NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Engineering

Department of Ocean Operations and Civil Engineering

Integration of GIS and Python for Sewer Condition Assessment

Tutorial 3

Lam Van Nguyen

Razak Seidu

Water and Environmental Engineering Group

NTNU Ålesund

12/2022

Objectives of this tutorial

The overall aim of this tutorial is to provide a user with a step-by-step guideline for using GIS to process geospatial data for sewer condition assessment. Python packages to implement machine learning models for predicting sewer conditions were also introduced in this tutorial. A small part of Ålesund city is selected as a study area in this tutorial.

Table of Contents

1. Introduction

1.1. Sewer Condition Assessment

Assessment of the structural condition of sewer pipes is one of the critical steps in asset management and support investment decisions; therefore, structural condition models with high accuracy are important that can help the utility managers and other authorities correctly assess the current condition of the sewage network and effectively initiate maintenance and rehabilitation strategies.

Sewer deterioration is a complex process affected by many factors including *physical*, *environmental*, and *operational* factors. The availability of data significantly depends on a specific study area and database that local water agencies manage. In this tutorial, we only focus on processing some of the physical and environmental factors which are achievable. Processing the operational factors is not included in this tutorial due to unavailable data.

Structural condition models can be classified into physical, statistical, and machine learning (**Figure 1-1**). The advantages and disadvantages of each above mathematical model can be found in (Nguyen et al., 2022). In this tutorial, we concentrate on implementing sewer deterioration models using machine learning algorithms which are state-of-the-art techniques.

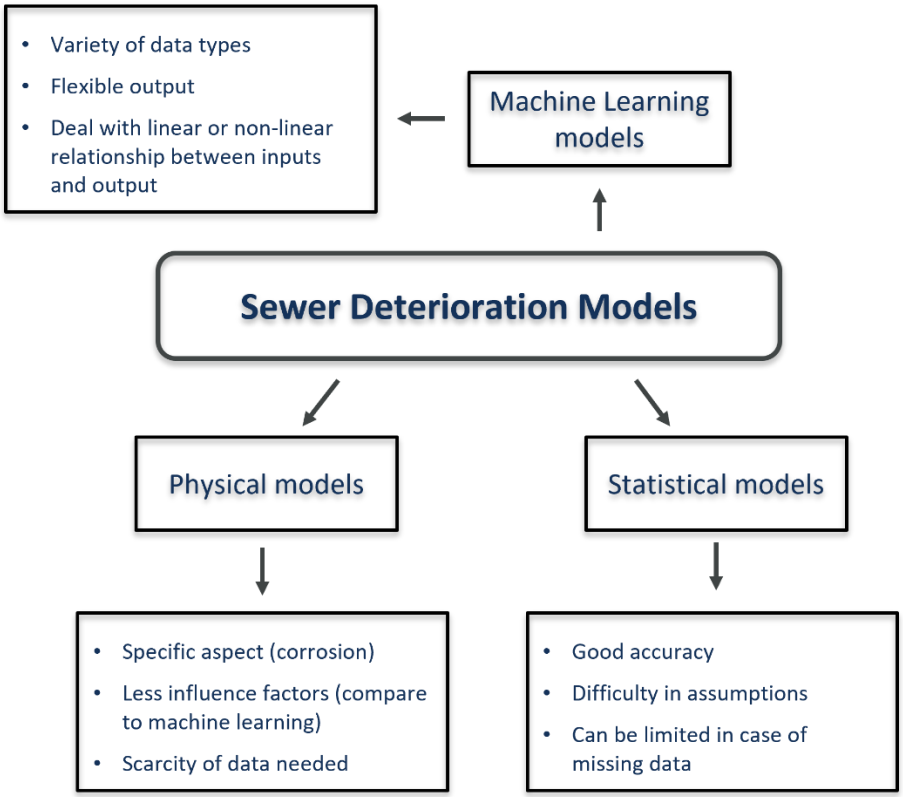


Figure 1-1. Sewer condition model classification

1.2. QGIS Installation

QGIS is one of the most widely used Geographic Information System (GIS) software, likely ArcMap, or ArcGIS Pro. QGIS is a free and open-source cross-platform desktop GIS application that supports viewing, editing, printing, and analysis of geospatial data.

To download QGIS, the user can follow the below steps:

- *Step 1:* Visit this link: https://www.qgis.org/en/site/ and download QGIS software (**Figure 1-2**). Documentations for using QGIS can be found at the above address.

Figure 1-2. QGIS downloading homepage

- *Step 2:* Download the correct QGIS version depending on the user's operating system (**Figure 1-3**).

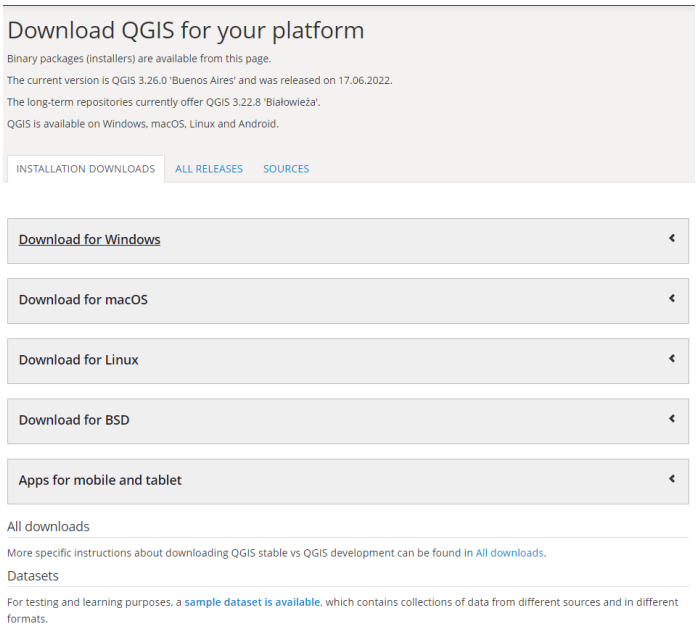


Figure 1-3. Multi-platform QGIS

- *Step 3:* Select to download the QGIS. In this tutorial, we used QGIS version 3.22 (the most stable long-term release version) (**Figure 1-4**).

