**Savarankiškas darbas**

**2023 m. balandžio 26 – gegužės 17 d.**

Projekto rezultatų pristatymas: Gegužės 18-24 dienomis vyks projektų rezultatų pristatymas kur kiekvienas studentas pristatys tiek projektą ir rezultatus, tiek pasirinktą darbo metodiką. Po pristatymo vyks aptarimas. Pristatymui paruoškite skaidres, taip pat projekto kodą įdėkite į viešą github repositoriją (nepamirškite pakomentuoti README.md)

**Problema #4: automobilių kainos prognozavimas**

**Problema:** Kaip teisingai nustatyti savo automobilio kainą jį parduodant? Kaip nuspręsti ar perkamo automobilio kaina adekvati?

**Užduotis:** Sukurti modelį, kurio pagalba būtų galima prognozuoti automobilio kainą pagal tam tikrus jo parametrus. Užduotį galima susiaurinti iki konkretaus gamintojo automobilių.

**Duomenys:** Duomenys yra saugomi automobilių portaluose, tokiuose kaip www.autoplius.lt, www.autogidas.lt, www.autobilis.lt ir pan. Norint sukurti duomenų rinkinį - reikia duomenis scrapinti.

**Galimas sprendimas:** Sukuriamas duomenų surinkimo įrankis (scraper), kurio pagalba sukuriamas apmokymo duomenų rinkinys. Turint duomenis - sprendžiamas regresijos uždavinys. Nustatomi parametrai, kurie bus naudojami kainos nustatymui, atliekamas duomenų išvalymas/ paruošimas, tada atliekamas duomenų modeliavimas.

**1. SCRAPE.**

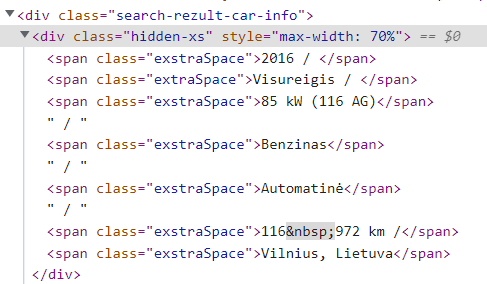
**Duomenys** modeliui nuscrape‘inti iš [www.autobilis.lt](http://www.autobilis.lt) 2023-05-02.

**Kodo aprašymas:**

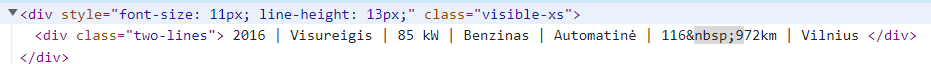
1. Importuojama biblioteka BeautifulSoup iš bs4 ir requests biblioteka.
2. Sukuriamas tuščias excel failas naudojant openpyxl biblioteką, ir sukuriamas naujas lapas (sheet) pavadinimu "Autobilis skelbimai". Excel failas pasirinktas todėl, kad tikrinantis kaip veikia kodas susitaupo duomenų konvertavimo laikas iš csv.
3. Į sheet pirmąją eilutę pridedamos reikiamos stulpelių pavadinimai.
4. Paleidžiamas ciklas, kuris iteruoja per puslapius nuo 1 iki 161. Tiek [www.autobilis.lt](http://www.autobilis.lt) svetainėje yra puslapių, kuriuose yra skelbimai.
5. Užkraunama puslapio HTML turinio informacija naudojant requests biblioteką ir nustatomas numatytasis tekstų analizavimo (parsing) metodas "html.parser" naudojant BeautifulSoup biblioteką.
6. Iš puslapio HTML turinio gaunami blokai (blokai yra HTML div elementai su klasės atributu "search-rezult-content").
7. Užkraunamas vidinis ciklas, kuris iteruoja per kiekvieną bloką.
8. Iš bloko gaunamos reikiamos informacijos apie automobilio markę, kainą, metus, kebulo tipą, variklio galią, kuro tipą, pavarų dėžės tipas, ir ridą, naudojant find() ir get\_text() metodus.
9. Gauta informacija patalpinama į sheet, naudojant sheet.append() metodą.
10. Jei gaunama klaida, vykdomas except blokas.
11. Išsaugojamas excel failas su visais surinktais duomenimis ir pabaigiama programa.

Gautas duomenų rinkinys iš 3108 eilučių.

**Sunkumai rašant kodą:**

1. HTML kode informacija apie automobilio parametrus yra sudėti į tą patį bloką su ta pačia klase - „'div', class\_="search-rezult-car-info", todėl jos „paėmimas“ galimas tik per teksto vietą, o jei informacija nėra suvesta tvarkingai, nuskaitant reikšmės susimaišo:

Dažniausiai nuskaitant informacija neteisingai „sukrisdavo“ į spalva ir rida stulpelius. Nusprendžiau atsisakyti spalvos kaip informacijos ir pasiėmiau kitą bloką, kur duomenys nusiskaitė kiek geriau:



Vis dėlto duomenis reikės tvarkyti, bet mažiau.

