Универзитет “Св. Кирил и Методиј“ - Скопје
Факултет за информатички науки и компјутерско инженерство

Анализа, визуелизации и модели
на податоци за автомобили од 1985-та година

Податочно Рударство

Ангела Маџар, 181010

Петар Поповски, 181007

Скопје, 2021

Содржина

Апстракт

Експоненцијалниот раст на количеството на достапни податоци, како и неверојатниот напредок во хардверот, овозможија реализација на концептите на машинското учење и податочното рударство кои некогаш постоеле само на хартија. Како резултат на технолошкиот развој, имплементацијата на широк спектар на техники за анализа, визуелизација, обработка и екстракција на скриени патерни на податоци во различни области е изводлива и практична. Целта на овој труд е да се споредат резултатите добиени со користење на вакви методи од надгледувано и ненадгледувано учење врз податоци за автомобили од 1985-та година за предвидување на нивната цена, одредување дали и колку автомобилот е безбеден за возење и групирање на автомобилите според нивните карактеристики и меѓусебни зависности. Според добиените резултати, може да се изведе генерален заклучок за тоа дали со текот на годините вредноста на автомобилите од 1985-та година се зголемува или опаѓа.

Вовед

Податочното рударство е интердисциплинарна област која спојува техники од компјутерските науки со статистика. Претставува процес на идентификација, односно, „рударење“ на интересни патерни и важни информации од големи податочни множества, по што го добива и своето име.
Користејќи низа на алатки и техники, како што се чистење на податоци, анализа на податоци, кластерирање, асоцијативно учење, класификација, регресија, како и машинско учење, оваа област обезбедува информации кои откриваат многу повеќе од она што веќе е познато. Во ерата на Big Data, кога податоците се движечка сила на светот, ова е од исклучителна важност. Оттука, е јасна потребата од податочното рударство и неговата широка примена во плејада области од секојдневниот живот.
Една ваква област е автомобилската индустрија. На високо ниво на апстракција, синџирот на вредности во автомобилската индустрија генерално може да се претстави со следните подпроцеси:

- Развој
- Набавки
- Производство
- Логистика
- Маркединг
- Продажба и услуга

Со примена на техниките на машинско учење и податочното рударство во автомобилската индустрија, може да се предвиди нејзината иднина од аспект на мобилност, автономност, конекција, електрификација, финансиска вредност и слично.

Опис на тема

Напредните технолошки решенија и иновации рапидно ја менуваат автомобилската индустрија во последната деценија. Традиционалните модели избледуваат со дигиталната револуција која се стреми кон реконструкција на целата софтверска и хардверска архитектура на возилата. За неколку години отсега, многу веројатно е автомобилите да не изгледаат како денес.

Иако широката употреба на „возила од иднината“, како што се електрични, автономни, самоуправувачки, конектирани возила е на повидок, побарувачката за ретки, класични и безвременски автомобили е во константен пораст. Класичните автомобили им дозволуваат на луѓето да уживаат искуство надвор од нивното време, овозможувајќи им пристап до иновациите и технологиите кои доаѓаат со денешницата. Како резултат на овие бенефиции, цената на некои стари автомобили денес е значително повисока од нивната оригинална цена во минатото.

Изработката на овој проект има за цел да обработи, визуелизира и моделира две податочни множества поврзани со карактеристики на автомобили произведени во 1985-та година. Едното податочно множество е преземено од Kaggle и содржи податоци од 1985 Ward’s Automotive Yearbook, а може да се погледне на следниот линк. Другото податочно множество е собрано со Web Crawling и содржи карактеристики на автомобили од 1985-та година кои се во продажба денес, а може да се погледне тука.
Употребени се соодветни техники за предпроцесирање за податоците да се доведат во соодветна форма која ќе ги задоволува барањата на користените модели. Исто така, податоците се претставени со дескриптивни и информативни визуелизации кои го прават идентификувањето на трендови, патерни и екстремни вредности полесно и поинтуитивно. Изведени се детекција и справување со мултиколинеарност, селекција на важни атрибути, редукција на димензионалниот простор и тренирање различни модели од надгледувано (линеарна регресија, класификација) и ненадгледувано (кластерирање) учење.

Содржината на оваа документација вклучува опис на секоја користена техника, како и објаснување на изведените заклучоци од добиените резултати. Користен е програмскиот јазик Python во работната околина Jupyter Notebook. Целосната имплементација може да се погледне на следниот линк.

Предпроцесирање

Предпроцесирањето е од фундаментално значење за секој процес на податочно рударство или машинско учење бидејќи директно влијае врз ратата на успех на проектот. Собраните податоци речиси секогаш се некомплетни, односно има вредности кои недостасуваат, содржат шум, екстремни вредности, дупликати или погрешни резултати. Доколку податочното множество врз кое се тренира некој модел содржи некоја од горенаведените карактеристики, може да предизвика грешки и да влијае деградирачки врз квалитетот на резултатите.
Во овој проект, применети се повеќе предпроцесирачки техники за справување со некомплетноста, празнините и грешките во податочните множества. Во продолжение, истите се етапно објаснети, почнувајќи од запознавање со податочното множество.

1. Запознавање со податочното множество

Податочното множество Automotive Dataset од Kaggle има 205 редици и 26 колони. Во продолжение следува објаснување на значењето на секој од атрибутите.

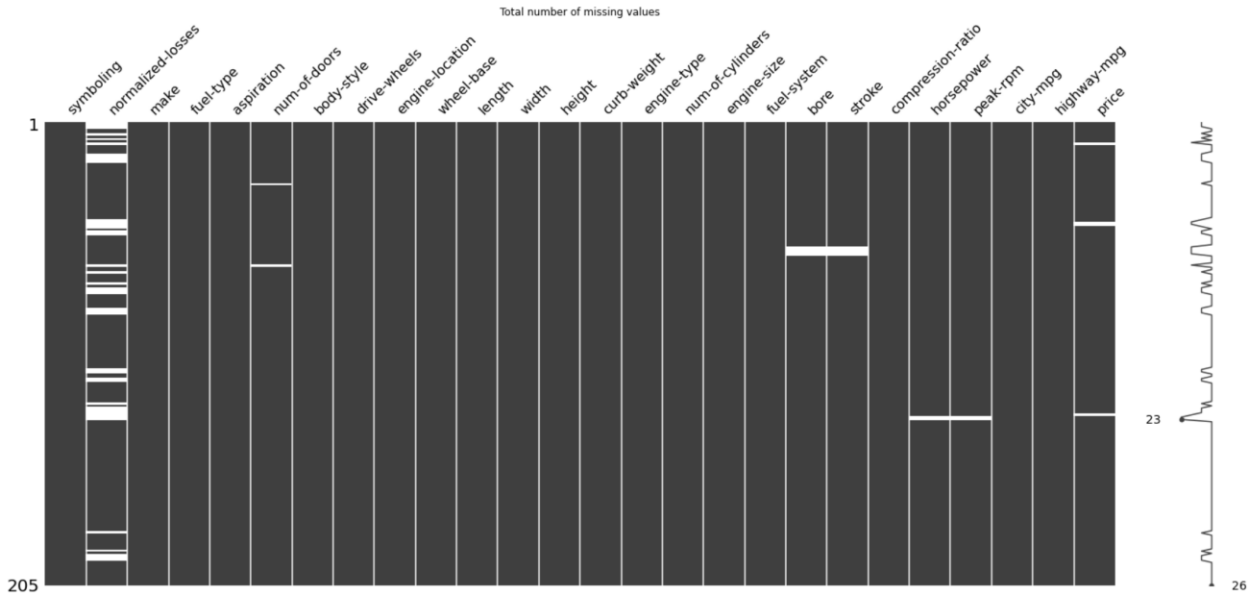
2. Missing values

2.1. Проверка за missing values

Со повикување на функцијата *info()* од софтверската библиотека *Pandas* во Python врз вчитаното податочно множество, на прв поглед изгледа како податочното множество да не содржи missing values. Што се однесува на типот на податоци, изгледа како 16 од атрибутите да се категориски, а 10 нумерички.

Но, доколку истото подетално се анализира, може да се забележи дека атрибутите **normalized-losses, num-of-doors, bore, stroke, horsepower, peak-rm** и **price** се всушност нумерички, но вредностите кои недостасуваат се означени со прашалник („?“). Поради ова, функцијата *info()* ги препознава како категориски променливи. Оттука, заклучуваме дека **11 атрибути се категориски**, а останатите **15 се нумерички** (целобројни или децимални вредности).

Еден начин за справување со ваквата состојба е првично симболите „?“ да се заменат со nan вредности од библиотеката *NumPy.*
Процентот на податоци кои недостасуваат на ниво на цело податочно множество е приближно **1.1%.** Поконкретно, во **26 редици** од вкупно 205 има вредности кои недостасуваат. Визуелно прикажано, тоа изгледа вака:



Слика 1. Missing values во податочното множество Automotive dataset

На *Слика 1.* Може да се забележи дека најголем број на missing-values има атрибутот normalized-losses.

2.2. Справување со missing values

Постојат различни начини на справување со податоци кои недостасуваат, во зависност од тоа дали податоците се категориски или нумерички. Еден ваков начин за нумерички атрибути е тие вредности да се заменат со **средната вредност** или со **медијаната**.

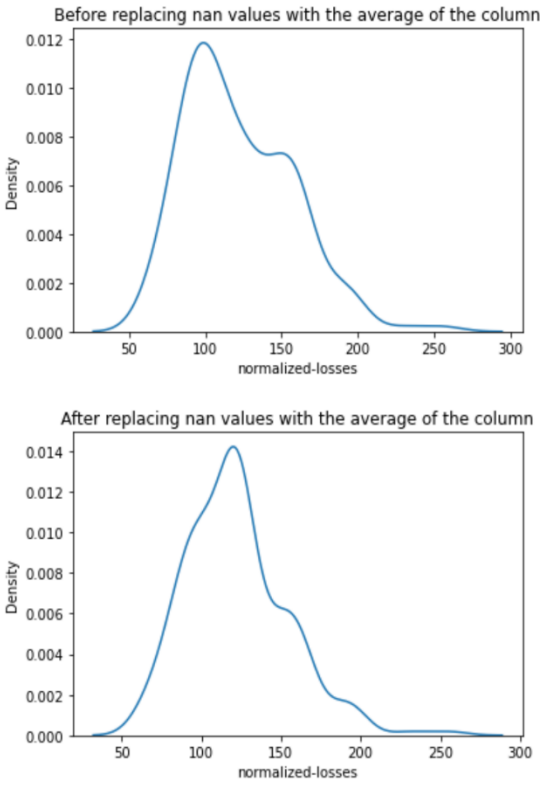
Се уште не знаеме дали во ова податочно множество има екстремни вредности. Она што го знаеме е дека **средната вредност е многу сензитивна на екстремни вредности, а медијаната останува незасегната**.

За да одлучиме со што да ги замениме податоците кои недостигаат за секој од атрибутите, прво ќе ги споредиме средната вредност и медијаната на секој од нумеричките атрибути кои содржат вакви вредности.

Доколку средната вредност и медијаната имаат **приближни вредности**, тоа значи дека податочното множество има **симетрична дистрибуција**, односно податоците се прилично балансирани или симетрични од двете страни.

Доколку пак **се разликуваат значително**, може да се донесе заклучок дека **податоците се наклонети** во некоја насока. Доколку има огромни екстремни вредности, дистрибуцијата на податоците е наклонета во насока на тие екстремни вредности. Ако **среднста вредност е помала од медијаната**, генерално, податоците се **налево наклонети**. Во обратен случај, податоците се **надесно наклонети**.

Кога имаат **приближна вредност** (во ова податочно множество, тоа важи за атрибутите **bore, stroke и normalized-losses**), не прави разлика дали вредностите кои недостасуваат ќе се заменат со средната вредност или медијаната. Затоа, за овие атрибути**, ќе замениме со средната вредност**. Пример за како оваа трансформација влијае врз дистрибуцијата е дистрибуцијата на атрибутот normalized-losses прикажана на *Слика 2*.



Слика 2. Дистрибуција на normalized-losses

Дистрибуцијата на атрибутот **horsepower** е **надесно наклонета**, а на атрибутот **peak-rpm** е **налево наклонета**, што значи дека е возможно да има екстремни вредности во податочното множество. Како последица на ова, за овие два атрибути, вредностите кои недостигаат **ќе ги заменима со медијаната**.

Податоците кои недостигаат за категорискиот атрибут **num-of-doors** ќе ги замениме со **модата** на овој атрибут, односно со неговата најфрекфентна вредност.

Друг начин за справување е **отстранување на редици** во кои има податоци кои недостигаат. Бидејќи атрибутот **price** содржи само 4 missing values, што е значително малку споредено со вкупниот број на редици во податочново множество, одлучивме овие редици целосно да ги отстраниме.

3. Проверка за уникатност и дупликати

Податочното множество Automotive dataset **не содржи дупликати**. Дополнително, приближно **18.71%** од податоците кои ги содржи се **уникатни**, а **најмногу уникатни** вредности има атрибутот **price**.

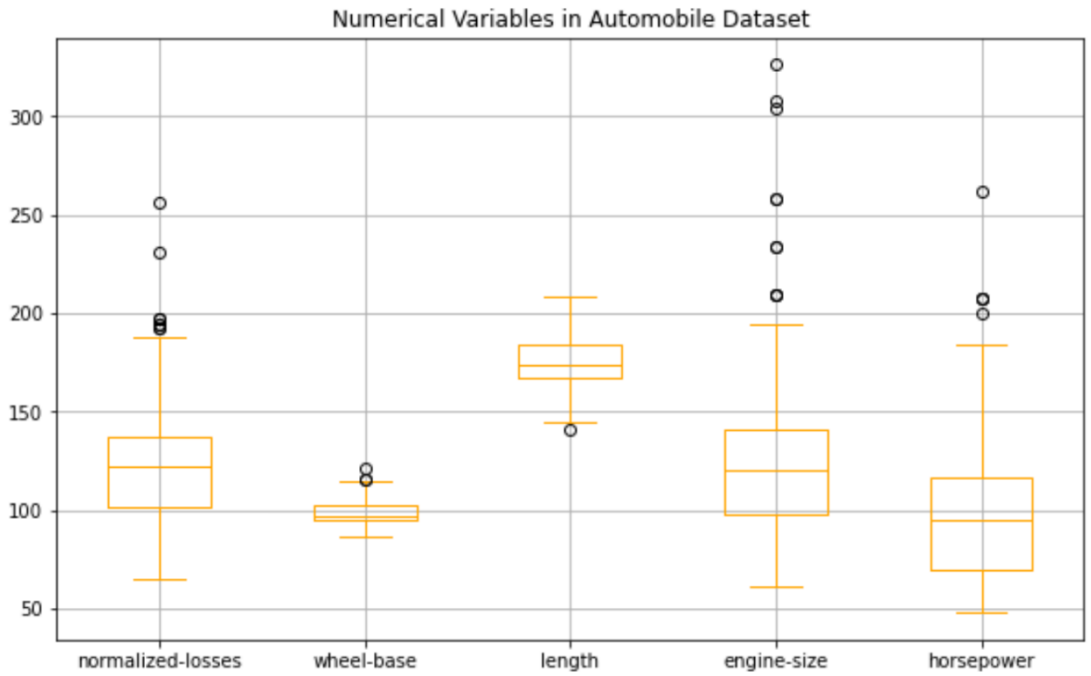
4. Екстремни вредности (Outliers)

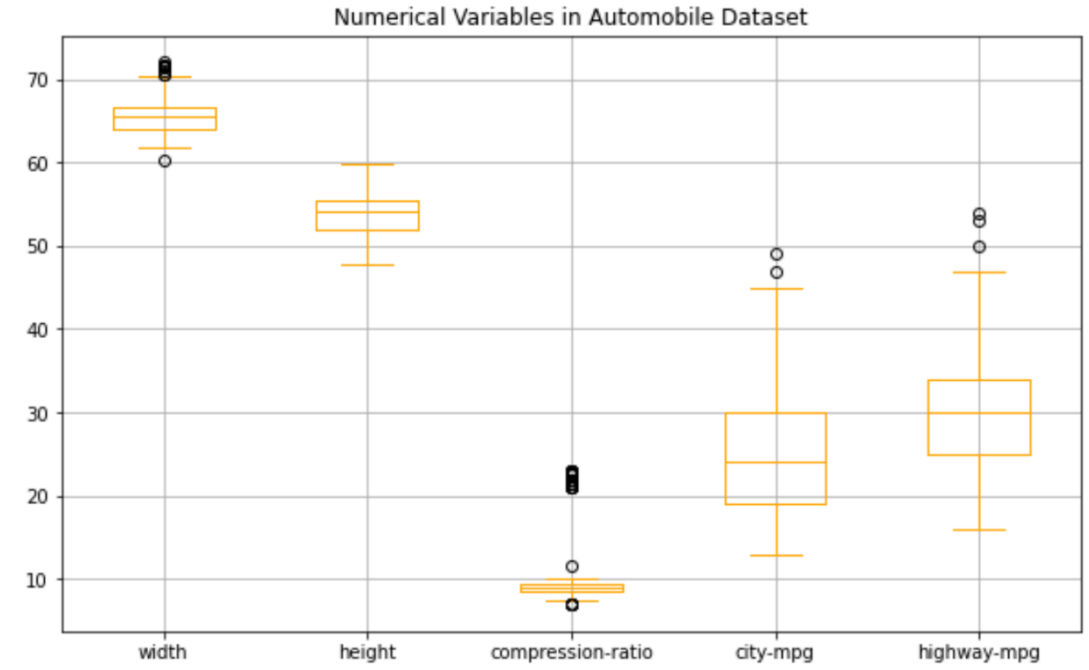
4.1. Проверка за екстремни вредности

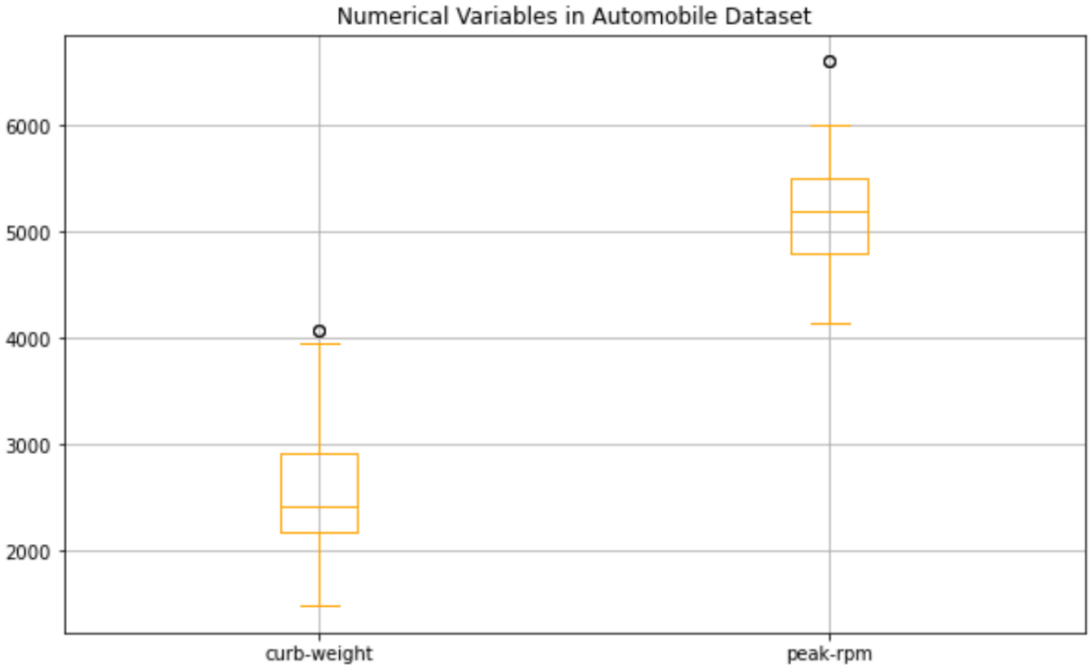
Доколку има значителна разлика помеѓу 75тиот перцентил и максималната вредност на секој од атрибутите, тогаш може да се претпостави дека тој атрибут содржи екстремни вредности кои треба подетално да се истражат.