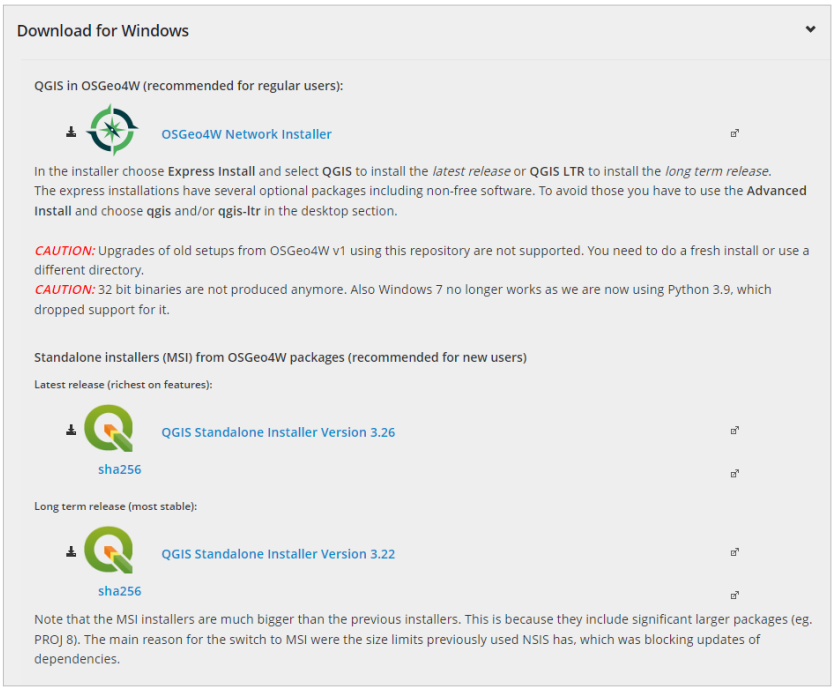


Figure 1-4. Selecting the wanted QGIS version to download

- *Step 4:* Store the downloaded QGIS software on the PC or hard disk. Install QGIS by double click on the downloaded item.

- *Step 5:* Complete the installation process by clicking the *"Next"* button in the next steps (**Figure 1-5**).

Figure 1-5. QGIS installation process

➢ *Step 6:* Finish installation and start a new QGIS project (**Figure 1-6**).

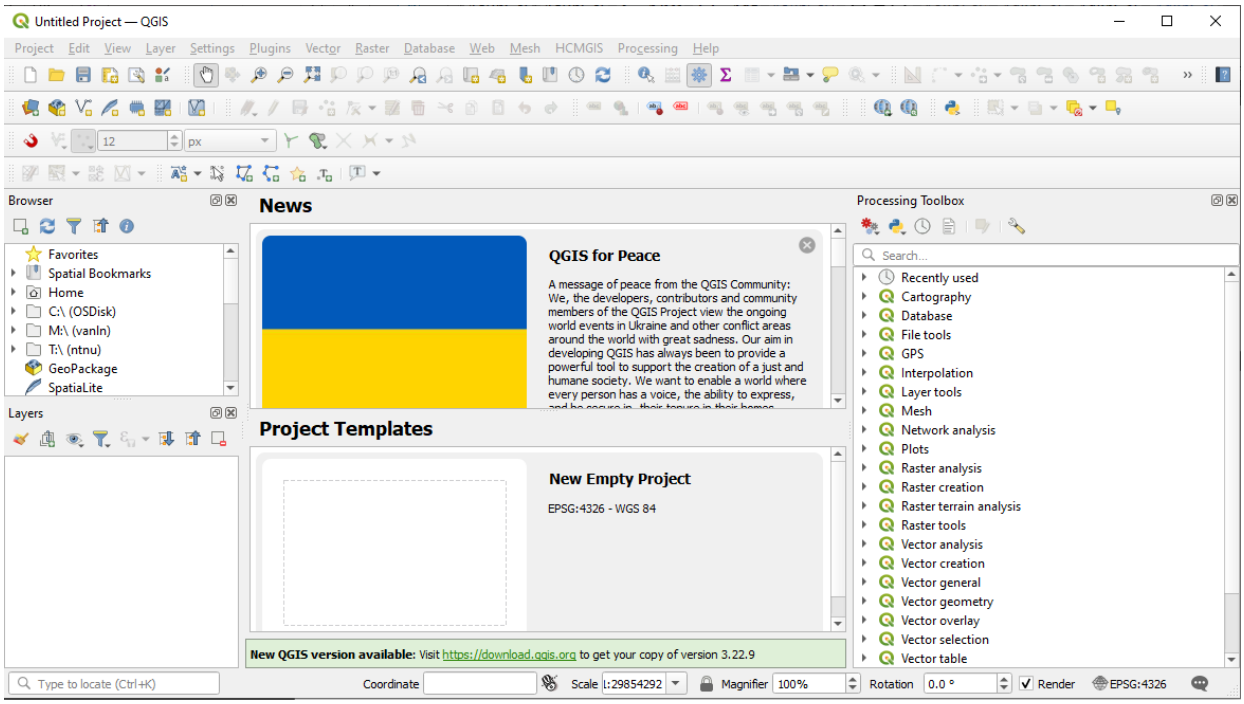


Figure 1-6. Opening a new QGIS project

1.3. Installing Python Environment

Python environment is critical for implementing machine learning models. To install python, users can do it in several ways. The simplest way to install python is to use Anaconda:

➢ *Step 1:* Download Anaconda from the homepage address: https://www.anaconda.com/products/distribution (**Figure 1-7**).