2. Skelbimai išdėstyti per 162 svetainės polapius, todėl reikėjo papildomo kodo, kad būtų galima iteruoti per visus polapius renkant informaciją.

3. Svetainėje yra išskirti TOP10 skelbimų, kurių div klasė yra kiek kitokia, kiek užtrukau kol supratau kodėl taip yra.

4. Ieškojau kitos svetainės, kur automobilių parametrai būtų sudėti į skirtingus div, tai radau [www.alio.lt](http://www.alio.lt), bet ten kilo problema, kad neina iteruoti per svetainės polapius, nes sistema tokia, kad scrolinant pasimato nauji skelbimai, bet URL nesikeičia, o scrape‘inant duomenis vis tiek pasiimdavau tik 25 skelbimus.

5. Nedaug svetainių leidžia scrape‘inti autoskelbimus.

**2. DUOMENŲ VALYMAS.**

1. Duomenų tvarkymo etapas:

1.1. Nuskaitomi duomenys iš failo "Autobilis skelbimai\_GALUTINIS.xlsx" naudojant pandas biblioteką.

1.2. Nuskaičius excel failą atsirado keistų simbolių, kaip \n, \xa, tarpai ir pan., todėl reikia išvalyti df. Duomenų stulpelių tipai patikrinami ciklu „for“. Jei stulpelio tipas yra „object“, tai naikinami nurodyti simboliai: '\xa0', 'km', ' kW', '€', '2+2'. Stulpelio reikšmės taip pat yra valomos naikinant tarpus, kurie galėjo likti po ankstesnių žingsnių.

1.3. Metai stulpelio reikšmės sutvarkomos naikinant simbolius '- ' ir po jo einančius skaičius, ir naudojamas tik pagrindinis metai reikšmės skaičius.

1.4. Rida ir Variklio\_galia stulpelių reikšmės yra patikrinamos ciklu „for“ ar jos yra skaičiai. Jei taip, jos lieka kaip skaičiai, jei ne, jos tampa 0 reikšmėmis.

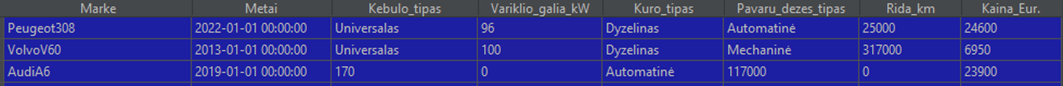
1.5. Metai stulpelis yra paverčiamas datetime tipo reikšme.

1.6. Kaina stulpelis yra paverčiamas int tipo reikšme.

1.7. Duomenų stulpelių pavadinimai yra atnaujinami, įrašant matavimo vnt. Stulpelių pavadinimai yra pakeičiami iš 'Variklio\_galia' į 'Variklio\_galia\_kW', 'Rida' į 'Rida\_km' ir 'Kaina' į 'Kaina\_Eur.'

2. Duomenų valymo etapas:

Scrape‘nant buvo gautos netinkamos kai kuriu stulpeliu reikšmės, nes skelbimuose ne visi laukeliai buvo užpildyti, pvz.



2.1. Sukuriamas df\_clean, kad nesugadinti pradinės duomenų rinkinio versijos.

2.2. Stulpeliuose 'Kebulo\_tipas', 'Kuro\_tipas', 'Pavaru\_dezes\_tipas' yra ieškoma skaitinių reikšmių, nes tai rodo, kad yra klaida:

sukuriamas tuščias sąrašas pavadinimu "klaidos", kuriame bus saugomos rastos klaidingos reikšmės; peržiūrimi stulpeliai 'Kebulo\_tipas', 'Kuro\_tipas', 'Pavaru\_dezes\_tipas' atskirai; iteruojant per eilutes tikrinama kurios reikšmės yra skaičiai; išrenkamas skaičių sąrašas, kuris saugomas kintamajame "klaidos\_col", naudojant funkciją "tolist()", galiausiai visos šios reikšmės pridedamos prie bendro sąrašo "klaidos" ir suskaičiuojamas.

2.3. Rastos klaidos pašalinamos iš "df\_clean" naudojant funkcijas "~" ir "dropna()". Duomenų rinkinys išvalomas nuo klaidingų reikšmių, ištrinant visas eilutes, kuriose yra bent viena klaidinga reikšmė, naudojant funkcijas isin().

2.4. Sutvarkomi indeksai, nes ištrynus eilutes, jie liko ne iš eilės.

2.5. Duomenų rinkinys išsaugomas .csv formatu.

**3. REGRESIJOS MODELIAI.**