Со повикување на функцијата *describe()* од библиотеката *Pandas*, се добиваат основни статистики како што се средна вредност, стандардна девијација, минимум, максимум, како и перцентилите на секој од нумеричките атрибути. За категориски атрибути, функцијата *describe(include='object')* не снабдува со информации околу бројот на уникатни вредности, како и најфрекфентната вредност на секој од атрибутите.

За подобра визуелизација, на три различни фигури ќе исцртаме boxplots за атрибути кои имаат сличен ранг на вредности. Од вака исцртаните boxplots на *Слика 3*. може да се воочи средната, минималната и максималната вредност за атрибутите, како и нивните квантили.







Слика 3. Визуелна претстава на екстремни вредности на нумеричките променливи

4.2. Справување со екстремни вредности

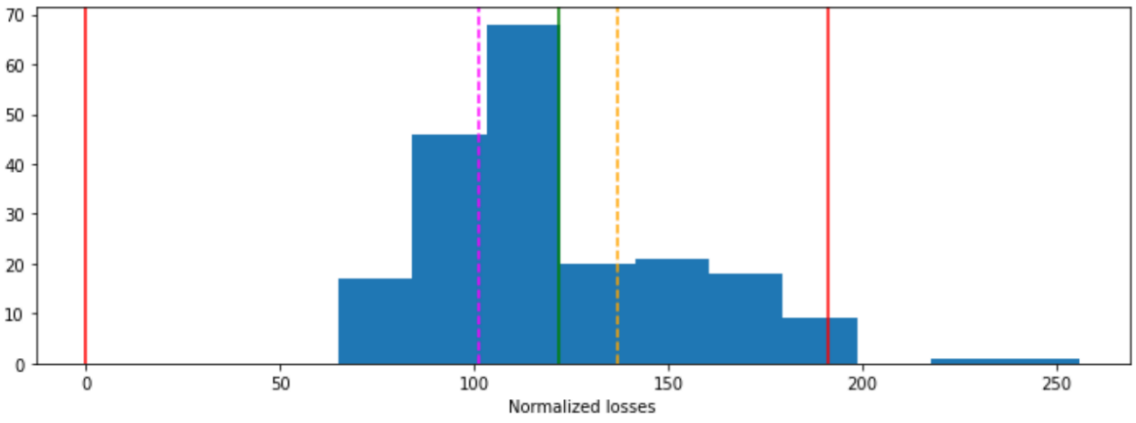
4.2.1. Детекција на редундантни примероци

Доколку примероците, односно редиците, се разликуваат по само еден или два атрибути, се отвора сомнеж дека тие се **дупликати** или се **редундантни**.
Во Automotive Dataset, ваков е случајот со атрибутите peak-rpm и curb-weight. Бидејќи бројот на вакви редундантни примероци е значително мал (по две редици за двата атрибути), овие редици **ќе ги отстраниме** од податочното множество сметајќи дека се дупликати.

4.2.2. Пресметка на интерквартален ранг

Атрибутот **normalized-losses** има 8 екстремни вредности. Ова го откривме со пресметка и визуелизација на интерквартилниот ранг на овој атрибут.

На **Слика 4**. Со **црвена линија** се обележани **долната** (Q25 -(1.5 * IQR)) и **горната** (Q75 +(1.5 * IQR)) **граница**. Со **розова** е обележан **25тиот** квантил, со **зелена** е обележан **50тиот**, а со **портокалова** е обележан **75тиот** квантил. Оние вредности кои се енадвор од долната и горната граница се сметаат за екстремни. На **Слика 4**. е забележително дека нема вредности надвор од долната, но има вредности надвор од горната граница. Наместо да ги отфрлиме, овие екстремни вредности **ќе ги замениме со средната вредност** на атрибутот (имајќи во предвид дека претходно во секција 2.2. воочивме дека средната вредност и медијаната имаат приближна вредност).



Слика 4. Интерквартален ранг на атрибутот normalized-losses

Екстремните вредности за атрибутите **city-mpg** ги оставивме како такви, бидејќи при истражување откривме дека просечното mpg во градско возење можело да

биде повисоко од 45 во 1985-та година.
Што се однесува на атрибутот **compression-ratio**, бројот на екстремни вредности е доста висок (20). Со истражување на интернет откривме дека дизел моторите користат поголем компресиски размер од оние на бензин, поради односот на компресијата која мора да ја зголеми температурата на воздухот во цилиндерот доволно за да го запали дизелот со помош на компресија. Односите на компресија често се помеѓу 14:1 и 23:1 за дизел мотори со директно вбризгување и помеѓу 18:1 и 23:1 за дизел мотори со индиректно вбризгување.

4.2.3. Имплементација на пристап од научен труд за детекција на екстремни вредности

Во оваа секција, објаснет е пристапот за детекција на екстремни вредности базиран врз основа на научниот труд со наслов: "A New Local Distance-Based Outlier Detection Approach for Scattered Real-World Data".
Детекција на екстремни вредности може да се спроведе преку едноставни, но моќни алгоритми. Четири вакви алгоритми се:

- **Kth Nearest Neighbors Distance (KNN):** Матрика за растојание која пресметува колку е оддалечена некоја точка од нејзиниот К-ти најблизок сосед. **Колку точката е поодалечена од К-тиот сосед, толку поверојатно е таа да е outlier.**

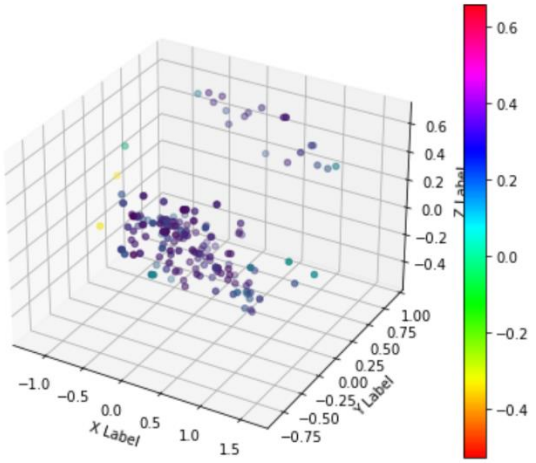
- **K Nearest Neighbors Total Distance (TNN):** Метрика за растојание која е просек од растојанијата до К-тиот најблизок сосед. Оваа метрика го решава проблемот на KNN, т.е. промашување на одредени екстремни вредности, кој настанува поради земањето на само еден сосед во предвид.

- **Local Distance-based outlier factor (LDoF):** Ова е алгоритам за одредување густина и растојание и е сличен на LoF, но наместо да се грижи за густината на соседството, пресметува колку некоја точка е оддалечена од центарот на соседството. Се пресметува како *TNN(x)/KNN_Inner_distance(KNN(x)).* Точка која има LDoF поголемо од 1 се наоѓа надвор од множеството на К најблиски соседи. Секоја точка со LDoF помало или приближно до 1 е обиколена од множеството К најбкиски соседи.

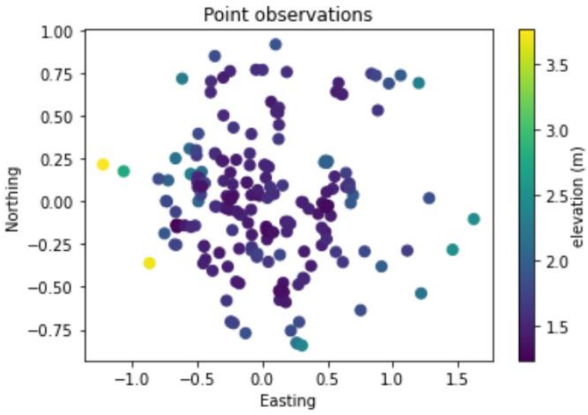
- **Local Outlier factor (LoF):** Метрика базирана на густина која одредува колку е густо, односно збиено „соседството“ на некоја точка. Соседството се одредува со помош на K-Nearest-Neighbors. Клучниот концепт на овој алгоритам е reachability_distance која се дефинира како: *reachability_distance(A, B) = max{distance(A,B), KthNN(B)}.*

Со други зборови, го претставува вистинското растојание меѓу A и B, но мора да биде најмалку еднакво на растојанието меѓу B и неговиот K-ти најблизок сосед. Со ова, reachability_distance е асиметрична метрика. Биејќи A и B имаат различно множество на K најблиски соседи, нивните растојанија до K-тиот сосед ќе се разликуваат.
Користејќи ја reachability_distance, може да се пресмета local_reach_density на густината на соседството на една точка. За некоја точка X, local_reach_density е 1 поделено со просекот на reachability_distance(x,y) за сите y во KNN(x), односно множеството на најблиските K соседи на x.
На овој начин, може да ја споредиме local_reach_density на x со таа на неговите соседи за да го добиеме LoF(x).

Клетвата на димензионалност (The curse of dimensionality) го прави KNN неефикасен во повивсок димензионален простор. Затоа, со помош на PCA податочното множество ќе го мапираме во дводимензионален и тридимензионален простор пред да продолжиме со применување на гореопишаниот пристап за детекција на екстремни вредности од овој научен труд.



Слика 5. Тридимензионален приказ на екстремни вредности базиран врз научен труд



Слика 6. Дводимензионален приказ на екстремни вредности базиран врз научен труд

На *Слика 5.* и *Слика 6.* со жолта боја се обележани екстремните вредности детектирани со овој научен пристап. На x-оска е претставена првата компонента, а на y-оска е претставена втората компонента од дводимензионалната репрезентација на податочното множество. На овој начин, овие екстремни вредности може да бидат лоцирани и отстранети.

Визуелизации и EDA

EDA или Exploratory Data Analysis е метод за анализирање и истражување на податочното множество со цел да се извлечат или сумаризираат неговите главни карактеристики. Често се користат методи и техники за визуелизација за да се откријат патерни, аномалии, да се тестираат претпоставки или хипотези преку различни графички репрезентации.

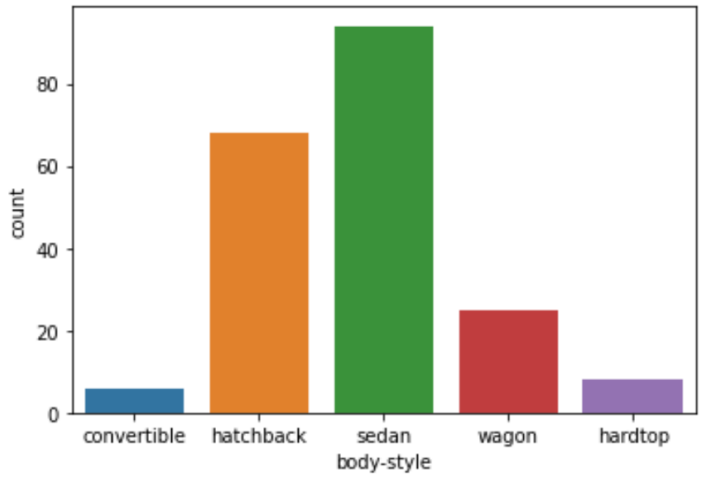
EDA излегува од рамките на формалното моделирање и хопотези за да овозможи максимален и длабински преглед на податочното множество и неговата структура, како и да ги идентификува највлијателните атрибути. Предпроцесирањето и EDA се слични и клучни фактори за истражувањата и проектите од областа на машинското учење и науката за податоци. Иако овие два термини се сродни бидејќи опфаќаат концепти кои се поклопуваат, битно е да се прави дистинкција помеѓу нив.

Во оваа секција, опишани се графички репрезентации и визуелизации со една и со повеќе променливи, нивната дистрибуција, меѓусебна зависност, колинеарност, монотоност, мултиколинеарност и селекција на најважните атрибути со различни техники.

1. Универијантни графички репрезентации

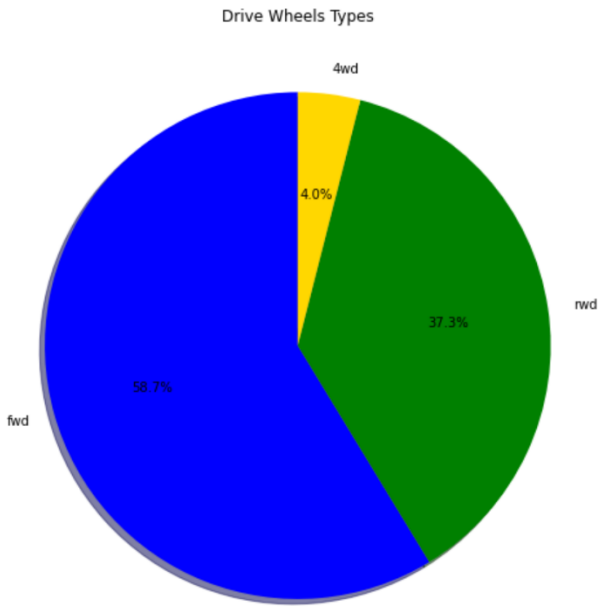
На визуелизациите прикажани подолу претставена е по една променлива од податочното множество Automobile Dataset.

На *Слика 7*. претставен е *countplot* од библиотеката за визуелизации во Python, *Seaborn* кој прикажува сооднос на променливата **body-style** во податочното множество. Може да се забележи дека **најголем број** од автомобилите во 1985-та се **sedan**, а **најмал број** се **convertible**.



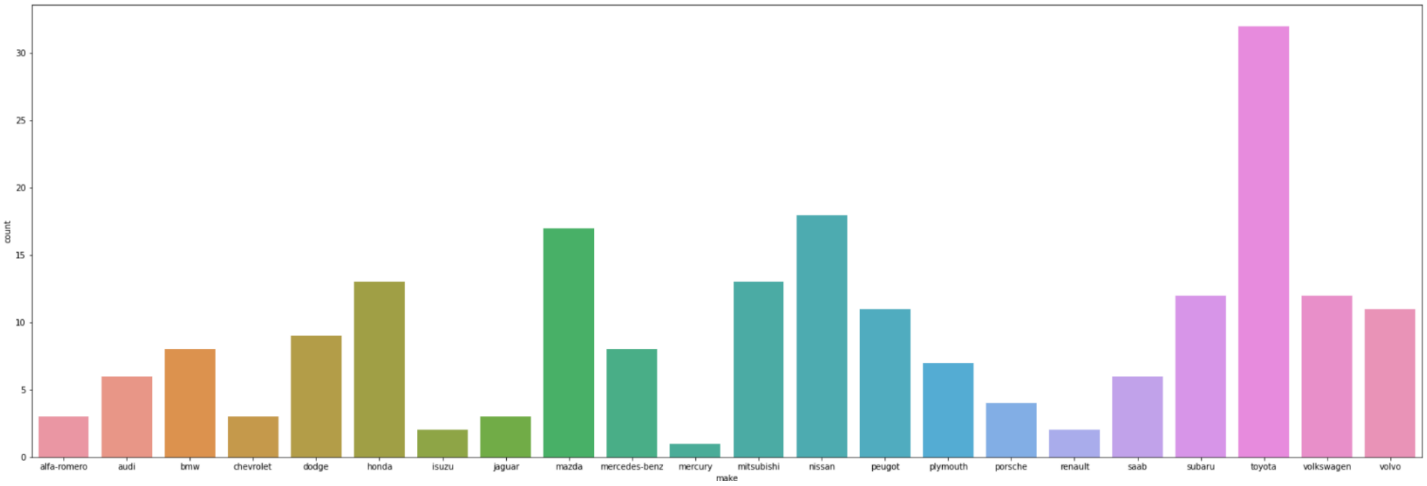
Слика 7. Визуелен приказ на атрибутот body-style

Понатаму, на *Слика 8*. прикажан е соодносот на различните типови на гуми на возилата (атрибутот **drive-wheels**) . Воочливо е дека **најголем број на возила имале fwd wheels (Fast Forward Drive)**, што значи дека силата од моторот се насочува кон предните тркала на автомобилот. **Средно застапени се rwd wheels (Rear-wheel Drive)** кај кои силата од моторот е насочена на задните тркала. Rwd тркалата биле најзастапени до крајот на дваесеттиот век. Од податочното множество, може да заклучиме дека во 1985-та година започнала помасовна употреба на fwd тркала, којашто трае и денес. **Најмалку застапени се 4wd wheels (Four-wheel drive 4x4)** кои овозможуваат симултана распределба на силата на моторот на сите четири тркала.