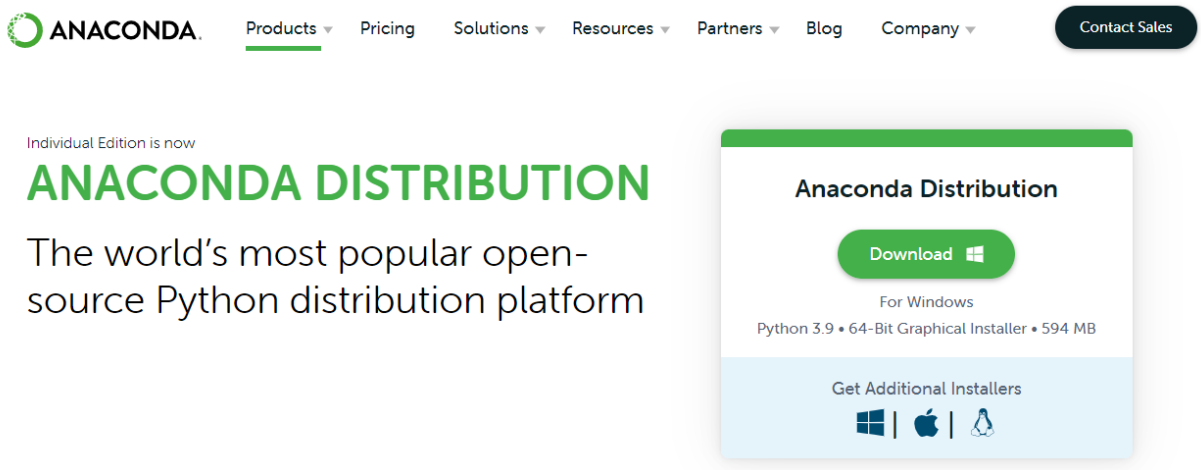


Figure 1-7. Anaconda homepage

➢ *Step 2:* Install Anaconda (**Figure 1-8**).

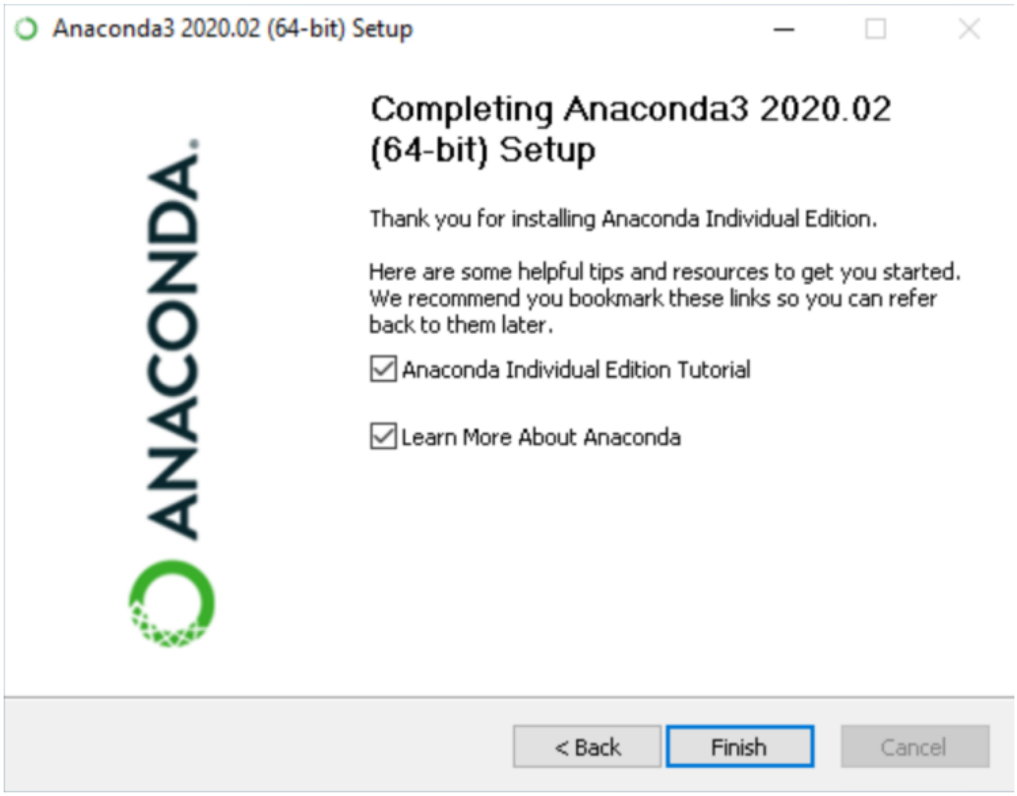


Figure 1-8. Installing Anaconda

2. Data Preparation

2.1. Sewer Network Preparation

In this section, we will introduce to the user two ways to create a sewer network using QGIS: 1) create components manually, and 2) import components automatically from CSV files.

- Creating a network manually is a suitable way for people who work with small sewer networks or for people who do not have an existing sewer network.
- Importing components from CSV files is a better choice when the attributes of the components are available in tabular format.

a. Setting up Background Image

A background image in the correct coordinate system is very critical for sewer network design no matter how it was created manually or automatically. For the manual design process, the image support properly putting sewer components (manholes or pipes) in the reality. For the automated importing process, this image is used to check the accuracy of the imported sewer's components.

In QGIS, to displace the background image, some plugins (such as HCMGIS or

QuickMapServices) can be installed. The plugin HCMGIS is used in this tutorial. The process for installing a plugin is represented in **Figure 2-1**.

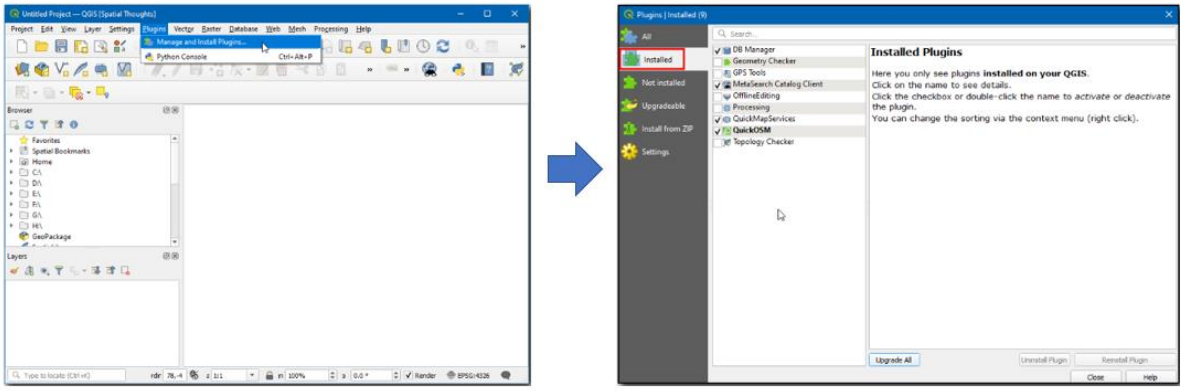


Figure 2-1. Installing a plugin in QGIS

Type the keyword *"HCMGIS"* (or *"QuickMapServices"*) into the search address to download and install HCMGIS (or QuickMapServices) plugin. To insert a background image into QGIS, select the wanted type of background image from the list (**Figure 2-2**).

