Andmetehnika projekti raport

Write (around 500-700 words) describing the project's objectives, data sources, ETL process, visualization, and any challenges encountered and how they were addressed.

Projekti eesmärk on anda ülevaade New Yorgi koerahammustuste statistikast, mille käigus katsetada ETL protsessi andmete laadimisest, puhastamisest, transformatsioonist kuni visuaalide loomiseni. Lisaks leida rühma jaoks protsessis olevad keerulisemad kohad.
Projekti kohta on koostatud Githubi repositoorium (https://github.com/Rauno19/NY_hammustavad_kutsid_2025), millele on kõigil rühmaliikmetel juurdepääs. Repositoorium sisaldab projektiks vajalikke andmefaile, juhiseid, ülevaadet projektist (README.md), skripte, valmis töölauda ning muid jooniseid.

Andmeallikaks on andmefail leheküljelt:
https://data.cityofnewyork.us/Health/DOHMH-Dog-Bite-Data/rsgh-akpg/about_data
Failis on veerud, mis vajavad puhastamist ja transformatsioone. Kokku on algses andmetabelis 9 veergu. Puhastamiseks on kasutatud nii OpenRefine kui ka Microsoft Excelit. Andmestiku tegi keerukamaks see, et suurem osa veerge on vabavastuselised ja võivad eeldada küll üht tüüpi vastust, aga lahtri täitjal on sama asja kirjutamiseks mitu võimalust (nt vanuse lahter (AGE) võib leida erinevaid variante 4-aastase koera märkimiseks: 4, 4Y, 4 YR, 4YR). Andmestiku keerulisus peitub lisaks vanusele tunnuses koerte tõud (Breed), mida saab väljendada veel rohkemat moodi. Algses andmestikus oli 1887 erineva koeratõu kirjaviisi, lõplikusse andmefaili jäi alles u 400 koeratõugu.
Algandmetes on veerg UniqueID, kuid tegelikult polnud tegemist unikaalse ID-ga. Esines ka kirjavigu, kus näiteks numbriveergu (zipcode/postiindeks) on null asemel sisestatud suur O või sisestatud postiindeks valesti või vigadega (11208 asemel 111208).
Kasutades MS Excelit on lisatud andmestikku veel lisaveeruks DogSize_v4, mis annab lisainfot koera suuruse kohta, DogType_v4, annab infot koera funktsiooni kohta (aretuseesmärk), ZipCode_v4 ja GeoPoint_v4, mis on vastavalt puhastatud algne ZipCode ja geograafiline pikkus- ja laiuskraad. Andmestiku veergude arv kasvas 12ni (kui arvestada, et geograafilised koordinaadid on küll ühes veerus, aga kasutatakse eraldi veergudena, siis on veergude arv 13).

Meie projekti oluliseks fookuseks kujunes tulenevalt andmestiku olemusest andmete puhastamine. Välistes allikatest lisasime täiendavalt teavet, et suurendada andmeanalüüsi võimalusi (DogSize, DogType, GeoPoint). GeoPoint-andmete lisamine annab võimalused luua täpsemad kaardivisualiseeringud.

ETL protsess näeb meie projekti puhul välja järgmine:

1. Andmete allalaadimine veebilehelt
2. Andmete puhastamine (OpenRefine, Excel)
3. Olemasolevatele andmetele täiendava info lisamine (MS_Excel).
4. Andmetöötluseks vajalike keskkondade allalaadimine ja loomine (vt “How-to_build_superset”).
5. Andmefailide transformeerimine (Pythoni skriptid devcontaineri sees) andmebaasile sobivasse formaati (.parquet) (vt “parquet_failiks_pythoni_kood.md”)
6. Andmete visualiseerimine Supersetis (vt “Data_visualisation_in_Superset.md”).
7. Kuna Apache Supersetis ei saanud teha päris kõiki tegevusi nii nagu me algselt lootsime (peamiselt geograafiliste andmete kuvamiseks), on kasutatud/katsetatud lisaks veel MS Exceli (nt 3D Maps), R-i, Google Colabi võimalusi.

Andmete visualiseerimine

Andmete visualiseerimiseks kasutati Apache Superseti, MS Excelit, R-Studiot

Kitsaskohad projektis

Andmestikust lähtuvalt kulus märkimisväärselt palju aega andmete puhastamisele, OpenRefine laseb andmestikku puhastada teatud piirini ja täiesti korda seal seda ei saanudki. Andmete puhastamist OpenRefine'is tehti mitu ringi.

Excel pole ka mugav töövahend andmete puhastamiseks. Näiteks palju probleeme tekitab excelis automaatne teisendus kuupäevaks kui numbri kümnendkoha eraldaja on punkt või kui andmed on csv-s salvestatud nii, et veerueraldaja on koma, aga sama võib ühe veeru kirje ka komakohti sisaldada.

Andmestiku teeb keeruliseks veel ka see, et on palju "unknown" andmeid või täitmata lahtreid. Peab vaatama kuidas ja mida analüüsida saab, otsustada, kas puudulikke ridu eemaldada või püüda "paranda" (nt kui zipcode puudu, siis linnaosa nime järgi sai lisada linnaosa keskkoha postiindeksi).

Geomapi tegemisel ilmnes takistus kaardi loomisel Supersetis, mis nõudis Mapboxi API key'd, et kaardi piirjooned oleks ka kuvatud, muidu olid andmed punktidena koordinaatide põhjal vaid valgele taustale kantud. Kaardi piirjoonte kuvamiseks oli ChatGPT soovitus kasutada Mapboxi default public tokenit, jooksutades vastavat koodi VS Code'i terminalis, tekitades olemasoleva superset image'i peale veel teise konteineri Dockeris. Paraku pärast korduvaid katseid seda aktiveerida ei õnnestunud, näidates korduvalt veateateid, mis viitasid Dockerfile'i loomisel kasutatud kasutajale.

Probleemiks kujunes ka see, geomapi kasutamiseks, vajalike uuenduste tegemisel läksid Supersetis tehtud graafikud kaduma.
Sellest ka õppetund varukoopiate tegemise vajalikkuse kohta.

Kohati oli raskusi ka GitHubi kasutamisega

Kuidas probleemidega toime tuldi

Andmestiku puhastamisega seotud probleemide lahendamiseks kulutati rohkem aega ja kasutati lisaks OpenRefinele ka Excelit. Paljudest "unknown" lahtritest päris lahti ei saadudki, kuid neid prooviti visualiseerimisel võimalikult vähe kasutada.
Kuna geomapi saadi lõpuks Supersetis korda on töölaual ka geograafilisi andmeid kasutatud. Lisaks on ruumiliste andmete kuvamiseks kasutatud R-Studiot ja MS Excelit, millega tehtud kaardid on githubi repositooriumisse samuti kaasa pandud (vt "valmis_toolauad_joonised").