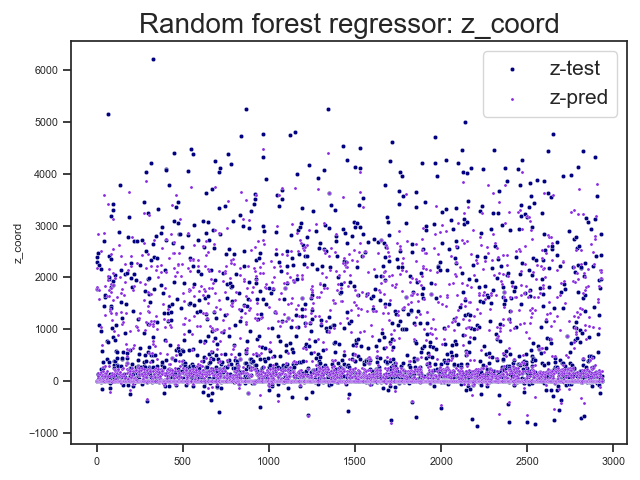
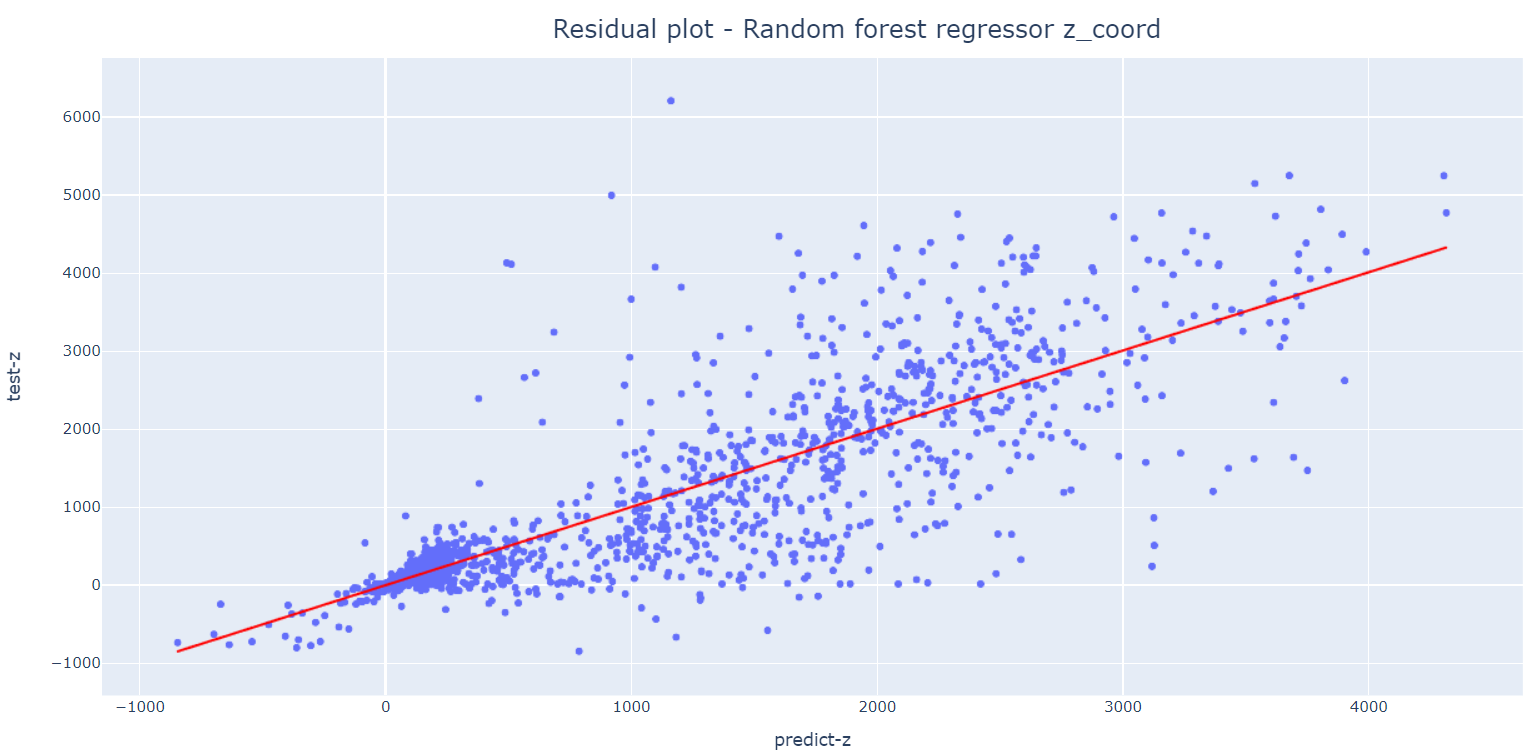
**x\_coord: porovnanie očakávaných hodnôt s predpovedanými + residual plot**