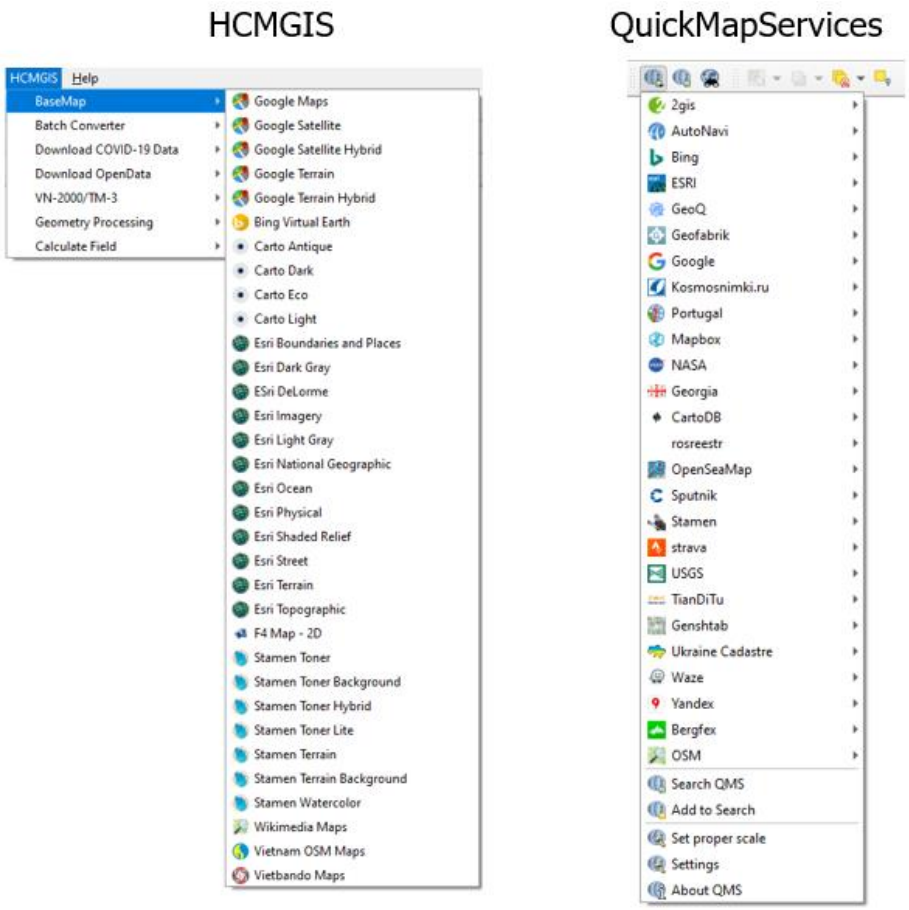


Figure 2-2. The interface of the HCMGIS and QuickMapServices plugins

The background image in QGIS is presented in **Figure 2-3**.

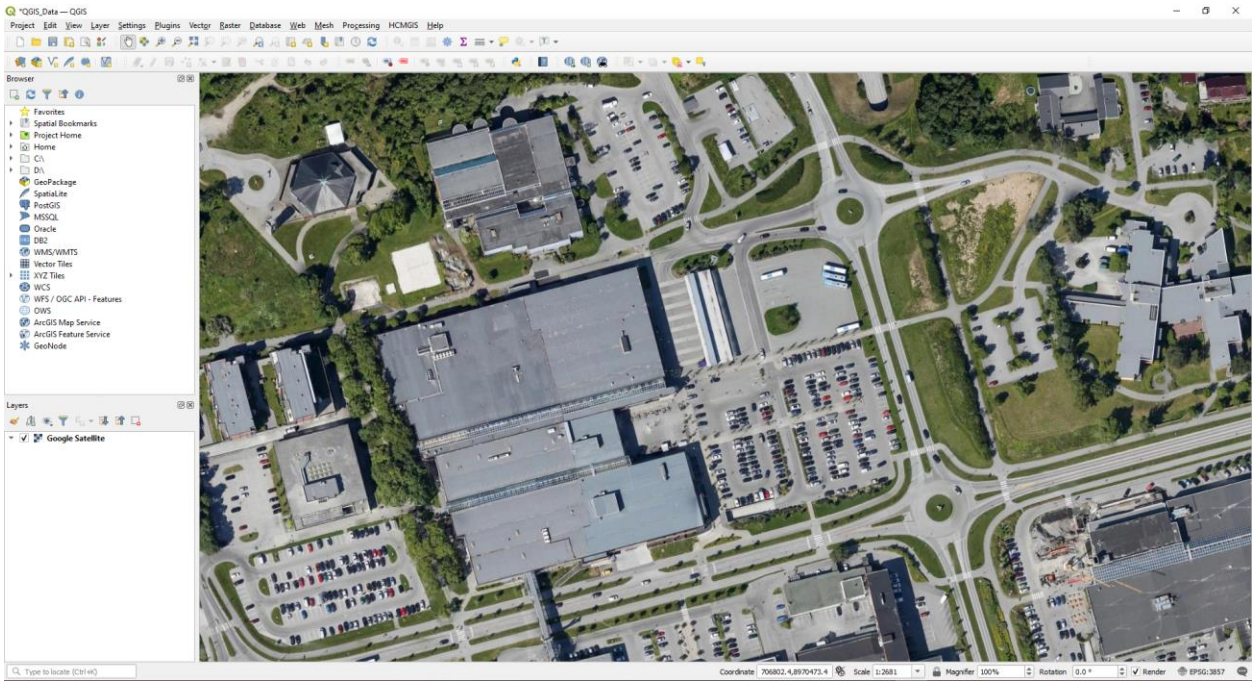


Figure 2-3. Background image in QGIS

b. Creating Sewer Pipe Layer and Assigning Physical Factors

❖ *Creating Sewer Pipes manually*

➢ *Step 1:* Before starting a new project in QGIS, the Coordinate Reference System (CRS) should be defined. The steps for defining initial CRS in QGIS are shown in **Figure 2-4**.