Слика 8. Визуелен приказ на атрибутот drive-wheels

На *Слика 9*. прикажана е застапеноста на различните брендови на автомобили во 1985-тата година (атрибутот **make**). **Најмногу возила** во податочното множество се од брендот **Toyota,** а пак **најмалку** се од брендот **Mercury.**

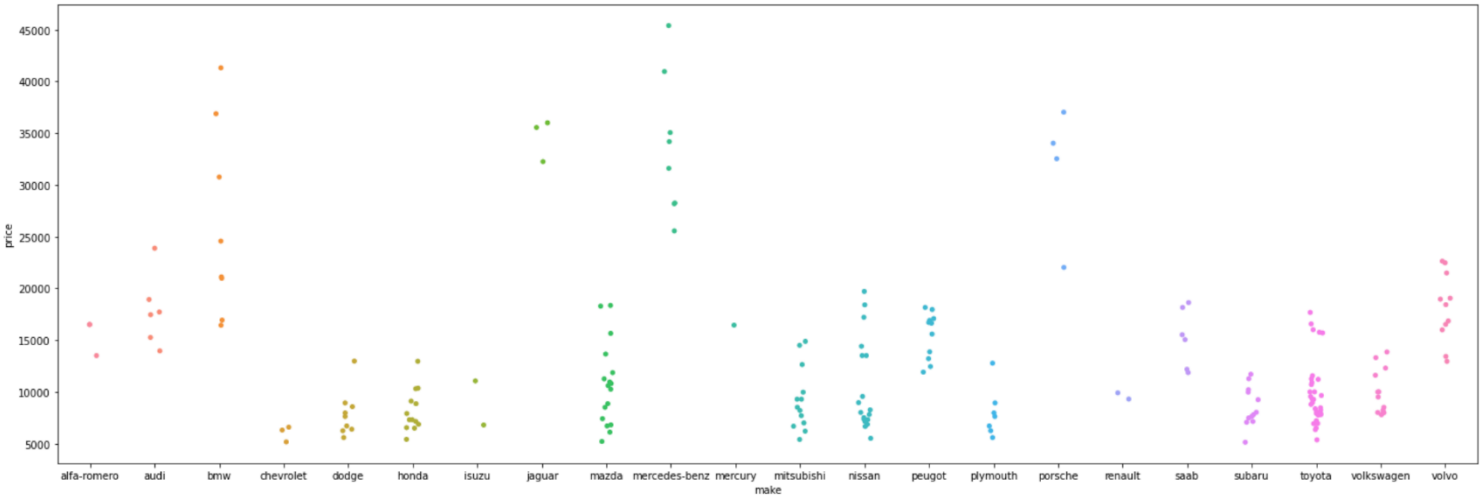


Слика 9. Визуелен приказ на атрибутот make

2. Мултиваријантни графички репрезентации

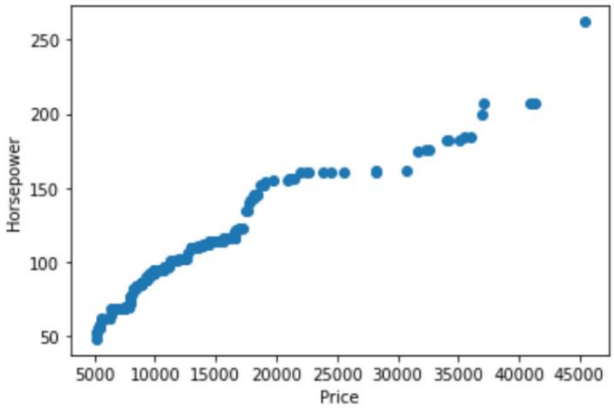
Во оваа секција, визуелно се прикажани меѓусебните врски и зависности на два или повеќе атрибути во податочното множество Automobile Dataset.

На *Слика 10*. прикажана е зависноста на цената на автомобилите во однос на брендот. Јасно забележливо е дека **највисока цена** имале автомобилите од брендот **Mercedes-Benz** која се движела од 25000 до 45000 долари. **Најниска цена** имале автомобилите од брендот **Chevrolet** која била пониска од 10000 долари.



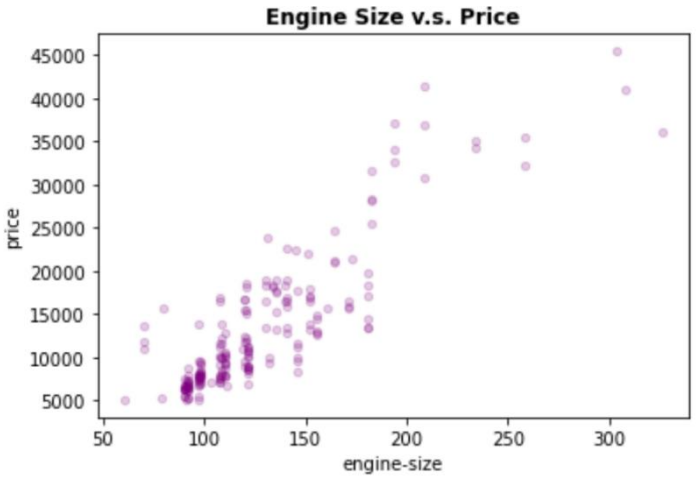
Слика 10. Приказ на атрибутот price наспроти атрибутот make

На *Слика 11.* Прикажана е линеарната зависност помеѓу цената и коњските сили на автомобилите. Јасна е корелацијата меѓу овие два атрибути, т.е. може да се заклучи дека **колку повеќе коњски сили има автомобилот, толку повисока му е цената**.



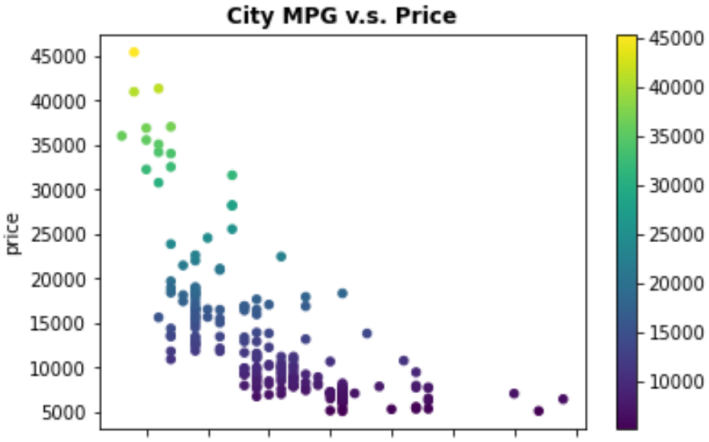
Слика 11. Приказ на атрибутот price наспроти атрибутот horsepower

Понатаму, слично како со атрибутот horsepower, воочлива е линеарна и монотона зависност помеѓу големината на моторот (атрибутот engine-size) и цената на автомобилот.



Слика 12. Приказ на атрибутот price наспроти атрибутот engine size

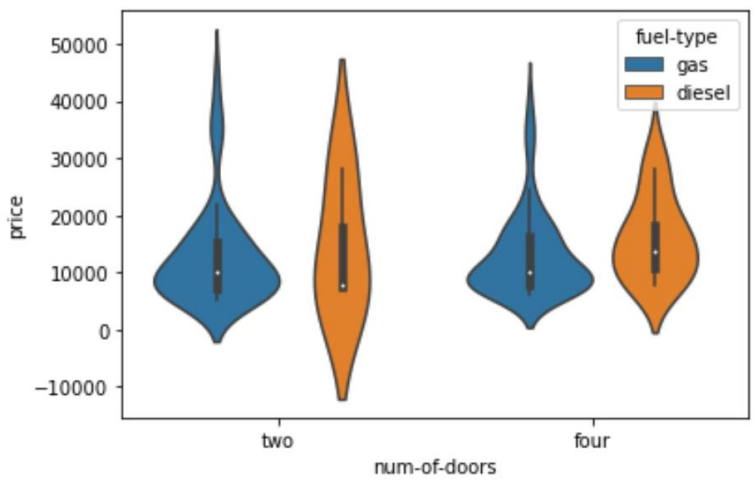
Како последица, **може да претпоставиме корелација меѓу атрибутите horsepower и engine-size.** Поконкретно, **колку е поголем моторот, толку е повисока цената**. Ова е генерален заклучок и го занемарува случајот каде за поголема вредност од 300 на атрибутот engine-size, цената е пониска отколку за вредност еднаква на 300. Ваквата зависност е прикажана на *Слика 12*.



Слика 13. Приказ на атрибутот price наспроти атрибутот city-mpg

Обратно-пропорционално се зависни атрибутите city-mpg и price. **Цената е најниска за автомобил кој има најмала градска потрошувачка**. Колку е помала потрошувачката на автомобилот во градски услови, толку автомобилот е повреден. Ваквата силна корелација може да се забележи на *Слика 13*.

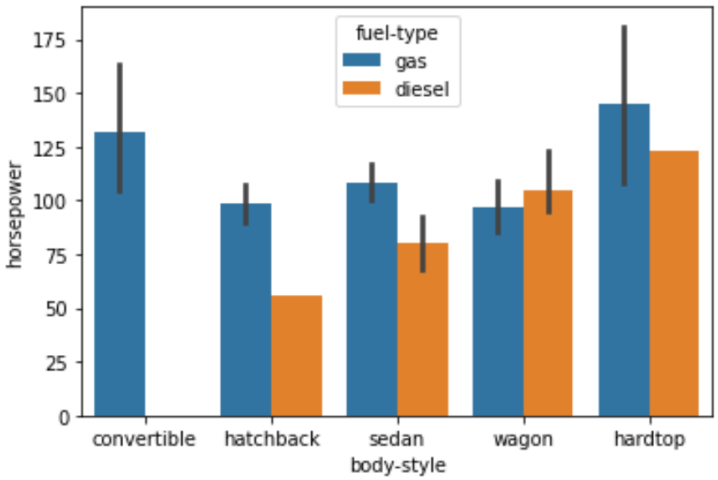
Violin plot е метод за графички приказ на дистрибуцијата на нумеричките податоци. Сличен е на boxplot, со таа разлика што дава дополнителни информации за густината на податоците за различни вредности. Еден недостаток на овој метод е тоа што иако претставениот атрибут нема негативни вредности, violin plot се проширува под нулата, до минус бескрај. Ова е проблематично бидејќи логаритмот е недефиниран за негативни броеви. На *Слика 14*. прикажана е **цената на автомобилите во зависност од типот на гориво (fuel-type) и бројот на врати на автомобилот (num-of-doors).** Иако средните



Слика 14. Атрибутите price и num-of-doors наспроти атрибутот fuel-type

вредности се приближно еднакви, дистрибуциите се прилично различни. Поретки се дизел автомобили со две врати, а **најчести се бензин автомобили со две или четири врати и со цена приближно 10000 долари.** Истенчените делови на оваа визуелизација (пример: бензин автомобили со две врати и цена во рангот од 30000 до 50000 долари) се ретки и може да се сметаат за екстремни вредности

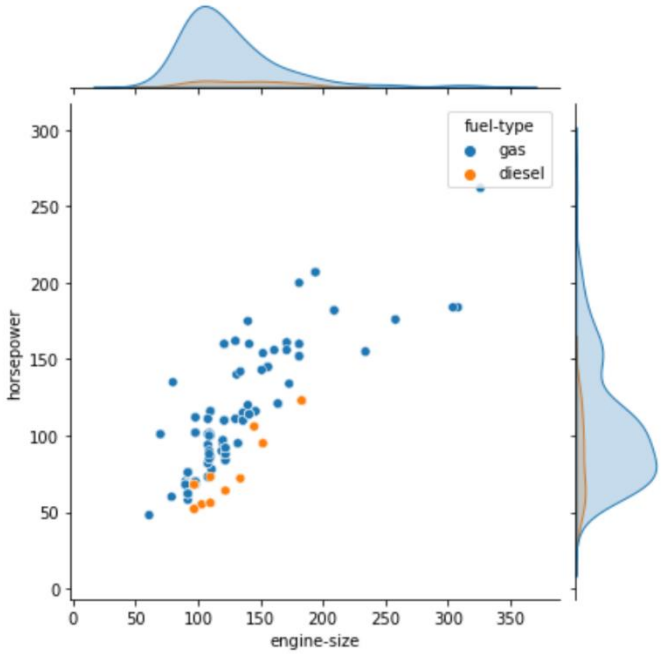
Ако и понатаму го истражуваме атрибутот fuel-type ќе откриеме дека **автомобили со wagon body style и <=100 коњски сили се почесто дизел**. Во останатите случаи, за автомобили со body style convertible, hatchback, sedan и hardtop, автомобилите се почесто на бензин. Во податочното множество **нема дизел автомобили со стил convertible**. Уште повеќе, може да се забележи дека **автомобили со повеќе од приближно 125 коњски сили не може да бидат дизел.** Овие заклучоци се изнесени од *Слика 15.*



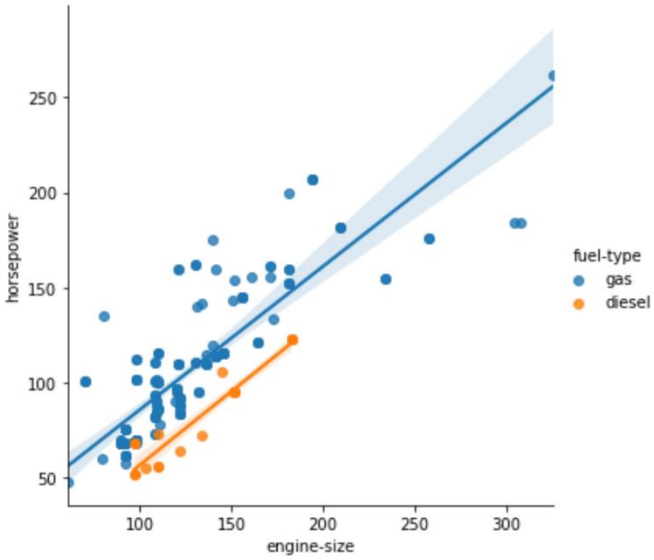
Слика 15. Атрибуите horsepower и body-style наспроти атрибутот fuel-type

Автомобилите може да бидат **дизел доколку имаат од 50 до приближно 125 коњски сили**. Надвор од овој ранг, автомобилите во 1985-тата биле на бензин. Ова се воочува од *Слика 16.* и *Слика 17.* Познато е дека конзумирањето на гориво се зголемува со зголемувањето на коњските сили на автомобилите. Автомобили кои конзумираат повеќе гориво, повеќе ја загадуваат животната средина. Оттука, во 1985-тата, **дизел возилата биле поеколошки.** Дополнително, интервалот на доверба на бензин автомобилите е значително поширок од оној на дизел автомобилите (Слика 17.). **Интервалот на доверба**

почнува да се шири за бензин автомобили со повеќе од 150 коњски сили и големина на мотор поголема од 200, што значи дека предвидувањата кои би ги правеле, би биле со поголема несигурност. Ова може да биде **резултат на недостиг на примероци кои имаат коњски сили во рангот од 200 до 250**.

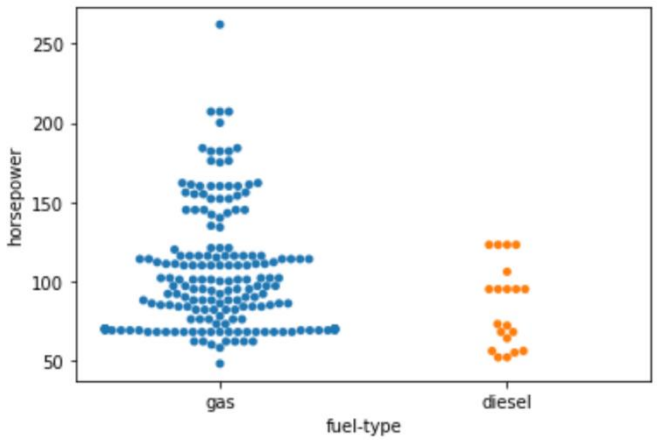


Слика 16. Атрибуите horsepower и body-style наспроти атрибутот fuel-type



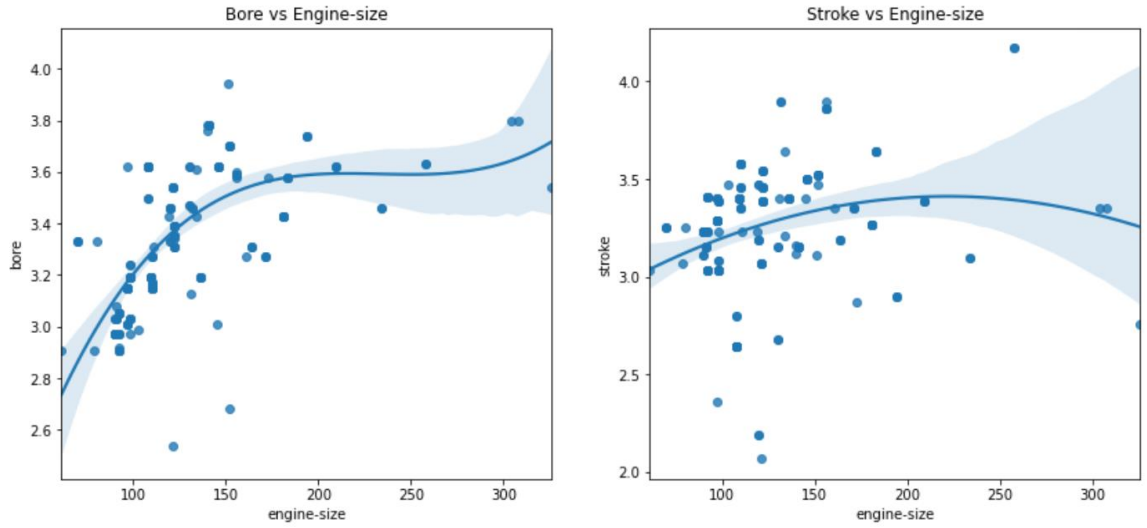
Слика 17. Слика 16. Атрибуите horsepower и body-style наспроти атрибутот fuel-type со интервал на доверба

Во 1985-тата година, **најчести биле бензин автомобили кои имале приближно 75 коњски сили**. За бензин автомобили со повеќе од 250 коњски сили, може да се каже дека се екстремни вредности (*Слика 18.)*

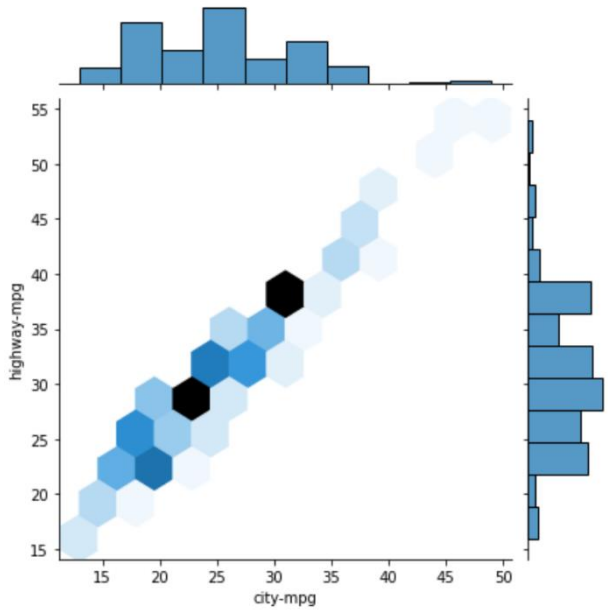


Слика 18. Визуелен приказ на атрибутот fuel-type наспроти атрибутот horsepower

Слика 19. Визуелно ја претставува зависноста на големината на моторот на автомобилите и атрибутите bore и stroke, заедно со интервалите на доверба. Она што привлекува внимание е тоа што атрибутот bore има помонотона зависност со големината на моторот отколку атрибутот stroke. Како и на *Слика 17*., и тука интервалот на доверба се шири за автомобили со големина на мотор поголема од 200, повторно најверојатно поради недостатокот на примероци. Интервалот на доверба на атрибутот stroke е доста поширок од оној на bore. Бидејќи bore го претставува дијаметарот на цилиндрите на моторот, а stroke ја претставува нивната длабочина, претпоставуваме корелација помеѓу овие два атрибути со големината на моторот.