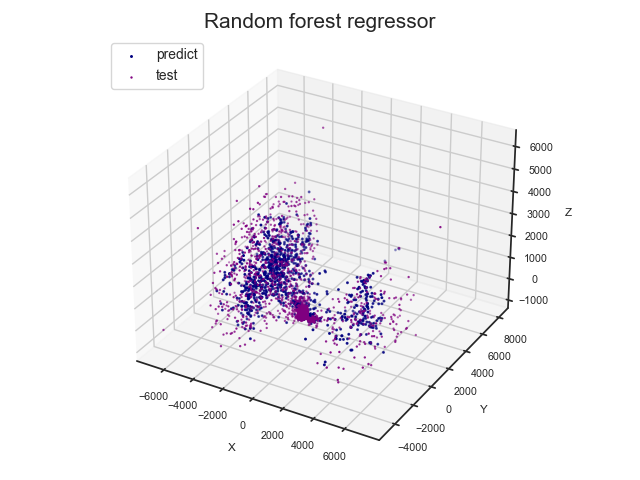
**y\_coord: porovnanie očakávaných hodnôt s predpovedanými + residual plot**

**z\_coord: porovnanie očakávaných hodnôt s predpovedanými + residual plot**

[**x\_coord, y\_coord, z\_coord]: porovnanie očakávaných hodnôt s predpovedanými**



## K-nearest neighbors regresor

K-nearest neighbors regresor predpovedá výsledné hodnoty na základe toho, aké hodnoty dosahujú iné dáta v jeho okolí (neighbors). Dáta tiež nebolo treba normalizovať, ale vstupný stĺpec *class* sa konvertoval na číselné reprezentácie tried.

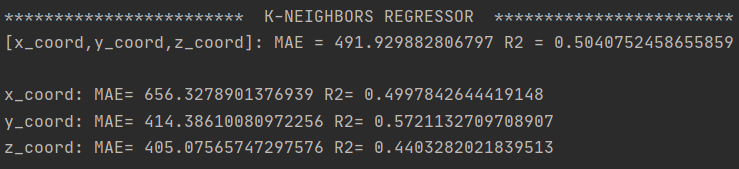
Nastavenie regresora:

regressor = **KNeighborsRegressor**(**n\_neighbors**=8, **weights**='distance')

* **n\_neighbors=8** – počet „susedov“, podľa ktorých sa vyhodnocuje výsledok
* **weights='distance'** – váhy susedov sa určujú podľa ich vzdialenosti k sledovanému bodu, t.j. body, ktoré sú bližšie viac ovplyvnia výsledok, ako tie, ktoré sú ďalej.