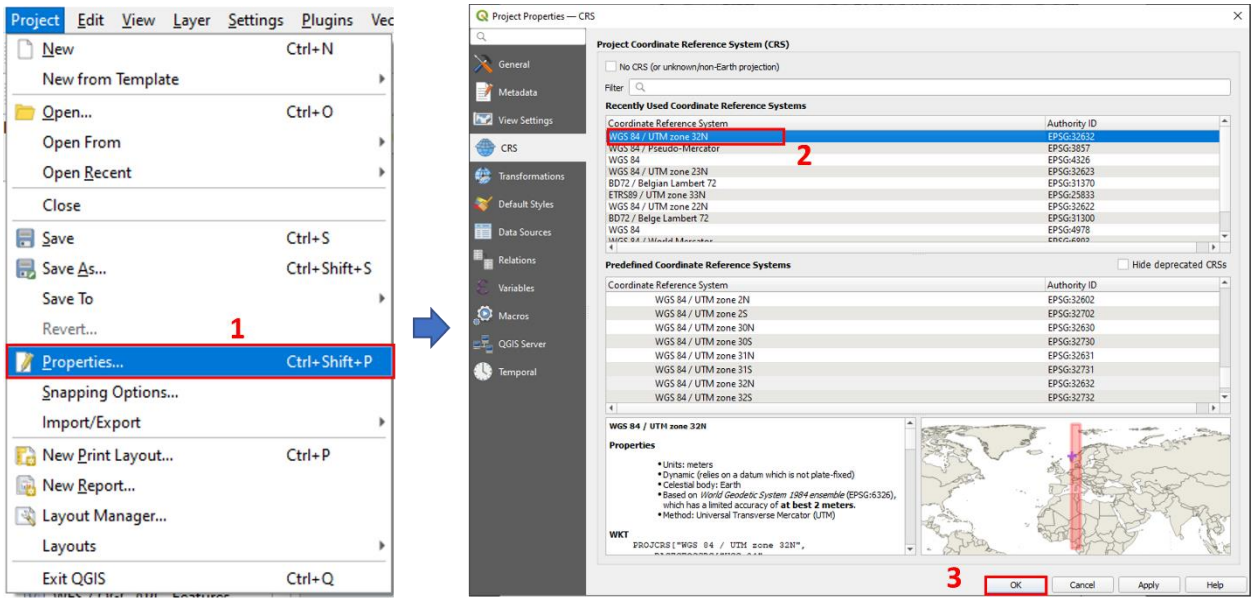


Figure 2-4. Defining a coordinate system in QGIS

➢ *Step 2:* Create a new layer in QGIS: *Layer* → *Create layer* → *New Shapefile Layer*… (**Figure 2-5**a).

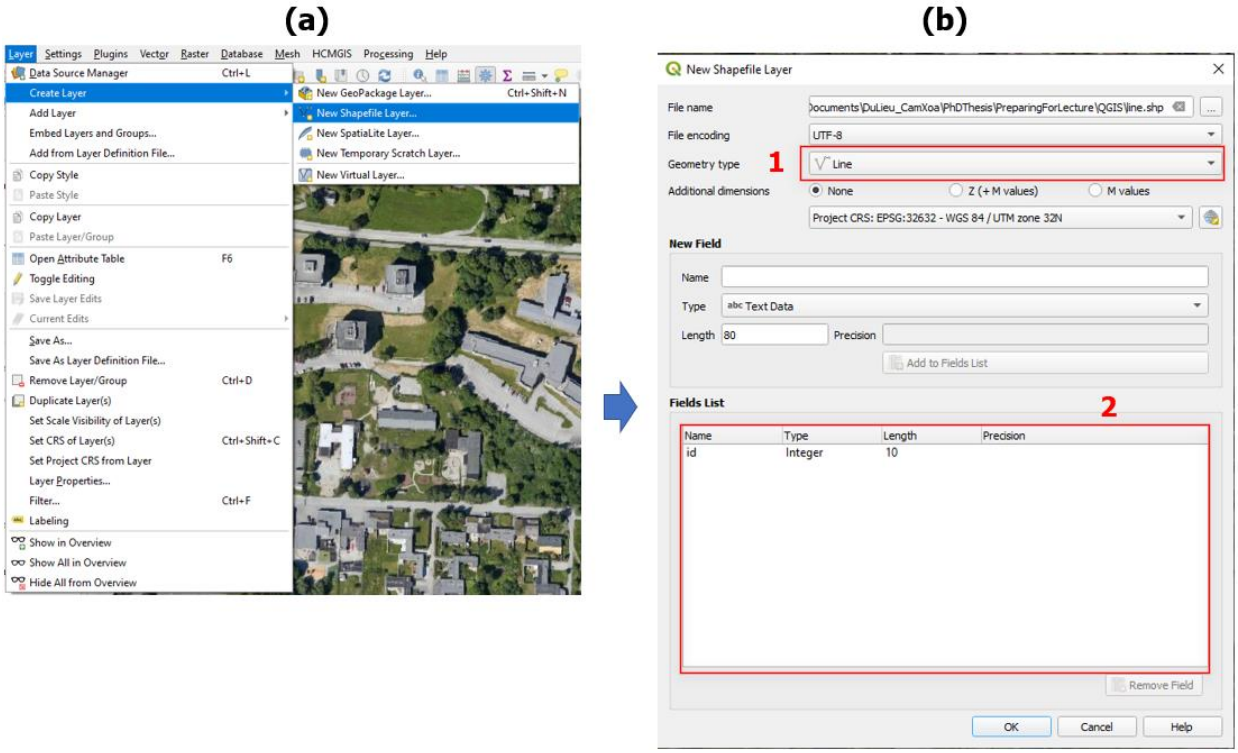


Figure 2-5. Creating a pipe layer in QGIS

- *Step 3:* Assign wanted properties for the sewer pipe. It is noticed that the *"Geometry type"* is selected as *"Line"* (**Figure 2-5**b).
- *Step 4:* Select created pipe layer and activate the edit mode in QGIS (**Figure 2-6**).