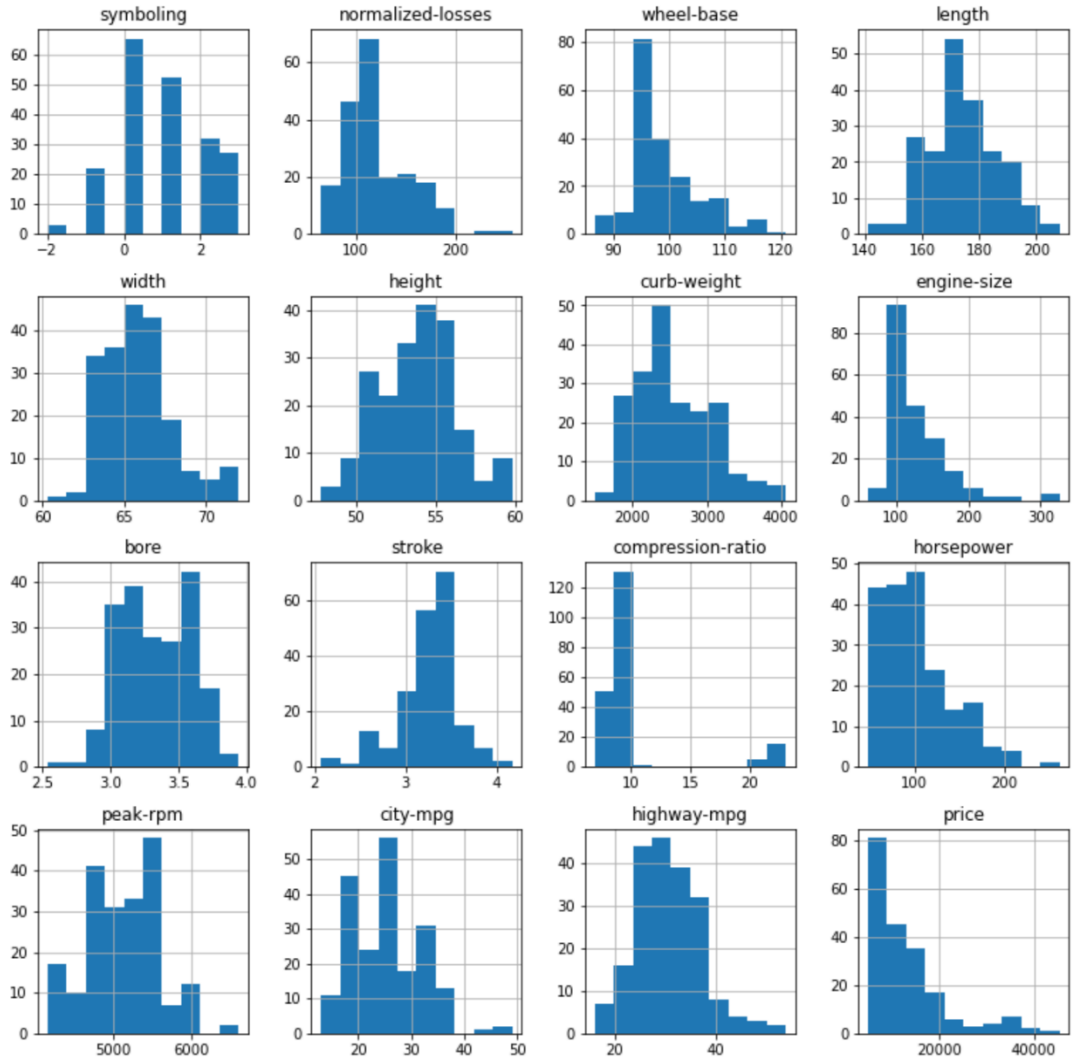
Слика 19. Визуелизација за зависноста помеѓу атрибутите bore, stroke и engine-size



Слика 20. Приказ на монотона зависност помеѓу атрибутите city-mpg и highway-mpg

Пример за **силна монотона зависност** е онаа помеѓу атрибутите city-mpg и highway-mpg од којашто може да се заклучи дека ако автомобилот товеќе троши кога се вози низ град, повеќе ќе троши и кога се вози на автопат и обратно. На *Слика 20*. забележуваме дека нзјчести се автомобилите со вредности за city-mpg во рангот [~ 20, ~ 25] и вредности за highway-mpg во рангот [~ 25, ~ 30] , како и автомобили со вредности за city-mpg во рангот [~ 28, ~ 32] и вредности за highway-mpg во рангот [~ 35, ~ 40]. Односно најчести се оние вредности за кои шестаголникот за *Слика 20*. е најтемен.

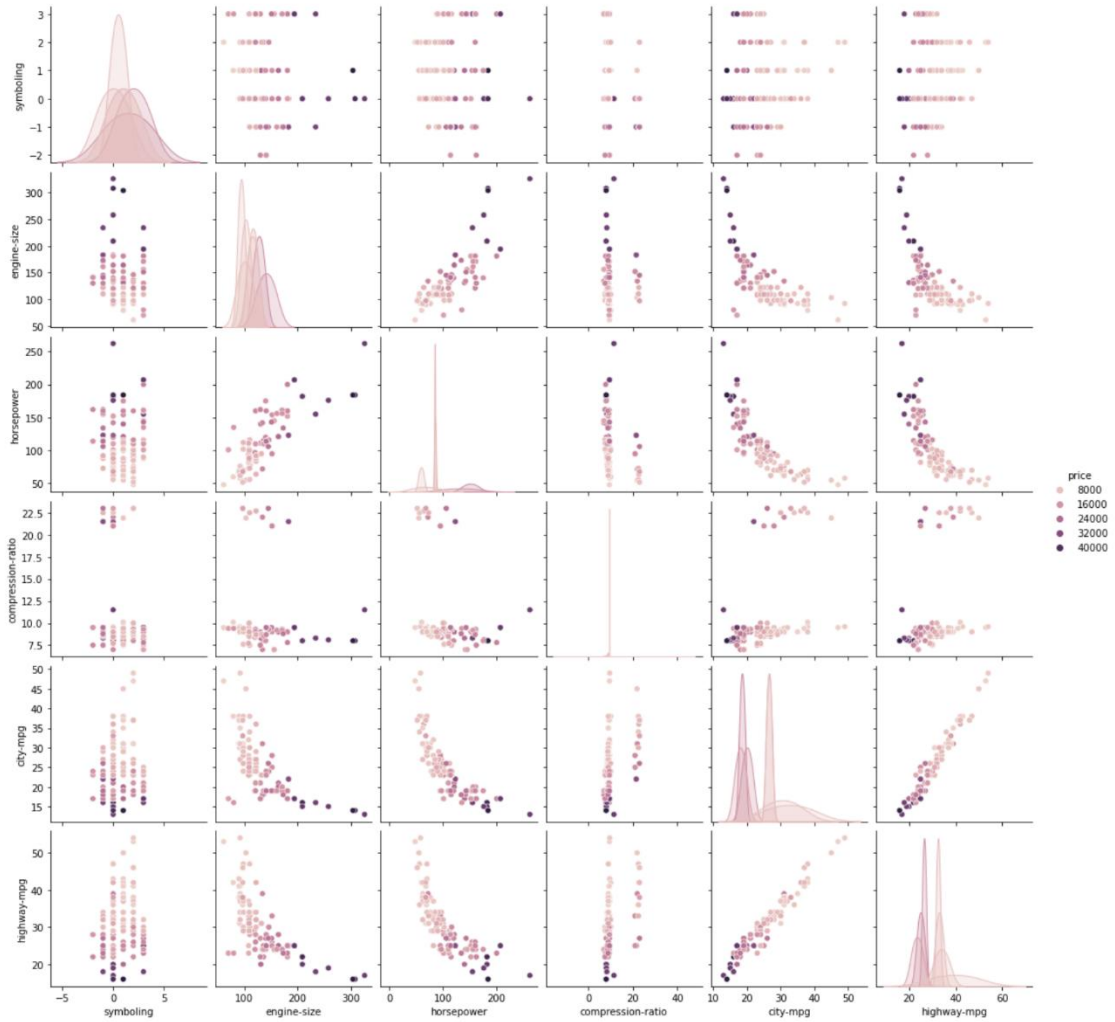
Од хистограмот на *Слика 21*. може да се анализира дистрибуцијата на секој од нумеричките атрибутите поединечно. Забележуваме дека некои од нив имаат налево или надесно **наклонети дистрибуции**, како што се дистрибуциите на атрибутите price, horsepower и engine-size. Во понатамошната работа со ова податочно множество, кога ќе тренираме модел на линеарна регресија за предикција на цената на автомобилите, наклонетоста на цената како зависна варијабла може да има **негативно влијание врз моделот за предикција**. Лошата предикција би била последица на тоа што цената на најголем број автомобили е концентрирана на околу 10000 долари. Како резултат, моделот најчесто ќе предвидува вредност приближна на 10000 долари и средната квадратна грешка би била голема. Постојат различни начини за справување со наклонетоста на зависната променлива кои го подобруваат квалитетот на предикциониот модел.



Слика 21. Поединечна дистрибуција на нумеричките атрибути

На *jointplot* фигурата (*Слика 22*.) исцртана со библиотеката за визуелизации *Seaborn*, може да се види зависноста на цената од атрибутите symbolling, engine-size, horsepower, compression-ratio, city-mpg и highway-mpg, како и нивната меѓусебна зависност. Очигледна е **силна линеарна зависност меѓу city-mpg и highway-mpg и помеѓу атрибутите engine-size и horsepower**. Од оваа визуелизација може да се извлечат многу интересни заклучоци, а некои од нив веќе беа опфатени од претходните визуелизации. Така, автомобилите се најскапи ако:

- Имаат ниски вредности за city-mpg и highway-mpg
- Имаат ниска вредност за city-mpg или highway-mpg, а висока вредност за engine-size
- Имаат високи вредности за horsepower и engine-size
- Имаат ниска вредност на city-mpg и вредност меѓу 0 и 20 за compression-ratio
- Имаат ниска вредност за compression-ratio, а висока вредност за engine-size

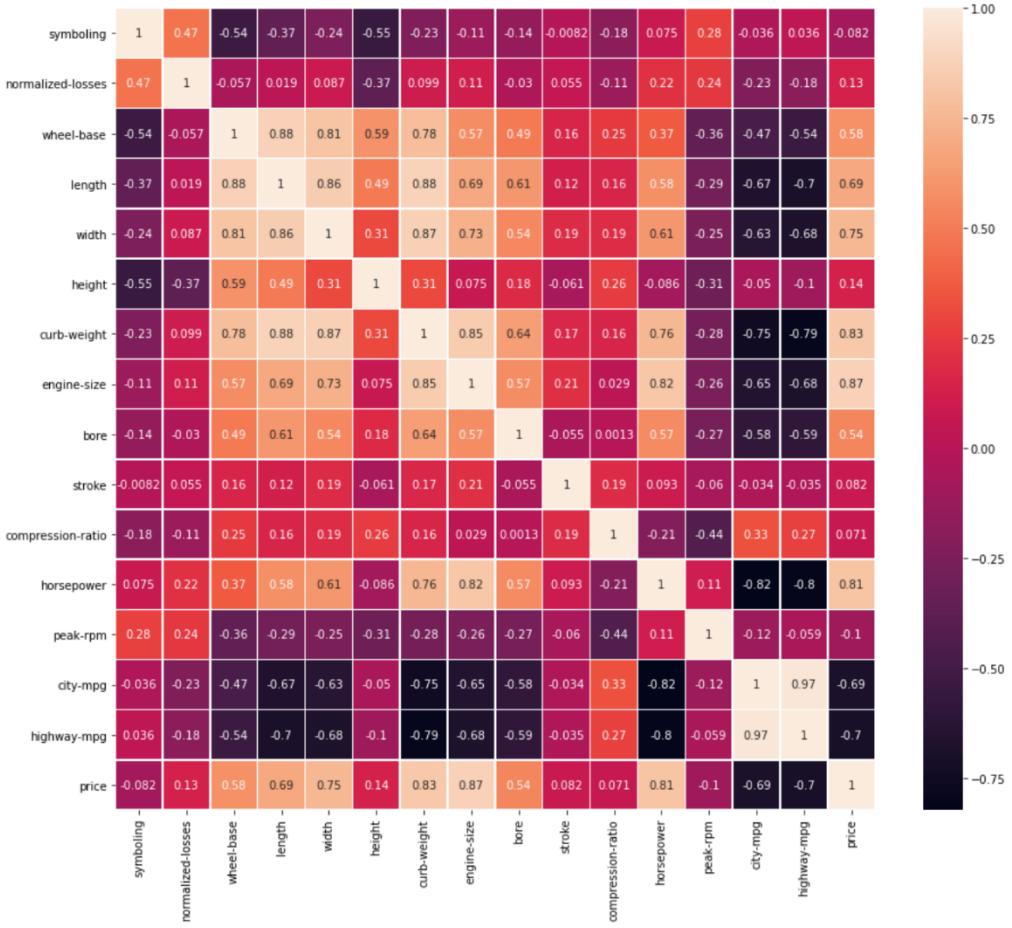


Слика 22. Меѓусебна зависност на цената со атрибутите symbolling, engine-size, horsepower, compression-ratio, city-mpg и highway-mpg

3. Пирсонов коефициент на корелација на нумеричките атрибути

Пирсоновиот коефициент на корелација е **статистички тест кој ја мери магнитудата на асоцијација, односно корелацијата помеѓу две континуални променливи**, како и насоката на нивната врска. Ако Пирсоновиот коефициент изнесува **-1**, тогаш двата атрибути имаат **совршена негативна линеарна зависност**. Ако изнесува **1**, имаат **совршена позитивна линеарна зависност.** Доколку пак изнесува **0**, атрибутите **не се линеарно зависни.** Битно е да се забележи дека доколку пирсоновиот коефициент изнесува 0, **не може да се заклучи дека атрибутите се независни**. Односно, ако коефициент еднаков на 0 означува дека помеѓу атрибутите не постои линеарна зависност, но може да постои нелинеарна зависност.

Освен наклонетоска на дистрибуцијата на атрибутите, негативно влијание врз квалитетот и интерпретацијата на еден модел има и мултиколинеарноста. **Мултиколинеарноста се случува кога предикторите на зависната променлива се високо корелирани меѓу себе**. Ова значи дека некој од предикторите може да биде изведен преку останатите предиктори и само додава комплексност на предикциониот модел. При тренирање на модели, оптимално е предикторите да се малку или воопшто да не се корелирани помеѓу себе, а да бидат силно корелирани со зависната променлива.

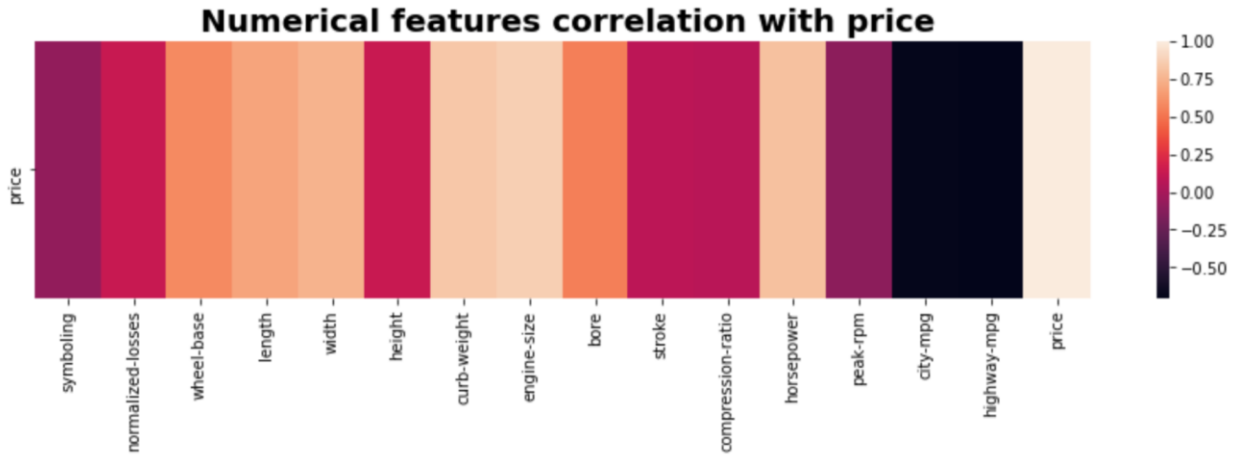


Слика 22. Пирсонов тест за корелација

Пирсоновиот тест го пресметува коефициентот на линеарна зависност помеѓу два атрибути во изолација. Сепак, доколку коефициентот на корелација помеѓу предикторите е блиску до -1 или 1, може да претпоставиме дека во податочното множество постои мултиколинеарност која понатаму треба да се детектира.

На heatmap-ата на *Слика 22.* гледаме дека силна негативна или позитивна линеарна корелација со зависната променлива price имаат атрибутите city-mpg, highway-mpg, horsepower, engine-size, curb-weight, width, length. Очигледно е дека во ова податочно множество има огромна мултиколинеарност. Освен што предикторите се корелирани со зависната променлива, тие се и меѓусебно корелирани. На пример, постои силна линеарна корелација помеѓу highway-mpg и length, highway-mpg и width, highway-mpg и curb-weight, highway-mpg и engine-size и т.н.