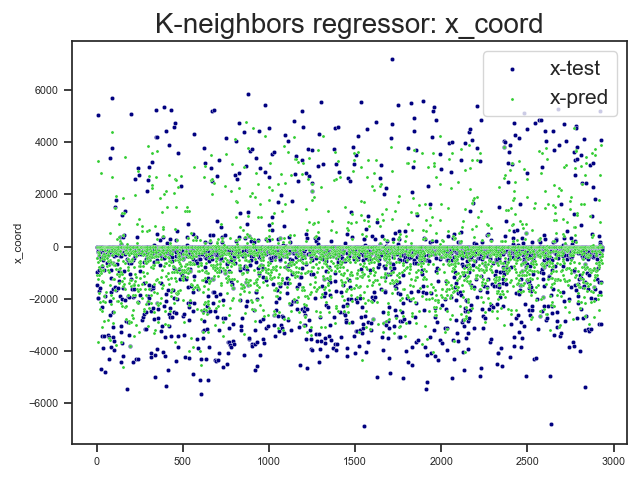
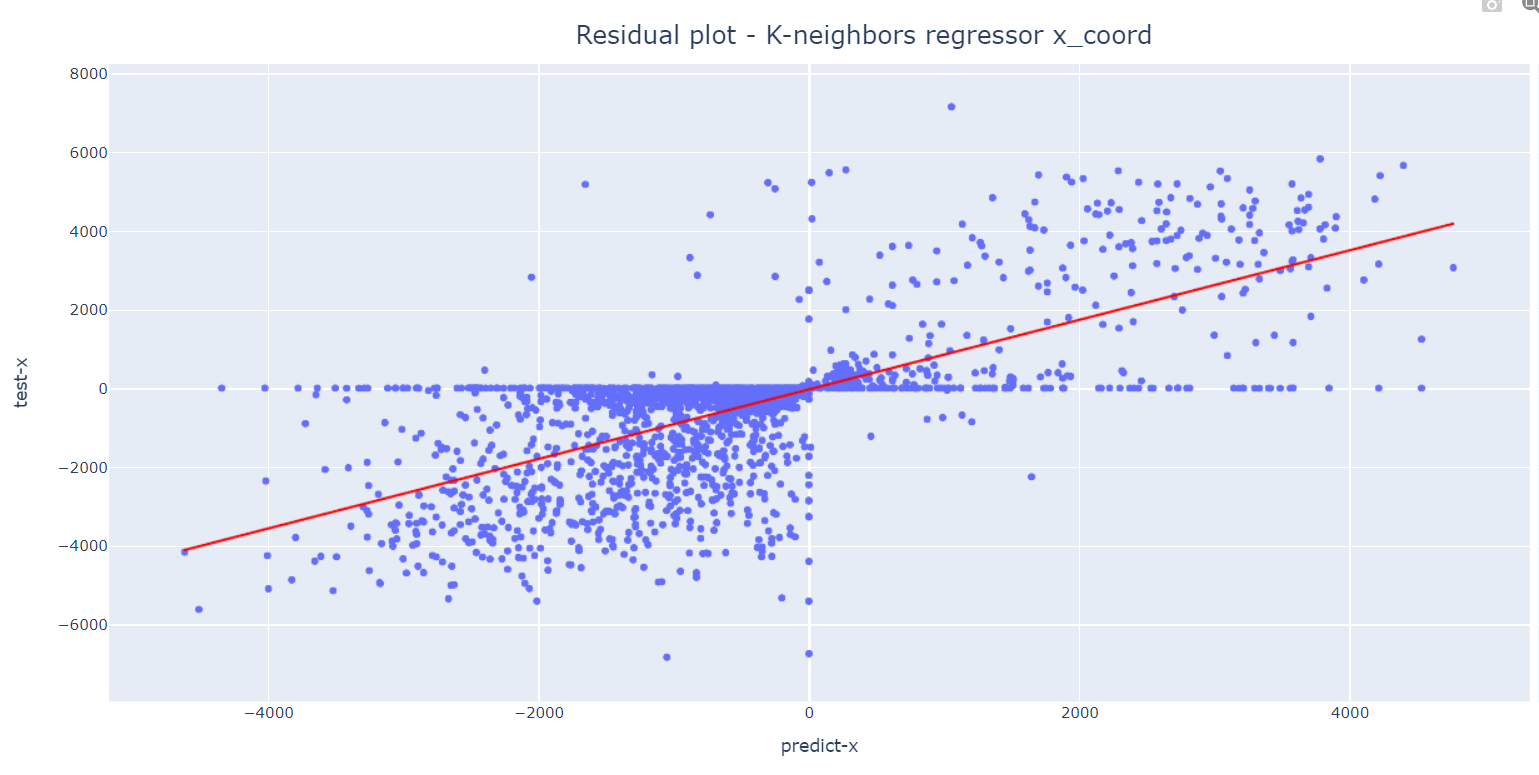
### Výsledky

Výsledky KNN regresora boli horšie, ako pri random forest, R2 bolo celkovo okolo 0.5, MAE cca 490, pre jednotlivé súradnice sa chyby pohybovali približne medzi 400-700, R2 score medzi 0.4-0.6.