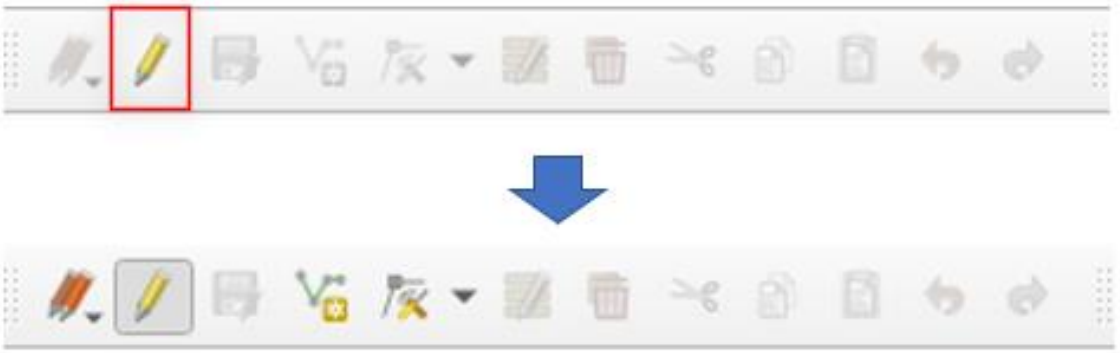


Figure 2-6. Editing mode in QGIS

- *Step 5:* Select the *"Add Line Feature"* button and pick points where the pipe goes through on the map by clicking the left mouse button. A sewer pipe can be created by a simple line (between two points) or a complex line (between multi-points). Click the right mouse button to finish creating a particular sewer pipe. After that, a *"Feature Attributes"* dialog appears to allow the user to change/assign the pipe's attributes (**Figure 2-7**). Click *"OK"* to execute the function.

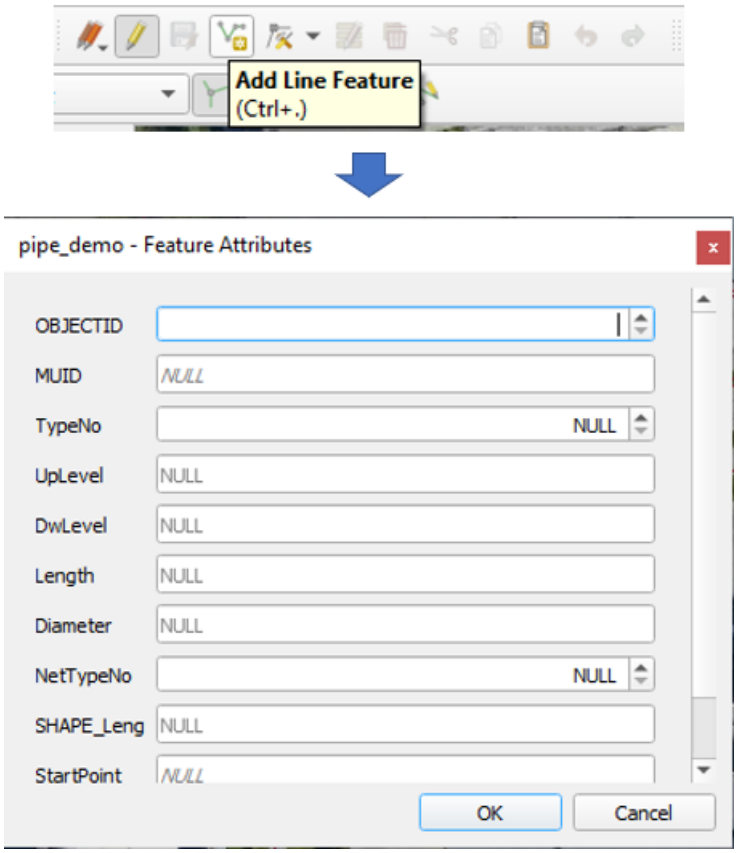


Figure 2-7. Adding a new sewer pipe and assigning its attributes in QGIS

- *Step 6:* Check created sewer pipelines on the map (**Figure 2-8**).

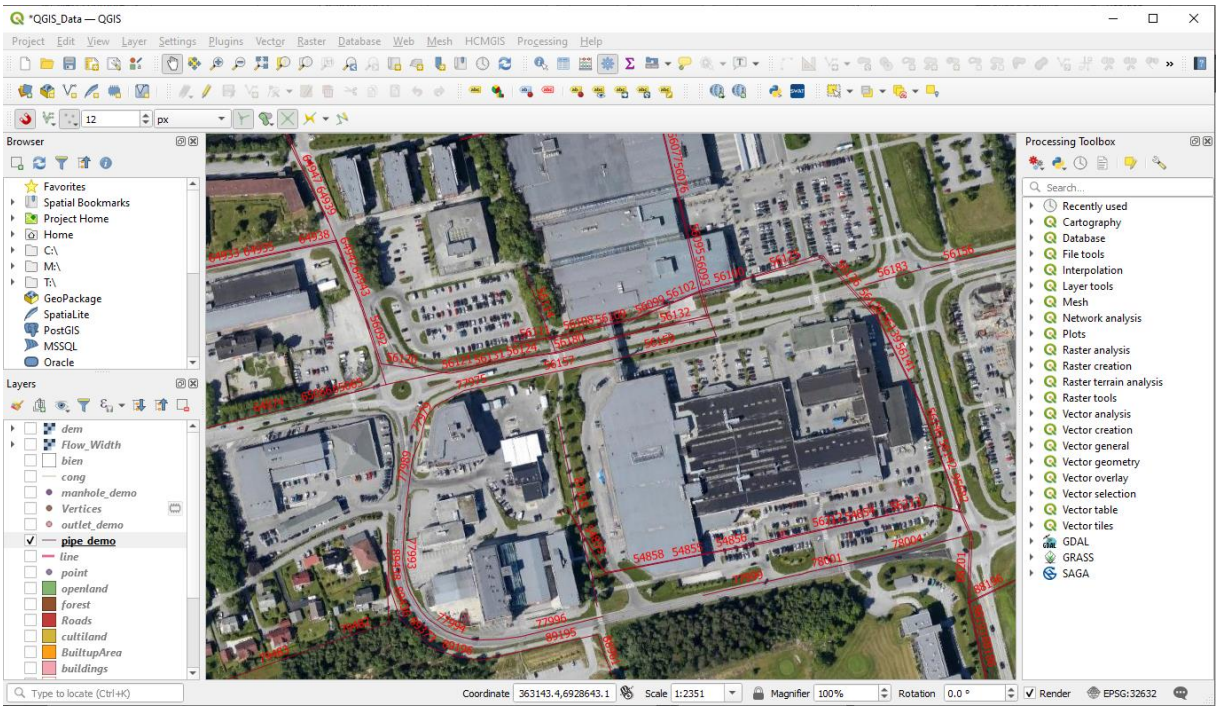


Figure 2-8. Sewer pipelines on the map

- *Step 7:* To adjust the attributes of the pipes, the user must open the attribute table of the

pipe layer (**Figure 2-9**).