На *Слика 23.* се прикажани предикторите на зависната варијабла price која подоцна ќе сакаме да ја предвидиме со тренирање на линеарен модел. **Најсилни предиктори се engine-size, city-mpg и highway-mpg.**



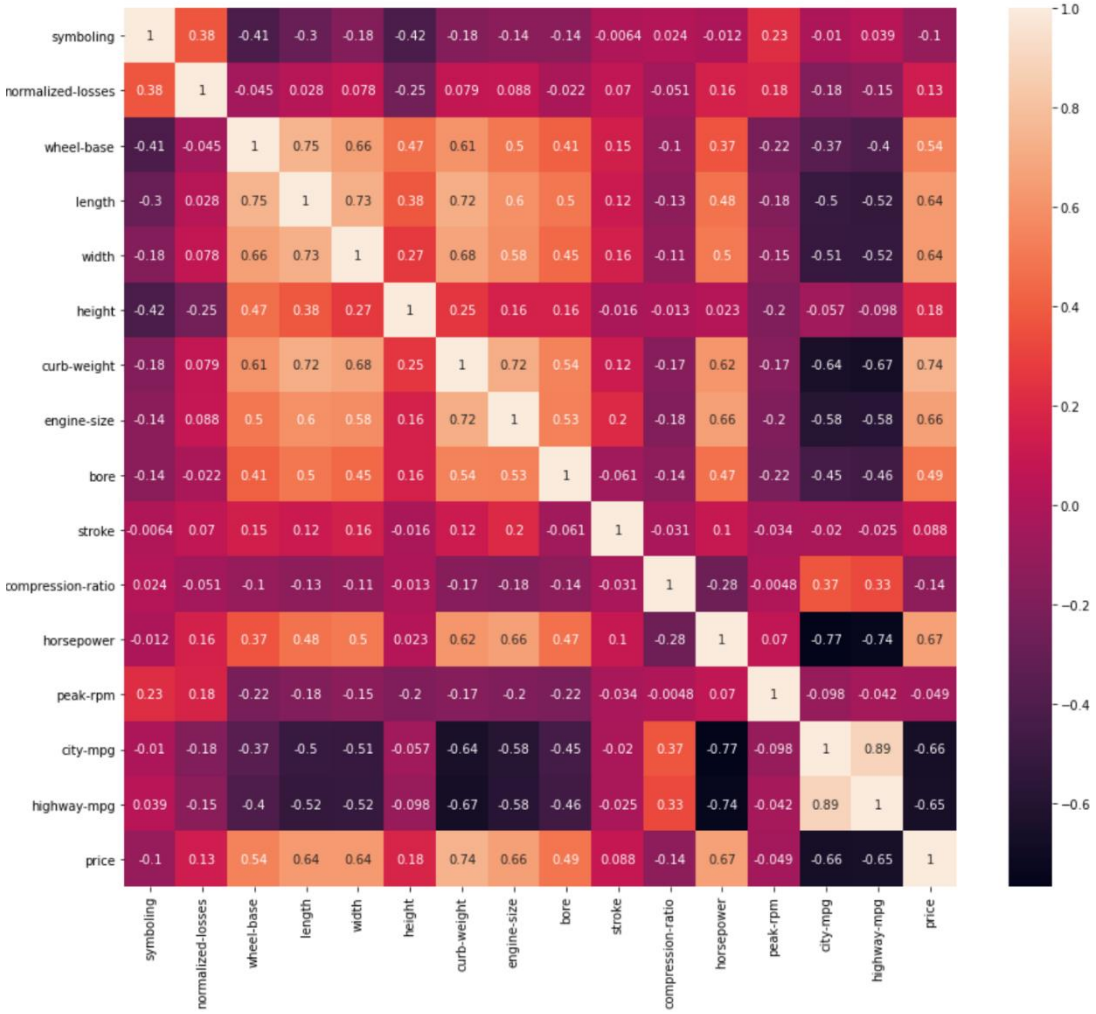
Слика 23. Предиктори на атрибутот price

4. Кендалов коефициент на корелација

Како што е опишано во точка 3, корелацијата е биваријантна анализа која ја мери јачината на асоцијацијата помеѓу два атрибути и насоката на нивната врска. За разлика од Пирсоновиот коефициент на корелација, Спермановиот и Кендаловиот коефициент се непараметарски тестови базирани на статистичкиот T-test кои мерат ординална, односно монотона зависност помеѓу пар атрибути. Врската меѓу два атрибути е монотона доколку само расте или само опаѓа.

И двата коефициенти, Сперманов и Кендалов, се алтернатива на Пирсоновиот коефициент на корелација. Сепак, Кендаловиот коефициент почесто се практикува бидејќи е поробусен и поефективен.

На *Слика 24*. е претставен Кендаловиот коефициент на атрибутите во податочното множество Automotive Dataset.



Слика 24. Кендалов коефициент на корелација

Од *Слика 24*. може да заклучиме дека **најсилна позитивна монотона зависност имаат атрибутите city-mpg и highway-mpg**, а **најсилна негативна монотона зависност имаат атрибутите city-mpg и horsepower**.

5. Мултиколинеарност

Мултиколинеарноста е појава на **високи интеркорелациони врски помеѓу два или повеќе независни променливи во мултиваријантен регресионен модел**. Генерано, мултиколинеарноста води кон **поширoки интервали на доверба**, поради што предвидените вредности се помалку релевантни.

Една од претпоставките на Линеарната Регресија е тоа дека предикторите треба да имаат малку или воопшто да немаат мултиколинеарност, за статистичкиот модел да биде робусен и ефикасен.

Има повеќе начини за детекција и справување со мултиколинеарност, меѓу кои справување со помош на **PCA**, со пресметка на **VIF** (Variance Inflation Factor) или пак со тренирање на регуларизирачки линеарен модел, како што се **Lasso** и **Ridge**.

Важно е да се има во предвид дека мултиколинеарноста повеќе влијае врз интерпретабилноста на моделот отколку на предиктивната моќ. Доколку моделот кој се тренира служи исклучиво за предикција, справувањето со мултиколинеарност не е задолжително и нема да влијае врз предикцијата се додека се задржиме во истиот ранг на вредности.

5.1. Имплементација на пристап од научен труд за справување со мултиколинеарност со помош на PCA

Како последица на мултиколинеарност, оценетите коефициенти може да се неточни и непрецизни. Кога во податочното множество има вакви статистичко-аналитички проблеми, **Principal Components Analysis** е една од мерките за справување со нив. Користи манипулација и анализа на матрици на податоци за да ги намали коваријансните димензии, максимизирајќи го количеството на варијација.

PCA е **алатка за редукција на димензионалноста која се користи за да се намали големо множество на корелирани предиктори во помало, помалку корелирано множество наречено principal components, во коешто се содржат најважните информации од големото множество.**
Првата компонента содржи најмногу варијабилност (колку што е возможно), а втората компонента е одговорна за остатокот од ваијабилноста.

Во продолжение следи објаснување на начинот на кој PCA се користи за справување со мултиколинеарност, базиран на научниот труд со наслов: „Principal Component Analysis to Address Multicollinearity“.

Првично потребно е да ги **енкодираме категориските променливи** во Automotive Dataset во нумерички. Со помош на *LabelEncoder* од пакетот *sklearn.preprocessing*, секоја вредност од категориските атрибути се енкодира во вредност од 1 до бројот на различни вредности кои ги има тој атрибут.

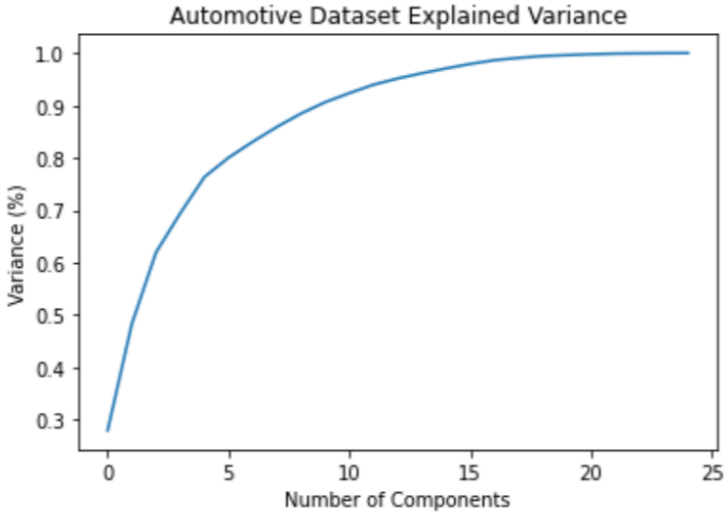
За проценка на параметрите, се тренира **Ordinary Least Squares (OLS)** модел. Како што самото име сугерира, OLS моделот наоѓа параметри со кои се минимизира средната квадратна грешка на резидуалите. Моделот **има за цел да ја предвиди цената** на автомобилите во однос на другите атрибути.