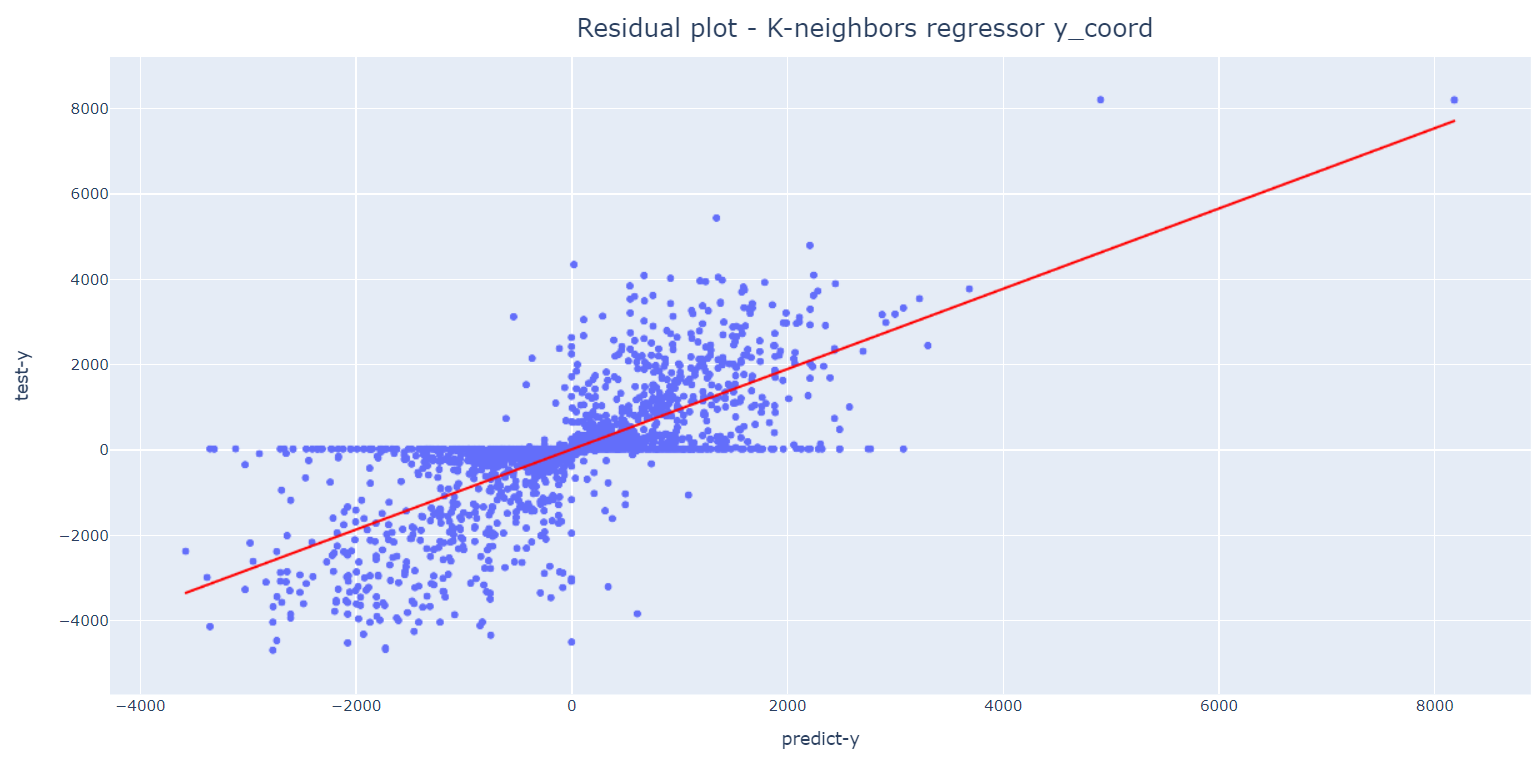
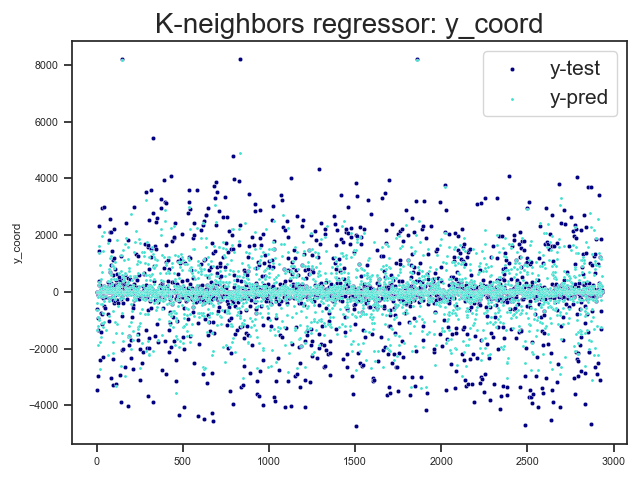


Grafy, na ktorých sú vykreslené pôvodné a predpovedané výstupy sú na prvý pohľad veľmi podobné s grafmi random forest regresora, rozdiely sú pozorovateľné na residual grafoch. Pri tomto regresore sú predpovedané body rozptýlené viac, ako pri random foreste a najviac sa ich zoskupuje pozdĺž nuly.