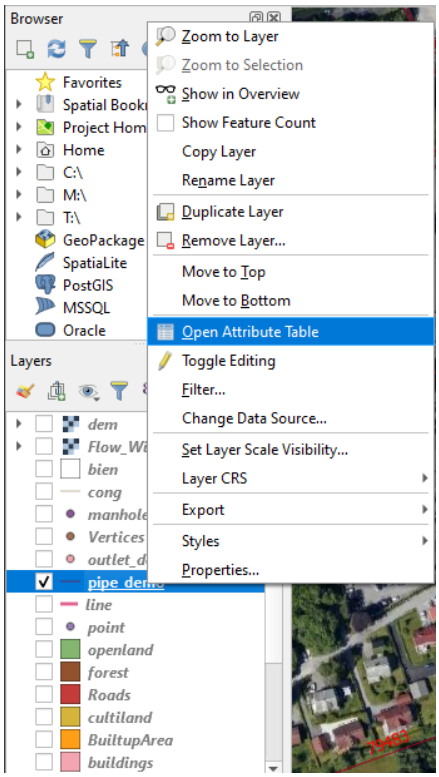


Figure 2-9. Sewer pipelines' attribute table in QGIS

❖ *Importing Sewer Pipelines from CSV file*

An example of a sewer pipe file is shown in **Figure 2-10**. It is worth noticing that all points of one sewer pipe (including the start point, endpoint, and vertices points) must have the same name.

	A	B	C	D	E	F	G
1	LSID	LNO	X	Y	Z	Q_XY	Q_Z
2	76	1	355479.8	6929662	15.48	1	1
3	76	2	355476.8	6929676	15	1	1
4	77	1	355476.8	6929676	14.48	1	1
5	77	2	355474.4	6929688	7.79	1	1
6	78	1	355474.4	6929688	7.79	1	1
7	78	2	355415.5	6929692	6.68	1	1
8	79	1	355415.5	6929692	6.68	1	1
9	79	2	355403	6929707	6.44	1	1
10	82	1	355475.6	6929676	14.5	1	1
11	82	2	355473	6929687	7.37	1	1
12	83	1	355473	6929687	7.16	1	1
13	83	2	355414.8	6929692	6.04	1	1
14	84	1	355414.8	6929692	6.04	1	1
15	84	2	355401.4	6929707	5.7	1	1
16	86	1	355481.3	6929660	15.89	1	1
17	86	2	355475.6	6929676	15.28	1	1
18	87	1	355514.4	6929673	14.99	1	1
19	87	2	355476.8	6929676	14.48	1	1

Figure 2-10. Sewer pipe file example

The first step in creating a pipe in QGIS is to import all vertices points into QGIS (**Figure 2-11**). The next step is to join all points with the same name into one line. To merge points into one line, the user can use the function *"Points to path"* in QGIS (steps 3 and 4 in **Figure 2-12**) from the *"Processing Toolbox"* dialog (step 2 in **Figure 2-12**). Click *"Run"* to execute the function. If this dialog is disabled, the user can enable it by selecting *Processing* → *Toolbox* from the main menu (step 1 in **Figure 2-12**).