При преглед на сумаризацијата на OLS моделот, првично може да се воочи дека вредностите на **R2** и **Adjusted R2** се одлични **(~ 0.97),** што не значи дека моделот за предикција е добар. Ова е последица на проблемот кој настанува со овие метрики за евалуација, односно, **колку повеќе атрибути се додаваат во податочното множество, толку повеќе R2 и Adjusted R2 имаат тенденција да растат, иако можеби атрибутите кои се додаваат не се добри предиктори на зависната променлива.** Дека моделот не е добар потврдува и забелешка која предупредува за голема мултиколинеарност во податочното множество:

```
[3] The condition number is large, 1.74e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Идејата зад PCA е да се намали мултиколинеарноста во целокупното податочно множество. Но пред да се примени оваа техника, потребно е вредностите **да се скалираат**. Еден начин да се направи ова, е да се примени *MinMaxScaler* од пакетот *sklearn.preprocessing* којшто ги скалира податоците во ранг од 0 до 1.или пак ако има негативни вредности во податочното множество, од -1 до 1.

На почеток, **PCA се применува без да се специфицира бројот на комоненти** на кои ќе се сведи множеството по редукцијата на димензионалноста. Со ова, овозможен ни е преглед на низа која содржи количество варијанси на податочното множество кои секој од атрибутите ги објаснува и која може визуелно да се прикаже. Од *Слика 25*. се забележува **дека 10 од 25 атрибути** (не ја вклучуваме зависната променлива) **објаснуваат повеќе од 91% од варијансата во податочното множество.** Останатите 15 атрибути, додаваат до 100% објаснување на варијансата.



Слика 25. Количество објаснета варијанса од секој атрибут

Низата *explained_variance_ratio_* која ја овозможува самата функција на *PCA* од пакетот *sklearn.decomposition*, не снабдува со информации за тоа колкава варијанса објаснува секој од атрибутите. Битно е да се напомене дека во овој момент, не се знае точно кој атрибут колкава варијанса објаснува, бидејќи по примената на PCA врз податочното множество, се губи интерпретабилноста на атрибутите.

Само **16 атрибути од 25 објаснуваат приближно 98% од варијансата во целото податочно множество**. Оттука, ја редуцираме димензионалноста на Automotive Dataset на **16 principal components**.

Доколку повторно се истренира OLS модел, забележливо е дека **иако значајно се намалиле вредностите на R2 и R2 Adjusted, веќе нема предупредување за голема мултиколинеарност, што значи дека успешно сме се справиле со истата**. Сепак, евалуациските метрики MSE (12889404.43) и MAE (2383.15) се прилично големи, а причината за ова треба дополнително да се истражи.

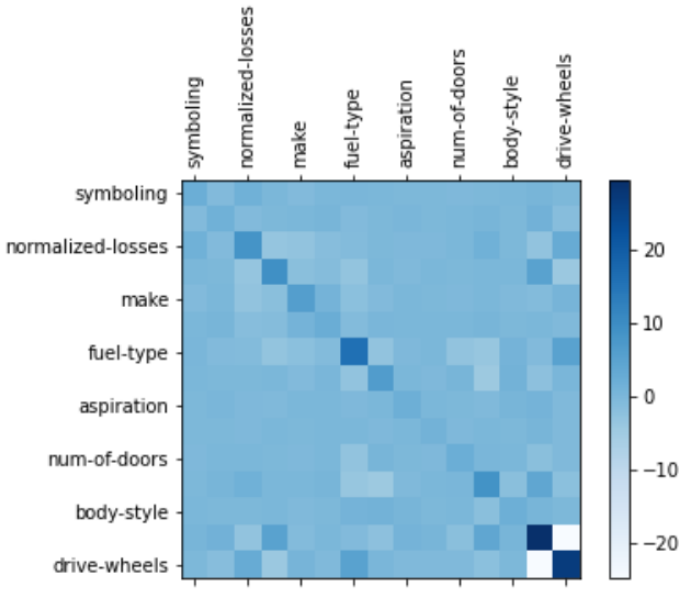
Пристапот со **PCA** за справување со мултиколинеарност **дава добри резултати кога интерпретабилноста на моделот не ни е важна**. Ова значи дека ако сакаме да разбериме кои атрибути се важни и колку се важни, овој метод не е вистинскиот избор. Во ваков случај би требало да размислиме за справување со мултиколинеарност преку пресметка на VIF коефициентот или други алтернативни техники.

5.2. Справување со мултиколинеарност со помош на VIF

Метод за справување со мултиколинеарност којшто овозможува **поголема интерпретабилност на моделот**, е пресметка на VIF вредност за секој од нумеричките атрибути. **VIF или Variance Inflation Factor** е мерка за количината на мултиколинеарност во множеството предиктори. Овој коефициент не може да се пресмета за категориски атрибути, а негова пресметка врз енкодирани категориски атрибути нема смисла.

Идејата е да се исфрлат оние атрибути кои имаат вредност за VIF поголема од 10. Оние чиешто VIF е поголемо од 5, пожелно е да се анализираат, но не мора да се отстранат. Пред да се отстрани атрибутот со најголема вредност за VIF, **се тестира и неговата линеарна зависност со зависната променлива** за да се има во предвид колку е добар тој атрибут како предиктор.

Во податочното множество Automotive Dataset има атрибути со огромно VIF, како што се атрибутите wheel-base со VIF=1945.21, length со VIF=1806 и многу други. На визуелизацијата креирана со помош на инверзна корелациска матрица на независните променливи, може да се забележат вредностите на VIF за секој од атрибутите по главната дијагонала (*Слика 26*.).



Слика 26. Инверзна коваријансна матрица за приказ на вредностите на VIF

Воочливо е дека итеративно, по секое отстранување на атрибутот со највисока вредност на VIF, вредноста на VIF на преостанатите атрибути се намалува. Итеративно**, се отстрануваат атрибутите width, wheel-base, length, height, highway-mpg, curb-weight, peak-rpm, bore, stroke, engine-size и normalized-losses.**
Во податочното множество **преостануваат само четири атрибути** со вредност за VIF помала од 10, а тоа се атрибутите: **symbolling, compression-ratio, horsepower и city-mpg**.

По секое отстранување на атрибут, податочното множество беше зачувано во листа на податочни множества, врз кои на крај се тренираа посебни модели на едноставна линеарна регресија за да се споредат вредностите на MSE, MAE и R2. Резултатите се прикажани во *Табела 1.*

Отстранувањето на атрибути со најголема вреднос на VIF може да резултира со **губиток на информации**. Решение за ова е да се креираат **композитни атрибути со интеракција** (wheel-base*length, width*height, bore*stroke, highway-mpg*city-mpg). За жал, резултатите не се подобрија, односно на крај, повторно останаа само четири атрибути во податочното множество.

	RMSE	MAE	R2
26 features	2609.858785	2046.639776	0.630123
25 features	2733.357722	2105.391037	0.594289
24 features	2739.056273	2143.412186	0.592596
23 features	2851.464568	2227.321543	0.558470
22 features	3020.609856	2571.220628	0.504535
21 features	3028.941676	2573.738153	0.501798
20 features	3083.006102	2554.927761	0.483854
19 features	3154.187736	2602.473802	0.459745
18 features	3167.676593	2613.954283	0.455114
17 features	3349.777646	2388.193504	0.390665
16 features	3712.642581	2838.744219	0.251503
15 features	3711.154635	2823.095385	0.252103

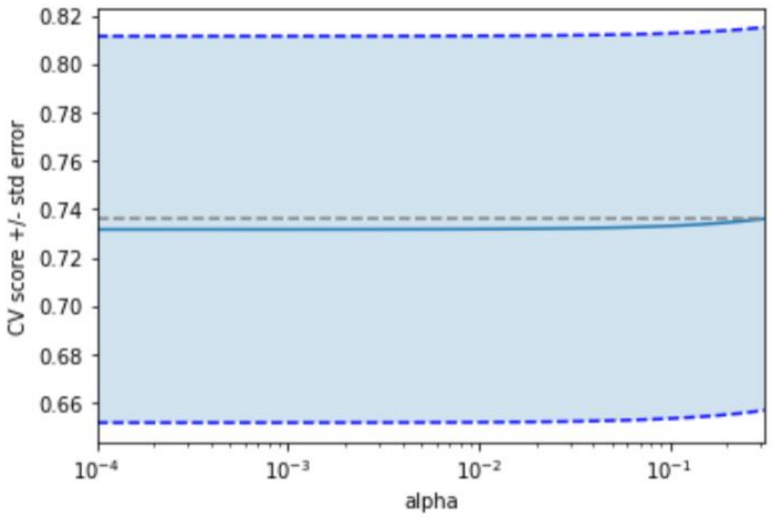
Табела 1. Евалуациски метрики на модели тренирани врз податочни множества со различен број на атрибути

5.3. Справување со мултиколинеарност со Lasso регресија

Друг начин за намалување на мултиколинеарноста е да се користи **регуларизација** за да се задржат сите атрибути од податочното множество, но **да се намали магнитудата на коефициентите** на моделот. Ова е добро решение кога секој предиктор во одредена мера придонесува кон предвидувањето на зависната променлива.

Lasso (Least Absolute Shrinkage and Selection Operator) регресијата го решава истиот оптимизациски проблем како и Ridge регресијата, но ја користи **L1 нормата** како мерка за комплексност, наместо L2.

Предпроцесираното податочно множество Automotive Dataset го поделивме на тренирачко и тестирачко множество со размер 80:20, а интервалот во кој може да се очекуваат резултатите при тренирање на Lasso регресија го претставивме со визуелизација на *Слика 27*. Воочливо е дека интервалот е доста широк, што значи дека вредностите за средната квадратна грешка ќе бидат повторно огромни. Околината во која што би очекувале да се наоѓаат резултатите е ограничена со темно сини испрекинати линии. Светло сината линија ги означува перформансите на моделот за различни вредности на алфа (параметар кој ја балансира минимизацијата на RSS наспроти максимизацијата на сумата на квадратите на коефициентите). Во овој случај, моделот има најдобри перформанси за алфа >=10.



Слика 27. Околина во која се очекуваат резултатите на модел на Lasso регресија

6. Recursive Feature Elimination (RFE) Feature Selection

Recursive Feature Elimination (RFE) е медода за селекција на атрибути, така што **рекурзивно го отстранува најслабиот атрибут (или атрибути) додека не се достигне специфицираниот број на атрибути**. **Не ги зема во предвид мултиколинеарноста и нивото на сигнификантност на атрибутите.**

За податочното множество Automotive Dataset, ги споредивме резултатите добиени со тренирање на едноставна линеарна регресија врз различен број на атрибути (од 1 до 25, којшто е вкупниот број на атрибути во тренирачкото множество). И тука се користи енкодираното податочно множество, а истото повторно се скалира со MinMaxScaler.

Според RFE, **најсигнификантен предиктор е атрибутот engin-size**. Најмало **MSE** се добива со **17 атрибути** (приближно **4089896**), најмало **MAE** се добива со **13 атрибути** (приближно **1160**), а најмало **R2** се добива исто така за **17 атрибути** (приближно **0.93**).

Како што може да се забележи, вредностите на овие евалуациски метрики се повторно **незадоволителни**. Останува да се открие причината која ги предизвикува овие резултати.

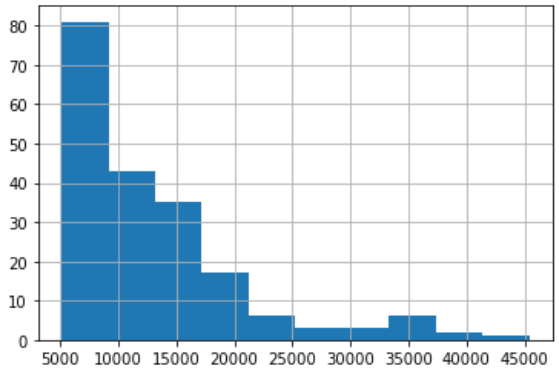
7. Справување со наклонетост (skewness)

Можеби наклонетоста на зависната променлива е причина резултатите на евалуациските метрики да бидат незадоволителни. Од хистограмот на *Слика 29*. може да асе забележи дека најголем број на автомобили во 1985-та година имале цена од 5000 до 10000 долари. Вака **наклонетите податоци може да имаат негативно влијание врз моќта на предиктивниот модел** доколку не се справиме соодветно со ваквиот проблем.

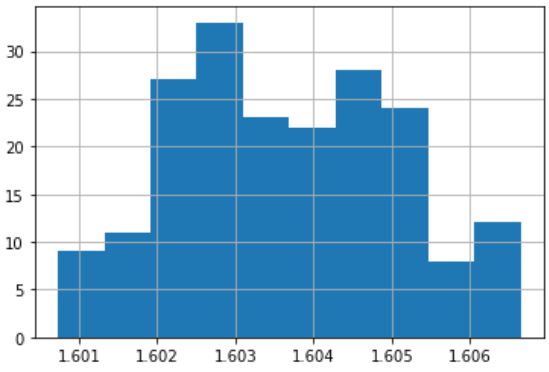
Има различни начини за справување со наклонетоста на зависната променлива, како заменување на нејзините вредности со нивниот корен, логаритам или пак со box-cox трансформација. **Box-cox трансформацијата ја трансформира дистрибуцијата на зависната променлива која не е нормална во нормален облик.**

На променливата price од податочното множество Automotive Dataset применивме box-cox трансформација (*Слика 30.)* и повторно направивме feature selection со RFE. **Резултатите драматично се подобрија!**

По трансформацијата, најдобро **MSE (1.671720297304309e-07)** се доби за **6 атрибути,** најдобро **MAE (0.0003209796912567953)** се доби со **5 атрибути**, а најдобро **R2 (0.92)** исто така со **5 атрибути**.



Слика 29. Дистрибуција на атрибутот price пред box-cox трансформација

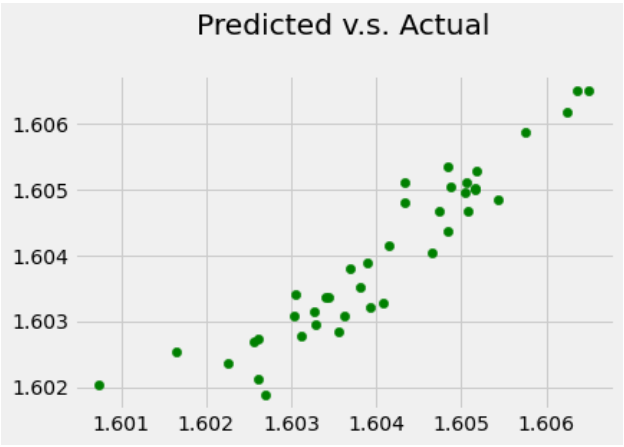


Слика 30. Дистрибуција на атрибутот price после box-cox трансформација

Линеарна Регресија

1. Едноставна Линеарна Регресија

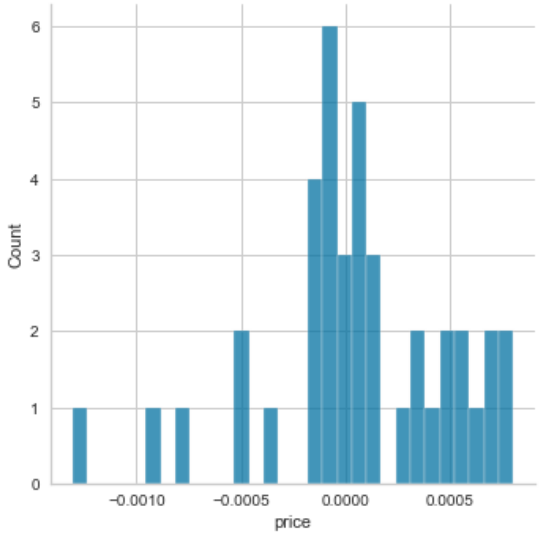
Мултиколинеарноста влијае само врз коефициентите и врз p-values на атрибутите, но не и врз прецизноста на предикцијата. Бидејќи **нашата примарна цел е да правиме предвидувања за цените на автомобили од 1985-та година**, не е нужно да ја разбереме улогата на секоја независна променлива. Оттука, не е неопходно да се справиме со колинеарноста, па затоа **ќе ги тренираме понатамошните модели врз целосното податочно множество.**



На *Слика 31.* гледаме дека постои **силна корелација помеѓу предвидените вредности за цената и вистинските вредности на цената**, па затоа може да заклучиме дека предиктивниот **модел ни е прецизен.**

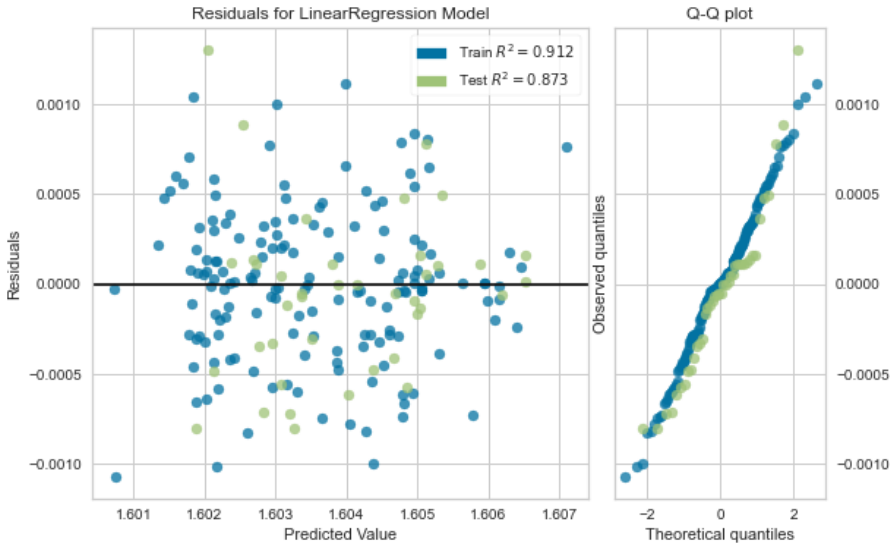
Слика 31. Предвидени наспроти вистински вредности на цената со модел на едноставна линеарна регресија

Од хистограмот на *Слика 32.* Се забележува дека **грешката за повеќето примероци е 0 или е многу блиску до 0.** Ова е нешто што сакаме да се постигне при тренирање на модел на линеарна регресија.



Слика 32. Ниво на грешка на примероците во Automotive Dataset

За анализа на варијансата на грешката на линеарниот модел често се користат така наречени **residual plots.** Ако точките се случајно распределени околу хоризонталната оска, тогаш линеарниот модел е соодветен за податоците. Во обратен случај, можеби нелинеарен модел ќе биде посоодветен за податоците. Во нашиот случај, на *Слика 33*. Гледаме дека дистрибуцијата на резидуалите е **нормална и рандом** во дводимензионален простор. Ова означува дека линеарниот модел дава добри резултати за ова податочно множество.



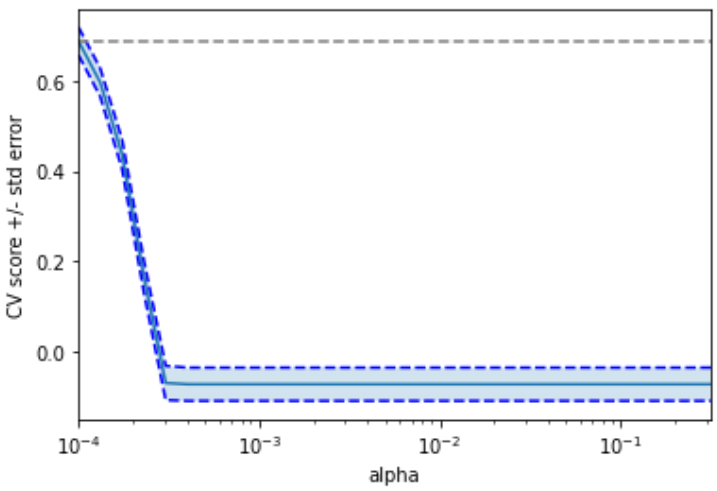
Слика 33. Residual Plot

2. Регуларизација

Функцијата на грешка може да се пенализира со различни регресиони модели, како што се: **Ridge Regression, Lasso Regression, Elastic Net** (ги комбинира L1 и L2 нормите) и **Bayesian Ridge Regression**. Добиените резултати со различните алгоритми се споредени во *Табела 2*. **BayesianRidge резултира со најдобри резултати**, додека пак Lasso и ElasticNet резултираат со идентични резултати.

	MSE	RMSE	MAE	R2	model variance
Ridge	2.173714e-07	0.000466	0.000365	0.906998	5.609584e-08
Lasso	2.379586e-06	0.001543	0.001310	-0.018105	6.140866e-07
ElasticNet	2.379586e-06	0.001543	0.001310	-0.018105	6.140866e-07
BayesianRidge	2.160445e-07	0.000465	0.000357	0.907565	5.575343e-08

Табела 2. Споредба на резултати добиени со различни алгоритми за регуларизација



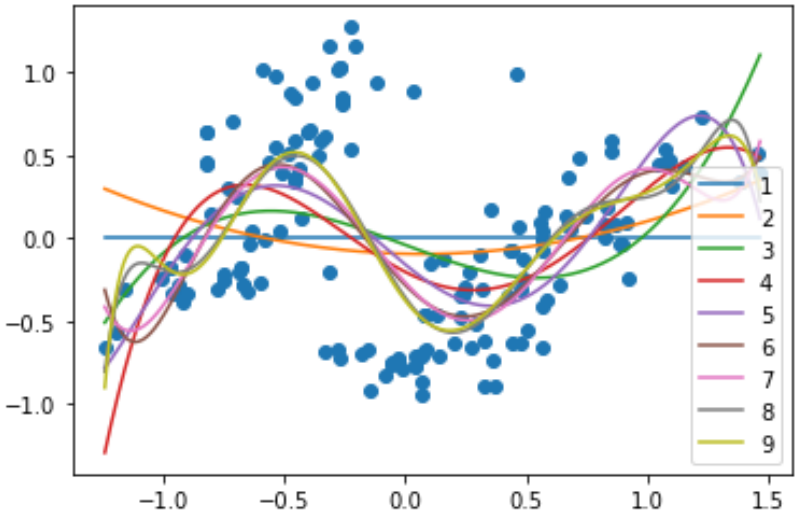
Слика 34. Интервал во кој се очекуваат предикциите по применување на box-cox трансформација

По применувањето на box-cox трансформација на зависната променлива, забележано е огромно подобрување, односно стеснување на интервалот во кој може да ги очекуваме вредностите на предвидувањата. Ова значи дека предвидувачката моќ на моделот е голема, а грешката во предвидување е мала (*Слика 34*.).

Моделот на Lasso регресија дава најдобри вредности за алфа ~10^(-3.5).

3. Feature Expansion

Иако од *Слика 33*. заклучивме дека соодветниот модел за Automotive Dataset е линеарен, ќе видиме како полиномен линеарен модел го фитува податочното множество. Од *Слика 35*. уште еднаш ќе заклучиме дека посоодветен модел за ова податочно множество е линеарниот, бидејќи подигнувањето на атрибутите на степен поголем од 1 ги влошува резултатите.

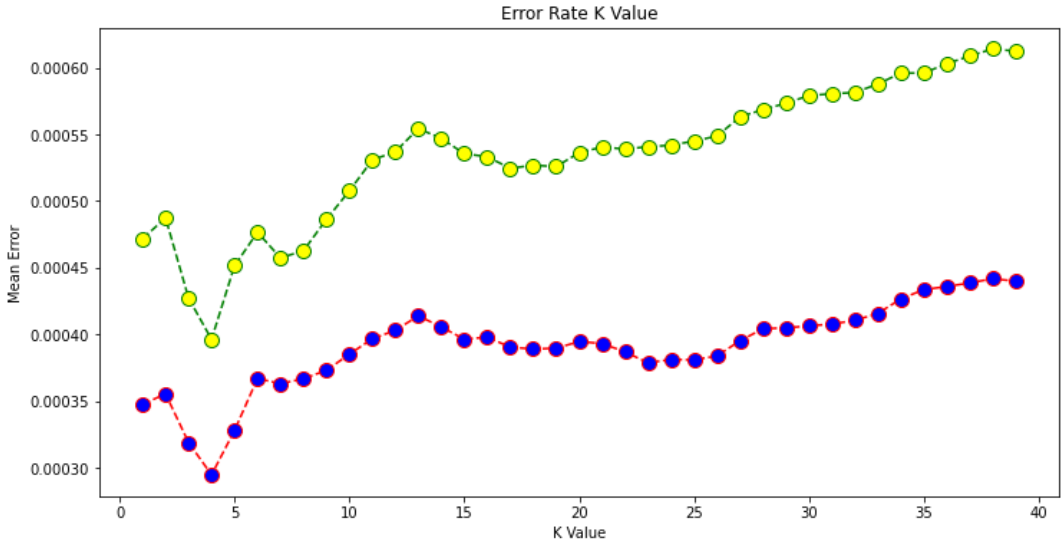


Слика 35. Полиномен регресионен модел на степен од 1 до 9

4. KNN за регресија

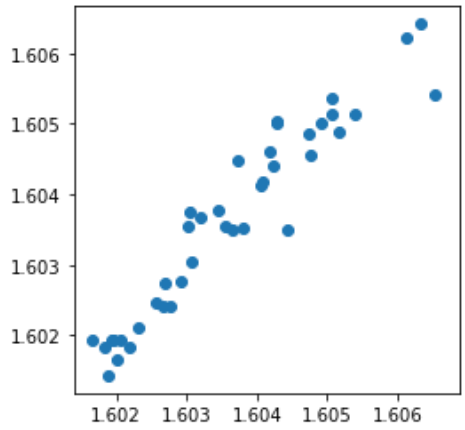
KNN алгоритмот ја користи сличноста на атрибутите за да ги предвиди вредностите на нови податочни точки (примероци). На новиот примерок се доделува вредност базирана на тоа колку е слична со останатите точки во податочното множество.

На Слика 36. Се гледа дека најмала е грешката кога бројот на најблиски соседи до кои треба да се пресмета растојанието е 4 (0.0006).



Слика 36. Грешка за различен број на соседи во KNN регресија

Истрениранот модел на KNN регресија со 4 најблиски соседи резултира со **MSE= 1.5709175195855604e-07, MAE=0.00029, R2=0.9.** Предвидените наспроти вистинските вредности со овој модел се прикажани на Слика 37. Од која саклучуваме дека моделот за предикција е добар.



Слика 37. Предвидени наспроти релни вредности со KNN за регресија

Иако со овој алгоритам се добиваат одлични резултати, треба да се земе во предвид фактот што се користи Евклидово растојание на податочно множество со категориски променливи, што не е соодветно. Тоа што категориските променливи се енкодирани, не е решение на овој проблем.

5. Дополнителни техники за Линеарна Регресија

5.1. Support Vector Regression (SVR)

SVR се базира врз основните идеи на Support Vector Machine (SVM) – моќен алгоритам за класификација – но ги применува за предвидување на реални вредности, наместо класи. SVR дава флексибилност при дефинирањето колкава грешка е прифатлива во моделот и во соглесност со тоа, бара соодветна линија (или рамнина во повисок димензионален простор) за да ги фитува податоците.

Добиените резултати со SVR со различни типови на кернели се прикажани во *Табела 3.* **Најдобри резултати се добиваат со Radial Basis Function кернел**.

Kernel	RMSE	MAE	R2
Linear	0.00052	0.00039	0.81
Radial Basis Function	0.00047	0.00038	0.85
Polynomial (d=3)	0.00092	0.00076	0.41

Табела 3. Резултати добиени со различни кернели на SVR

5.2. Relevance Vector Machines (RVR)

RVR е техника која користи Bayesian inference за да добие решение за регресија или веројатносна класификација. Има идентична функционална форма како Support Vector Machine, но со веројатносен пристап.

Врз Automotive Dataset истрениравме RVM модел со Radial Basis Function кернел, кој резултираше со (*Табела 4.*):

Kernel	RMSE	MAE	R2
Radial Basis Function	0.00047	0.00039	0.89

Табела 4. Резултати добиени со RVM со rbf кернел

5.3. XGBoost

XGBoost е **ensemble** алгоритам базиран на дрва на одлучување кој користи gradient boosting рамка. Како комбинација на предностите на **random forest** (базиран на bagging со што се бара просекот на резултатите на многу одлучувачки дрва) и **gradient boosting**, грешката на предвидување на овој алгоритам може да е десет пати помала отколку грешката со која резултираат random forest или boosting индивидуално.

Сепак, **за нашиот случај овој алгоритам не беше соодветен** и резултираше со повисоки MSE и MAE од претходните алгоритми, како и многу мало R2 (-3063.96).

5.4. Random Forest Regressor

Random Forest е алгоритам кој користи **ensemble методи (bagging)** за да ги реши проблемите на регресија или пак класификација. Алгоритмот функционира така што конструира многу одлучувачки дрва за време на тренирање и како излез ја пресметува средната вредност (за регресија) или модата (за класификација) на предикцијата од индивидуалните дрва.

За Automotive Dataset искористивме **Random Forest за регресија** чиишто резултати беа одлични, а се прикажани во *Табела 5.:*

Algorithm	RMSE	MAE	R2
Random Forest Regressor	0.00038	0.00031	0.93

Табела 5. Резултати добиени со Random Forest Regressor

6. Заклучок

Резултатите значително се подобрија по примената на box-cox трансформација, со што уште еднаш заклучивме дека наклонетоста на зависната променлива може да ја намали предвидувачката моќ на моделите. **Со најдобри резултати за предикција на цена на автомобили од 1985-та година од податочното множество Automotive Dataset резултираше алгоритмот Random Forest Regressor.**

Класификација

Во машинското учење, класификацијата се однесува на проблем за предвидливо моделирање каде ознаката на класата ќе биде предвидена според даден пример на внесени влезни податоци. Во нашиот случај, класификацијата како техника ја користиме за предвидување ознаки на **бинарни и мултикласни** атрибути. За почеток започнуваме со предикција на ознака на мултиклас атрибутот *symbolling*. *Symboling* одговара на нивото на ризик од осигурување на автомобил. На автомобилите првично им е доделен *symbolig* на фактор на ризик поврзан со нивната цена. Потоа, ако автомобилот е поризичен, овој *symboling* се прилагодува со негово поместување на скалата. Вредноста од +3 покажува дека автомобилот е ризичен, -3 дека веројатно е прилично безбеден. Подоцна, за бинарниот атрибут **num-of-doors** кој означува колку врати има дадена кола правиме предикција со помош на Логистичка регресија кадешто добиваме одлични резултати.

1. Класификација на атрибутот *symbolling* користејќи HEOM (Heterogeneous Euclidean-Overlap Metric) метрика за растојанија

Детално објаснување за HEOM има во делот на Кластерирање 2.2.

Најпрвин, како независни атрибути ги користиме сите атрибути освен таргетот *symbolling* и истите ги чуваме во променлива **X**, додека во **y** ја чуваме таргет променливата.

Правилото што го следиме овде е доколку користиме алгоритам што пресметува растојание,тогаш ги скалираме нашите атрибути. Во овој случај тоа го правиме со RobustScaler бидејќи ќе го користиме **Kneighbors** класификаторот. K во името на овој класификатор ги претставува k-те најблиски соседи, каде што k е цела вредност одредена со функцијата која пресметува грешка за вредности на k помеѓу 1 и 30. Оттука, како што сугерира името, овој класификатор спроведува учење врз основа на k најблиските соседи. Како што може да забележиме на *Графикот 1.1*. , најмала грешка добиваме за k=7.
Со тоа, го фитуваме моделот со параметар k=7.