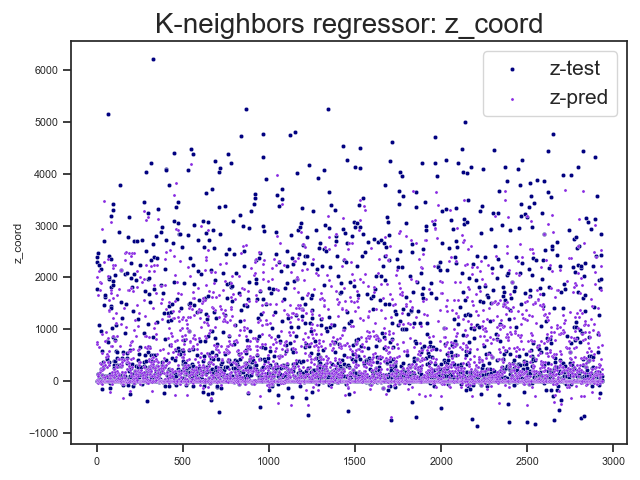
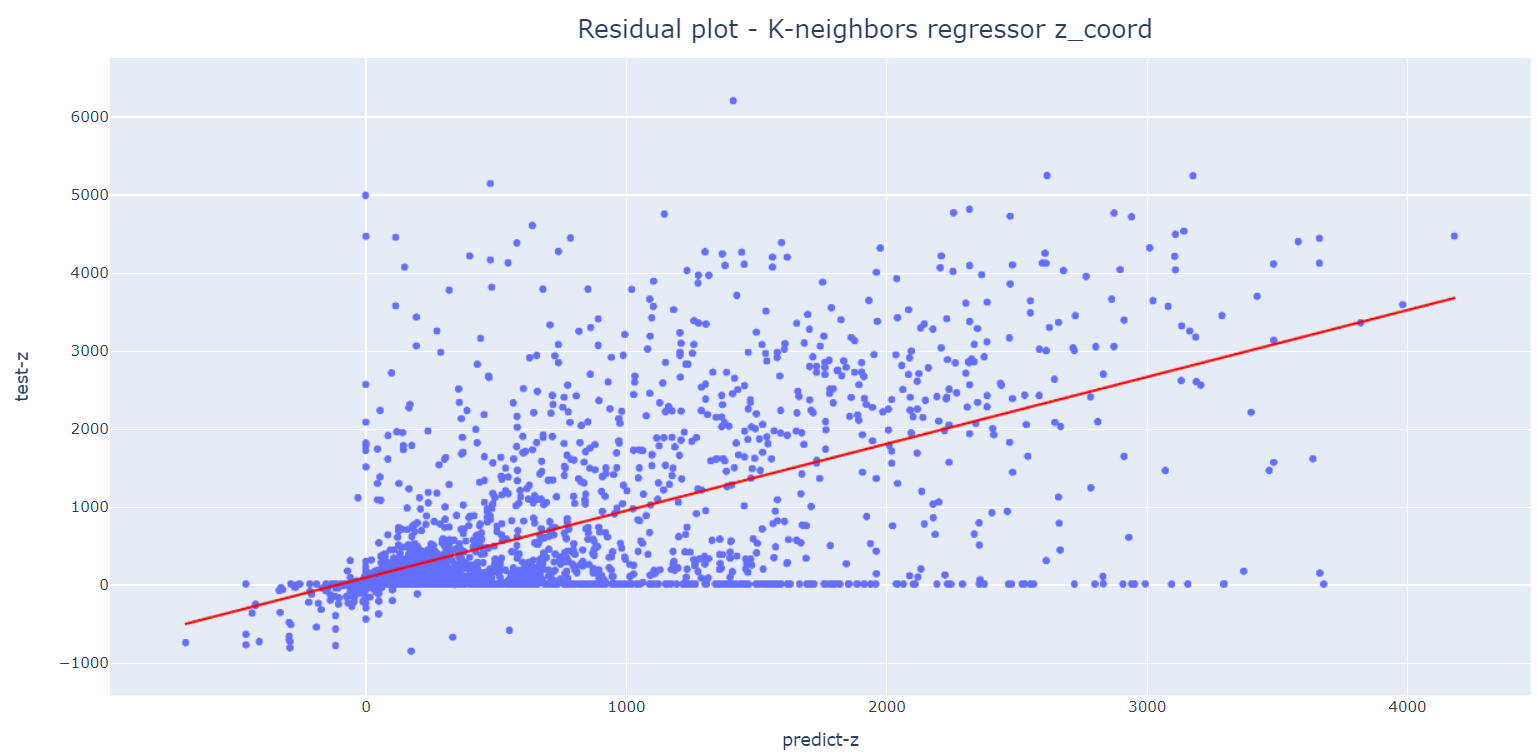
**x\_coord: porovnanie očakávaných hodnôt s predpovedanými + residual plot**

**y\_coord: porovnanie očakávaných hodnôt s predpovedanými + residual plot**



**z\_coord: porovnanie očakávaných hodnôt s predpovedanými + residual plot**

[**x\_coord, y\_coord, z\_coord]: porovnanie očakávaných hodnôt s predpovedanými**