```
Out[393]: Text(0, 0.5, 'Mean Error')
```

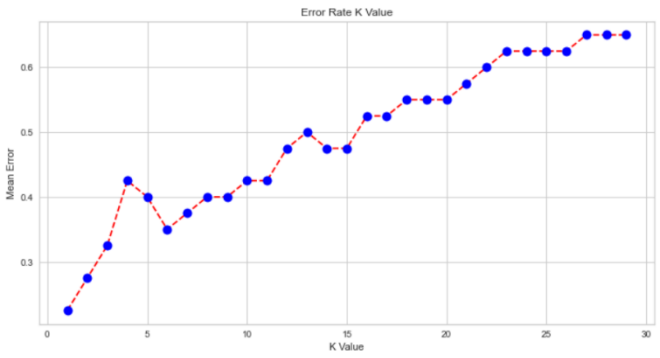


График 1. Error Rate K Value

2.Логистичка регресија на бинарен (num-of-doors) и повеќекласен атрибут (symbolling)

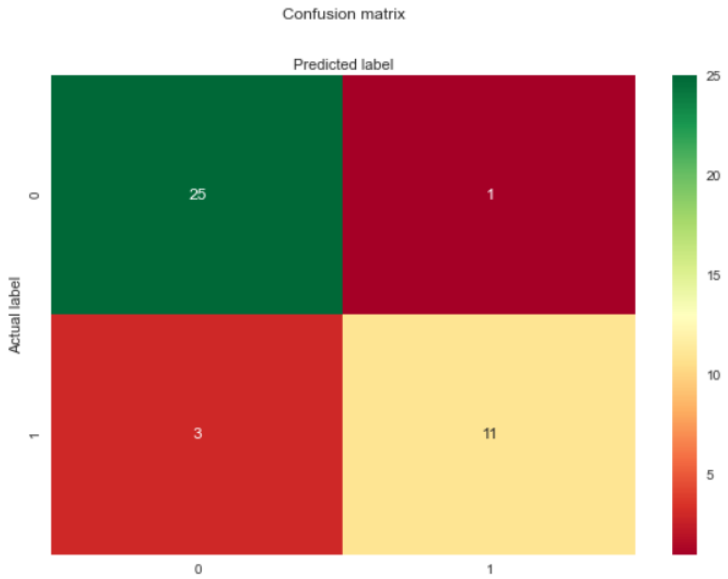
Логистичката регресија е процес на моделирање на веројатноста за дискретен исход со оглед на влезна променлива. Највообичаените логистички регресии моделираат бинарен исход: нешто што може да земе две вредности како што се точно/неточно, да/не (0,1), итн.

Ние користиме и мултиномна логистичка регресија која може да моделира сценарија каде што има повеќе од два можни дискретни исходи (multiclass).

Логистичката регресија е корисен метод за анализа за проблеми со класификација, каде што се обидуваме да одредиме дали новиот примерок најдобро се вклопува во категоријата.

Во тетратката е прикажана класификацијата со помош на логистичка регресија најпрвин на атрибутот num-of-doors кадешто добиваме одлични резултати (accuracy_score = 0.9) и истото е прикажано со Confusion matrix (Слика 2).

```
In [103]: fig, ax = plt.subplots()
          sns.heatmap(confusion_matrix(y_test, predictions), annot=True, cmap="RdYlGn" ,fmt='g')
          ax.xaxis.set_label_position("top")
          plt.tight_layout()
          plt.title('Confusion matrix', y=1.1)
          plt.ylabel('Actual label')
          plt.xlabel('Predicted label');
```



Слика 2. Confusion matrix за логистичка регресија на num-of-doors

2. ROC-AUC со различни класификатори (LogisticRegression, GradientBoostingClassifier, KNeighborsClassifier)

ROC-AUC (Receiver operating characteristic - Area under curve) крива е график што ги прикажува перформансите на класификацискиот модел на сите прагови на класификација. Оваа крива прикажува два параметри: **True Positive Rate** и **False Positive Rate.**

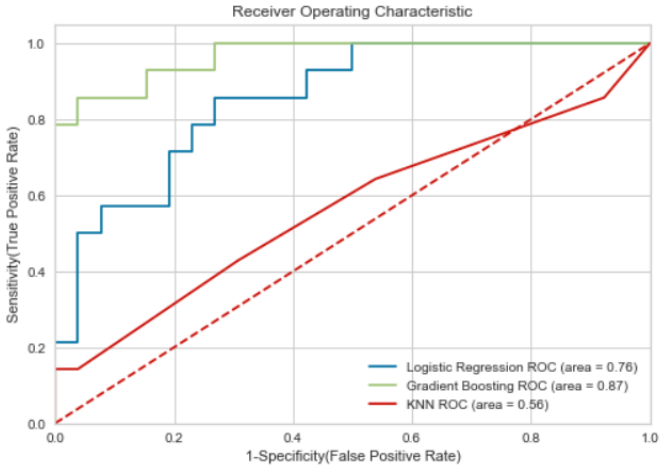
Намалувањето на прагот на класификација (**treshold**) класифицира повеќе примероци како позитивни, со што се зголемуваат и False Positive и True позитивни.

За да ги пресметаме точките во кривата на ROC, би можеле многу пати да оцениме модел на логистичка регресија со различни прагови на класификација (**threshold**), но тоа би било неефикасно. За среќа, постои ефикасен алгоритам базиран на сортирање што може да ни ги обезбеди овие информации, наречен AUC.

AUC означува „Површина под ROC кривата“. Односно, AUC ја мери целата дводимензионална област под целата ROC крива од (0,0) до (1,1).

Кога AUC е 0.76, тоа значи дека постои 76% шанса моделот да може да направи разлика помеѓу позитивна класа и негативна класа. Најлошата ситуација е кога AUC е приближно 0.5. Тогаш моделот нема капацитет да направи разлика помеѓу позитивна класа и негативна класа.

Од Слика 3. можеме да заклучиме дека KNN е најлош алгоритам што треба да се користи за класификација на овој атрибут, бидејќи моделот нема да прави разлика помеѓу позитивна класа и негативна класа (AUC ~ 0.5). Gradient Boosting алгоритмот има најдобри резултати за овие податоци. AUC е најголем кога го користиме овој алгоритам.



Слика 3. ROC-AUC за num-of-doors со различни алгоритми

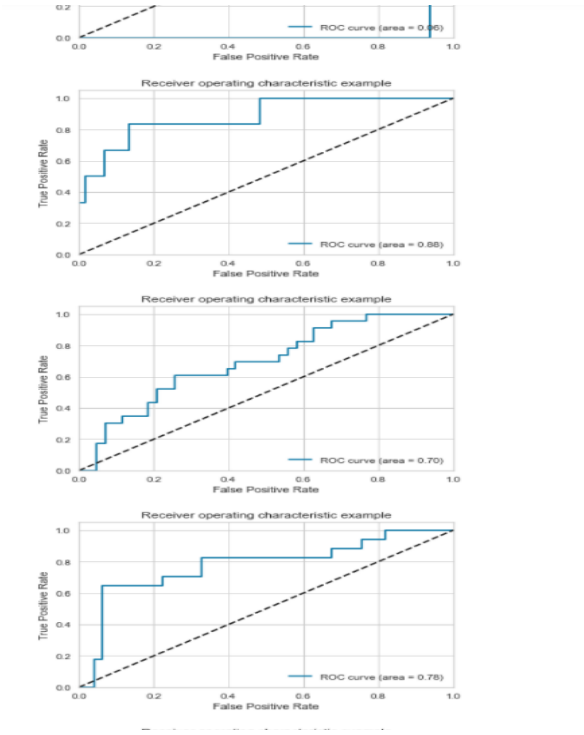
3. Мултиномна логистичка регресија за атрибут symbolling

Стандардно, логистичката регресија не може да се користи за задачи за класификација кои имаат повеќе од две ознаки на класа, таканаречена класификација со повеќе класи.

Наместо тоа, потребна е модификација за поддршка на проблеми со класификација на повеќе класи. Еден популарен пристап за прилагодување на логистичката регресија на проблемите со класификација на повеќе класи е да се подели проблемот со повеќе класи на класификација во повеќе проблеми со бинарна класификација и да се вклопи стандарден модел на логистичка регресија на секој потпроблем. Техниките од овој тип вклучуваат модели на OneVsRest и OneVsOne.

Алтернативниот пристап вклучува промена на моделот на логистичка регресија за да се поддржи директно предвидување на вредности за повеќе класи. Поточно, да се предвиди веројатноста дека влезниот примерок припаѓа на секоја позната класа.

Иако најдобриот начин да се направи визуелизација на мултиномна логистичка регресија е матрица на конфузија, успеавме да направиме ROC-AUC со пристап OneVsAll. Всушност, ги земаме сите класи и го бинаризираме проблемот за секоја класа (Слика 4).



Слика 4. One-vs-all ROC-AUC за symbolling

4. LDA (Linear Discriminant Analysis) за symbolling

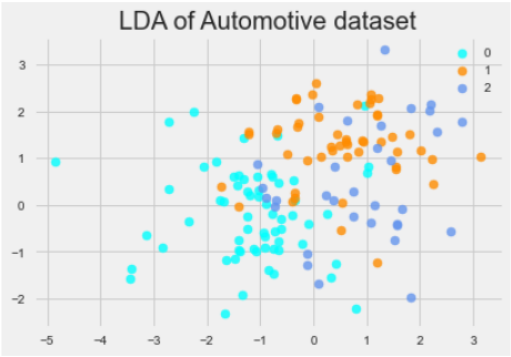
Логистичката регресија е едноставен и моќен линеарен алгоритам за класификација. Исто така, има ограничувања што укажуваат на потребата за алтернативни алгоритми за линеарна класификација.

Логистичката регресија е наменета за проблеми од две класи или бинарна класификација. Може да се прошири за класификација во повеќе класи, но резултатите може да не се значајни. Не е стабилна со добро поделени класи. Логистичката регресија може да стане нестабилна кога класите се добро разделени.

Затоа, **ЛДА** се однесува на секој од овие проблеми и е линеарен метод за проблеми со класификација на повеќе класи. ЛДА може да се користи и за бинарни и за проблеми со повеќе класи.

Помага да се намалат високо-димензионалните податоци претставени на пониско-димензионален простор. Целта е да се направи ова за раздвојување на класите и намалување на ресурсите и трошоците за компјутерот.

Во нашето податочно множество направивме ЛДА за атрибутот symbolling (Слика 5).



In [199]:
```
lda = LDA()
lda.fit(X, y)

# define model evaluation method
# kfold with k=10
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3)

# define grid
grid = dict()
grid['solver'] = ['svd', 'lsqr', 'eigen']
# define search
search = GridSearchCV(lda, grid, scoring='accuracy', cv=cv, n_jobs=-1)
# perform the search
results = search.fit(X, y)
# summarize
print('Mean Accuracy: %.3f' % results.best_score_)
print('Config: %s' % results.best_params_)
```

```
Mean Accuracy: 0.682
Config: {'solver': 'lsqr'}
```

Слика 5. LDA за symboling

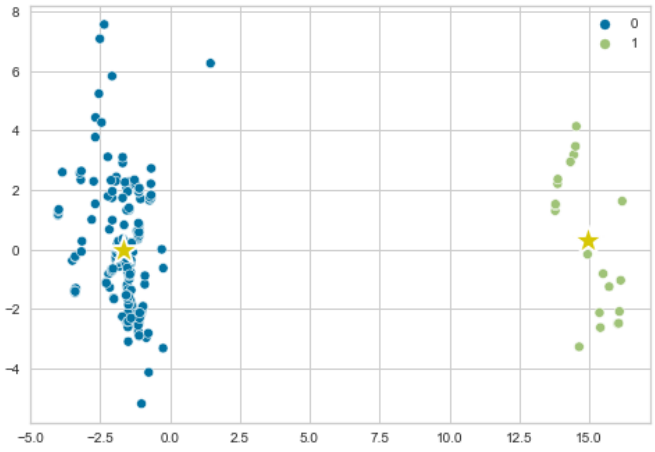
Кластерирање

Кластерирање е процес на поделба на податочното множество во групи, така што **податоците кои припаѓаат во иста група се послични помеѓу себе отколку со податоците во другите групи**. Поконкретно, целта е податоците да се поделат во групи според некоја заедничка **латентна (непозната) променлива**. За податочно множество чија главна намена е ненадгледувано учење, латентната променлива не е очигледна, односно добиените кластери не секогаш може да се толкуваат.

Бидејќи Automotive Dataset е множество наменето за нагледувано учење, карактеристиките во однос на кои се добиени кластерите може полесно да се толкуваат со низа мерки за евалуација.

Пред примена на некој од алгоритмите за кластерирање, врз веќе енкодираното податочно множество, применивме *RobustScaler* којшто соодветно ги адресира екстремните вредности. Потоа, ја намаливме димензионалноста на две компоненти.

1. K-means

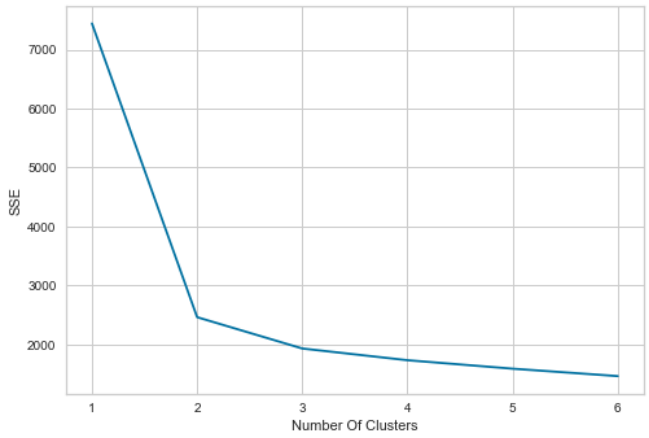


Слика 38. Класрери добиени со K-means

K-means е **hard-clustering** алгоритам којшто секоја точка ја доделува на еден од K-те кластери. На самиот почеток, иницијализира **K рандом точки (центроиди)** во просторот. Во секоја итерација го пресметува растојанието од секоја обзервација во податочното множество до центроидите. Процесот се повторува додека нема промена во припадноста на обзервациите на некој од кластерите.

Како и секој алгоритам во машинското учење, и K-means има свои предности и недостатоци. Поради својата фиксна коваријансна матрица, се добиваат лоши резултати доколку податоците во дводимензионален простор се групирани во кластери со разлилни големини, густини или пак имаат неглобуларен облик.

Дополнително, овој алгоритам го користи **Евклидовото растојание** за да ја одреди оддалеченоста на секоја точка до секој од центроидите. Слично како кај KNN алгоритмот за надгледувано учење, повторно **ова е несоодветно поради присуството на категориски променливи** во податочното множество.

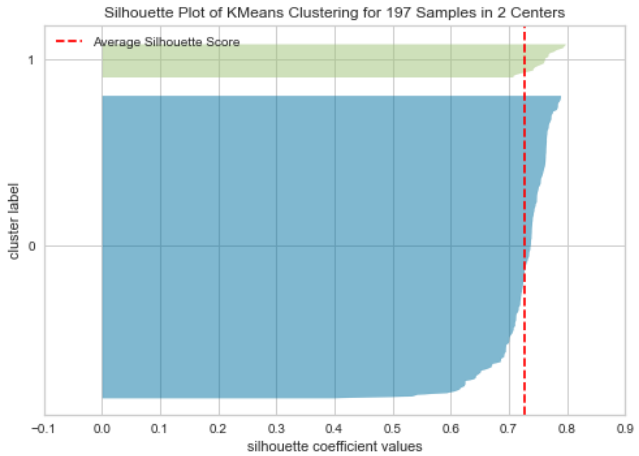


За да се одреди оптималниот број на кластери, визуелно ќе ја претставиме **сумата на квадратните грешки добиена со различни вредности на К**. Ваквиот метод се нарекува **Elbow method** бидејќи се избира тоа К каде што има превиткување на кривата, односно она К до кое грешката нагло опаѓа. Овој метод може да се погледне на *Слика 39.*, од која е очигледно дека оптикалното К = 2.

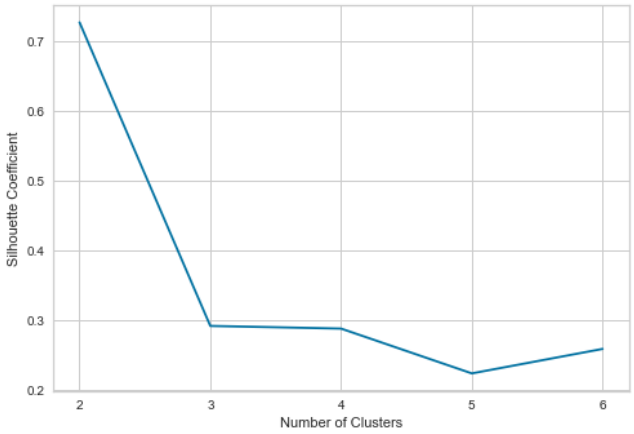
Слика 39. Elbow method

Друг начин да се одреди оптималната вредност на К е пресметка на **Silhouette коефициентот**. Со *Yellowbrick* библиотеката за визуелизации во *Python* постои функција *silhouette_visualizer* со која визуелно може да се претстави silhouette коефициентот за секој примерок по кластер, евалуирајќи ја густината и сепарабилноста меѓу кластерите. Резултатот се добива со барање просек на silhouette коефициентите за секој примерок, а истиот е вредност во ранг од -1 до +1 каде +1 означува високо ниво на сепарабилност, а -1 означува дека примероците се доделени на погрешни кластери.

На ваквите визуелизации, кластерите со повисоки резултати имаат пошироки силуети. **Добар silhouette коефициент има вредност над 0.5.** Од *Слика 40*. И *Слика 41.,* уште еднаш заклучуваме дека во нашиот случај, **оптималната вредност за бројот на кластери е 2.**



Слика 40. Silhouette coefficient

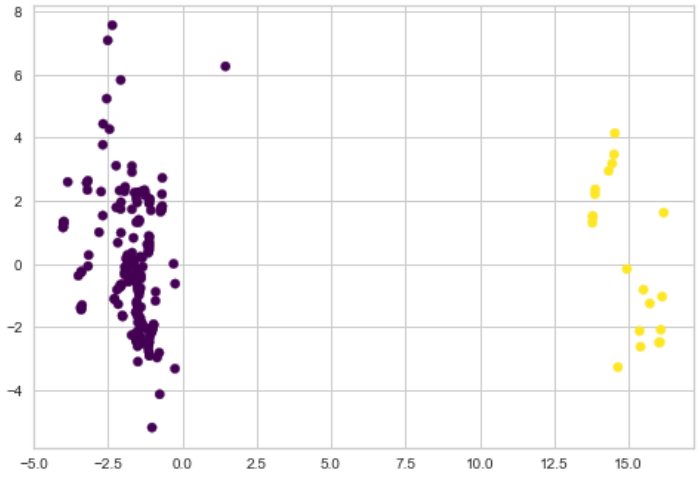


Слика 41. Silhouette coefficient

2. Дополнителни техники за кластерирање

2.1. Gaussian Mixture Models

Gaussian Mixture Models е **soft-clustering веројатносен модел** којшто претпоставува дека сите точки во податочното множество доаѓаат од микс на конечен број гаусови распределби со непознати параметри. Оттука, **GMMs се обидува да ги групира податоците кои припаѓаат на иста дистрибуција**.



Слика 42. Кластери добиени со GMMs

За разлика од K-means алгоритмот, којшто секоја точка ја доделува на точно еден кластер, Gaussian Mixture Models секоја точка ја доделува на секој од кластерите со одредена веројатност. Друга предност на овој алгоритам е **дијагоналната коваријансна матрица** со која го надминува проблемот на кластерирање податоци со различни големини.

2.2. Имплементација на пристап базиран на научен труд за кластерирање со помош на алгоритмот Nearest Neighbors

Низ овој труд, повеќепати беше нагласена несоодветноста на користење на Евклидовото растојание за податочно множество кое содржи различни типови на податоци, вклучувајќи нумерички и категориски.

Следува објаснување на пристап од научен труд со наслов „Improved Heterogeneous Distance Functions“ во кој се дефинирани различни метрики за растојание соодветни за вакви мешани податочни множества.

На наше изненадување, библиотеката *Scikit-Learn* во *Python* не овозможува користење на метрика за растојание соодветна за податочни множества со различни типови податоци. Дозволува користење на големо множество метрики за растојание, но сите се за множества кои имаат или нумерички или категориски атрибути.

Овој проблем бил мотивација за развој на пакетот *Distython* во *Python* чија цел е да **овозможи хетерогени метрики за растојание** кои се компатабилни со

Scikit-learn библиотеката и кои може да се употребат во многу алгоритми кои таа ги нуди. Содржи имплементација на метрики за растојание опишани во научниот труг споменат погоре, кои **можат да се справат со множества кои имаат missing values и различни типови на податоци.**

Метриката за растојание која ја искористивме за кластерирање со **Nearest Neighbors** алгоритмот е **HEOM (Heterogeneous Euclidean-Overlap Metric)**, која комбинира три различни алгоритми за да се справи со ваквите проблеми (доколку атрибутот е категориски, враќа 0 ако атрибутот е од иста класа, а 1 во обратен случај; доколку е нумерички, го пресметува растојанието користејќи нормализирана Евклидова метрика; доколку вредноста на атрибутот недостасува, враќа 1).

Со овој алгоритам, повторно **се добија два кластери** за Automotive Dataset.

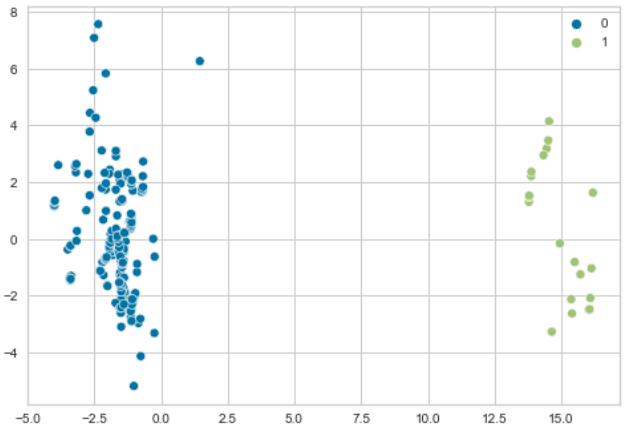
2.3. K-medoids со хетерогена метрика за растојание (HEOM)

Слично на K-means алгоритмот, и K-medoids ги дели податоците во K кластери, со таа разлика што **K-те центроиди мора да бидат податоци во податочното множество**. Додека K-means се обидува да ја минимизира тоталната квадратна грешла, **K-medoids ја минимизира сумата на различности** помеѓу точки кои се доделени на некој кластер и точката дефинирана како центроид на тој кластер. Уште повеќе, K-medoids е недефиниран за Евклидовото растојание и може да се специфицира друга метрика за растојание.

Во нашиот случај, **ја употребивме метриката за растојание HEOM** објаснета во точка 2.2 за K=2.

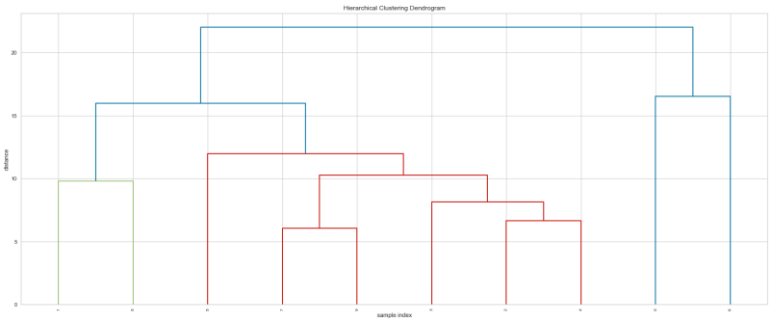
3. Хиерархиско кластерирање (Агломеративно)

Агломеративното кластерирање е најчест начин на хиерархиско кластерирање кој се користи за **групирање на објектите во кластери според нивната сличност**. Користи **bottom-up пристап**, што значи дека на почетокот секоја точка (податок) во податочното множество е **синглтон кластер**. Потоа, двата најслични кластери се спојуваат на алчен начин, се додека не се формира еден хиерархиски кластер или додека не се достигне специфицираниот број на кластери.

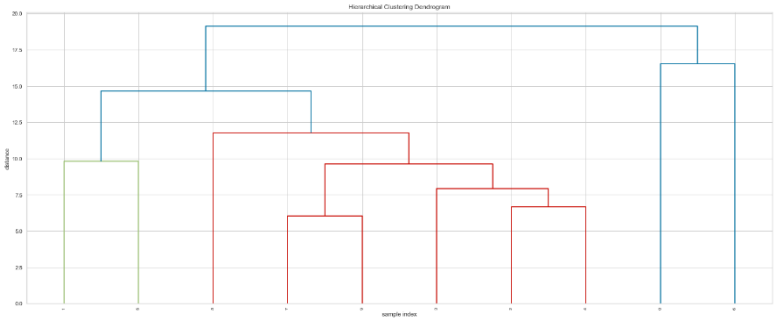


Слика 43. Кластери добиени со агломеративно кластерирање

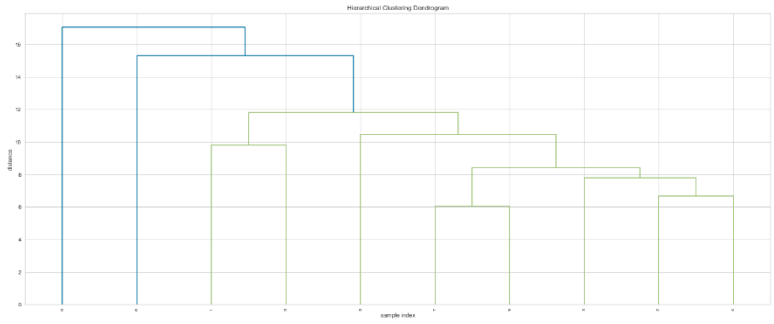
Агломеративното кластерирање користи **linkage методи** со кои се пресметуваат растојанијата, т.е. сличностите помеѓу сите објекти во податочното множество. Потоа, најблискиот пар кластери се групира во еден кластер, намалувајќи го бројот на преостанати кластери. Вакви linkage методи се: **single, complete, average и ward's linkage.** Различните кластери добиени со различни linkage методи може да се погледнат на дендограмите на *Слика 44., Слика 45., Слика 46., и Слика 47.* **Дендограмот** има облик на дрво и претставува дијаграм за приказ на кластери, а се користи за визуелна репрезентација на хиерархиско кластерирање. Со помош на ваквите визуелизации, интуитивно може да се одреди оптималниот број на кластеири.



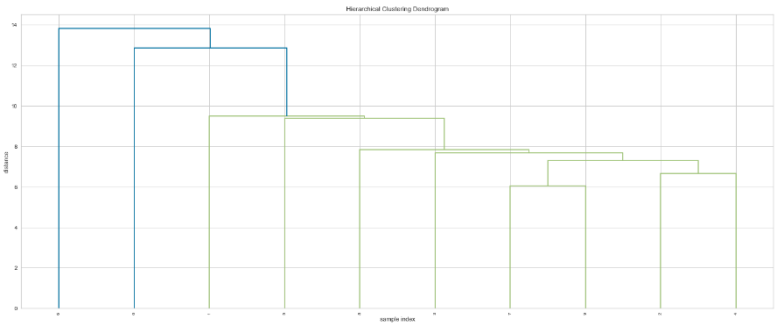
Слика 44. Кластери добиени со Ward's linkage



Слика 45. Кластери добиени со complete linkage



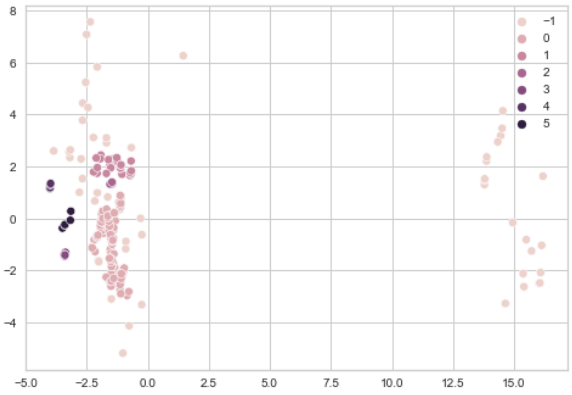
Слика 46. Кластери добиени со average linkage



Слика 47. Кластери добиени со single linkage

4. DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) е кластерирачки алгоритам кој се користи за да ги оддели кластерите со висока густина од оние со ниска густина. Се базира на претпоставката дека кластери се густи региони во просторот поделени според региони со помала густина. Ги групира најзбиените точки во еден кластер. Функционира добро за најразлични форми на кластери и кластери кои содржат шум, како на пример екстремни вредности.



Слика 48. Кластери добиени со DBSCAN

Како последица на ваквите концепти, овој алгоритам **дава лоши резултати кога точките во простотот се распределени со различни дустини, како што е случајот со нашето Automotive Dataset множество**. Добиените кластери изгледаат како на *Слика 48.*

5. Метрики за евалуација

За евалуација на добиените кластери со различните алгоритми за кластерирање, ги пресметавме следните метрики за евалуација:

- **ARI (Adjusted Rand Index)** – мерка на согласност помеѓу две партиции, едниот е даден од процесот на кластерирање, а другиот е дефиниран од екстерни критериуми.
- **AMI (Adjusted Mutual Information)** – варијација на mutual information, може да се користи за споредба на кластери.
- **Homogeneity** – одредува дали сите кластери содржат податочни точки кои припаѓаат на една иста класа.
- **Completeness** – одредува дали сите податочни точки кои припаѓаат на иста класа се кластерирани во еден кластер.
- **V-measure** – хармонична средина на Homogeneity и Completeness
- **Silhouette score** – го одредува квалитетот на кластерирањето, т.е. колку секој објект соодветно припаѓа на неговиот кластер.

Бидејќи податочното множество Automotive Dataset примарно е предвидено за надгледувано учење, **лесно може да се дефинираат екстерни критериуми за евалуација на кластерите.** Од сите атрибути во податочното множество, **најдобри резултати за евалуациските мертики се добија кога атрибутот fuel-type беше екстерен критериум, односно кога за fuel-type се претпостави дека е латентна променлива според која се групирани кластерите** (*Табела 6*.). Овој атрибут е бинарен, односно ги содржи категориите gas и diesel.

Оттука, **може да претпоставиме дека во едниот кластер се содржани автомобилите кои се напојуваат на гас, а во другиот кластер се содржани дизел автомобилите.**

	ARI	AMI	Homogenity	Completeness	V-measure	Silhouette score
K-means	1.000000	1.00000	1.000000	1.000000	1.000000	0.841223
Agglomerative	1.000000	1.00000	1.000000	1.000000	1.000000	0.841223
DBSCAN	0.108757	0.17378	0.472088	0.119018	0.190108	0.109050

Табела 6. Резултати добиени со различни алгоритми за кластерирање

Податочно множество добиено со Web Crawling

Што се однесува до податочното множество кое сами го собравме со веб кролинг и мануелна работа содржи 146 редици и 15 колони.

Како и секое raw податочно множество така и ова, без делот со претпроцесирање истото беше неупотребливо поради неконзистентност на типот податоците, метриките на самите атрибути, вредности кои недостасуваат итн. Исто така наидовме на overlapping вредности на атрибутите кои всушност се синоними.

Со таа цел, најпрвин почнуваме со чистење на множеството атрибут по атрибут додека не го доведовме во форма соодветна за примена на модели од машинско учење. Процесот на чистење вклучуваше справување со веќенаведените проблеми, справување со наклонетост на таргет променливата – цена, справување со екстремни вредности.

Врз ова податочно множество применивме техники само на линеарна регресија со цел да ја постигнеме крајната цел на проектот. Меѓу моделите кои ги истреniравме се: Simple Linear Regression, SVR, Bayesian Ridge Regression и Random Forest Regressor со кој и во овој случај се добија најдобри резултати.

Заклучок

Како заклучок на овој проект сакаме да истакнеме кои автомобиле останале во “мода” и ден денес. Брендовите **Audi, BMW, Mercedes** и **Chevrolet** можеме да ги истакнеме како автомобили кои ден денес држат најголема цена и се ал-тимери. Најинтересен е случајот со **Chevrolet** кој во далечната 1985-та година имал многу пониска цена во споредба со денешната цена на истите автомобили. Исто така, автомобилите од брендот **Porsche** бележат значителен пораст на цената со текот на времето.

Референци:

Вовед:

https://ignition.altran.com/en/article/what-will-the-car-of-the-future-look-like/

https://www.hotcars.com/classic-cars-that-are-way-more-expensive-than-when-they-were-new/

Научен труд за екстремни вредности:

https://benwtrent.github.io/2019/04/26/outlier-detection-from-scratch/?fbclid=IwAR1EbNN9uDHqCmIq6_NjVadd1U2cEIHF_2ZHnHg3qpiO9-qoD_HgOd3_Zzk

https://arxiv.org/pdf/0903.3257.pdf

Мултиколинеарност:

https://www.youtube.com/watch?v=ReXesvtkS4A&ab_channel=EXFINSISExpertFinancialAnalysis (inverse heatmap)

https://www.youtube.com/watch?v=7A6vukBvooE&ab_channel=BhaveshBhatt(addressing multicollinearity with PCA)

https://www.investopedia.com/terms/m/multicollinearity.asp

https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/

https://www.researchgate.net/post/Multicollinearity_issues_is_a_value_less_than_10_acceptable_for_VIF

https://www.researchgate.net/post/Whats_the_difference_between_correlation_and_VIF

https://statisticalhorizons.com/multicollinearity

https://blog.minitab.com/en/understanding-statistics/handling-multicollinearity-in-regression-analysis

https://towardsdatascience.com/how-to-detect-and-deal-with-multicollinearity-9e02b18695f1

https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/

ANOVA:

https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient

https://www.youtube.com/watch?v=EWYzeZbchR0&ab_channel=DataDaft

EDA:

https://analyticsindiamag.com/exploratory-data-analysis-functions-types-tools/

https://medium.com/code-heroku/introduction-to-exploratory-data-analysis-eda-c0257f888676

Skewness:

https://www.statisticshowto.com/box-cox-transformation/

Хетерогени метрики за растојание:

https://github.com/KacperKubara/distython

https://towardsdatascience.com/the-proper-way-of-handling-mixed-type-data-state-of-the-art-distance-metrics-505eda236400

https://arxiv.org/pdf/cs/9701101.pdf