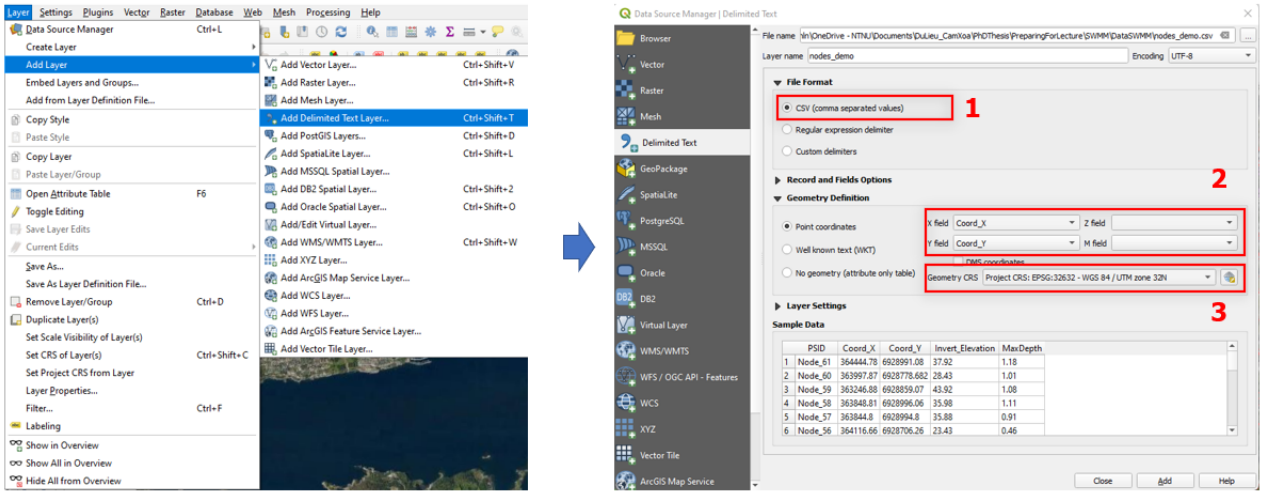


Figure 2-11. Importing points from the CSV file

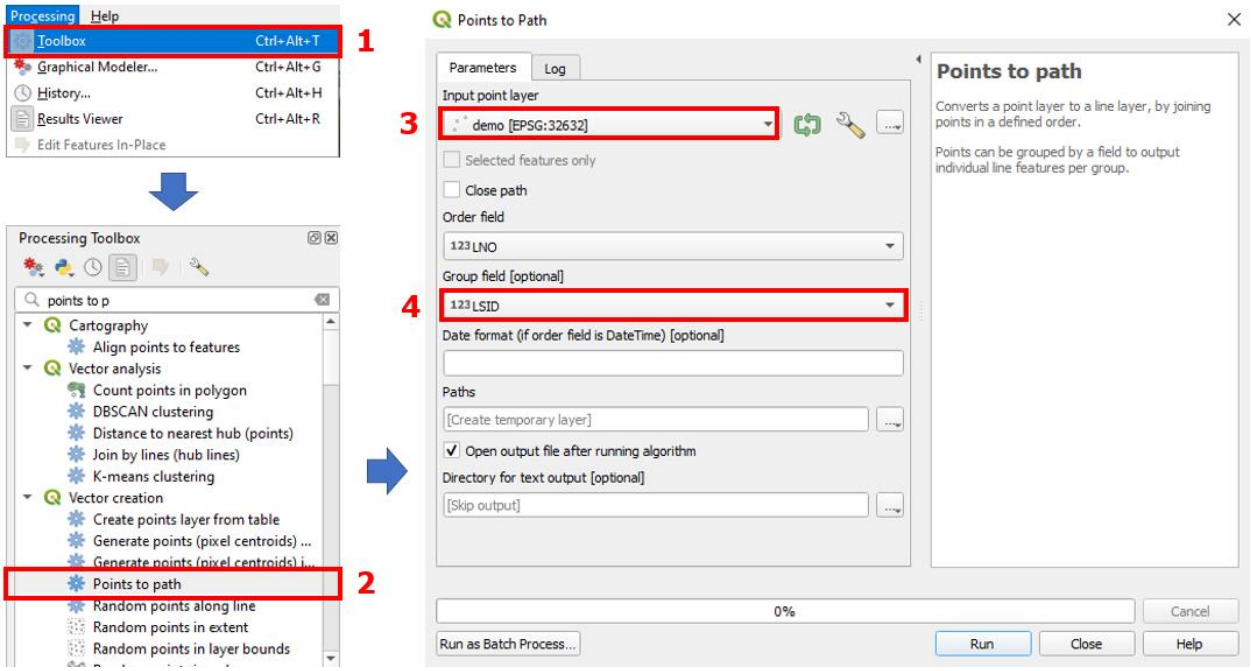


Figure 2-12. Converting points to a line in QGIS

Check the links on the map (**Figure 2-13**).

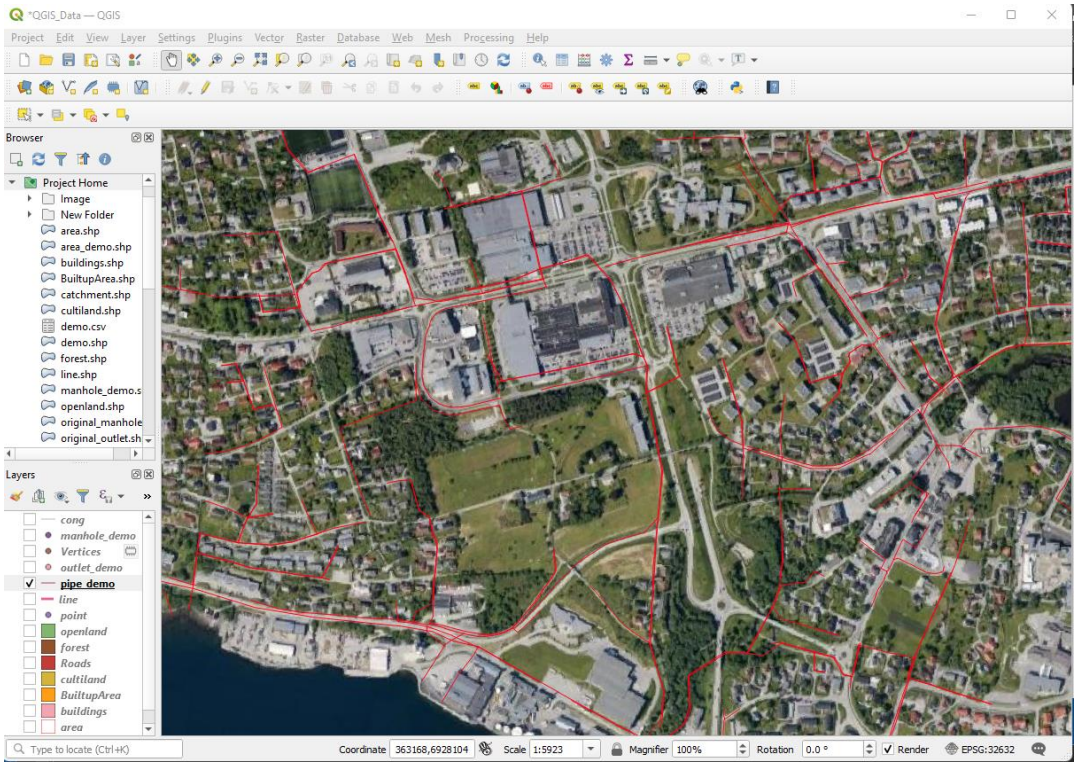


Figure 2-13. Sewer pipes in QGIS

Sewer pipes created by the above steps do not have any attributes. If the user has a CSV file that contains the sewer's properties like in **Figure 2-14**, this file can be joined with the sewers in **Figure 2-13**. The physic-related characteristics of sewer pipes (e.g., diameter, material, network type, etc.,) are stored and managed by local water utilities and they are easily obtained. All needed information is easily merged into one CSV file.

	A	B	C	D	E	F	G
1	LSID	FCODE	DIM	LENGTH	FROM_PSID	TO_PSID	Roughness
2	981	AF	0.23	46.93	151886	981_83019	0.020982467
3	1187	OV	0.16	25.06	459	54676	0.023396565
4	1221	SP	0.23	11.51	55137	1221_Outlet	0.024614898
5	1222	AF	0.3	66.61	472	54956	0.024749715
6	1229	SP	0.12	69.31	596	56495	0.02525517
7	1231	OV	0.15	2.85	1231_Inlet	153852	0.024614898
8	1272	SP	0.15	18.63	649	55950	0.02525517
9	1280	SP	0.15	16.14	653	56516	0.02525517
10	1328	OV	0.2	34.56	733	732	0.024405479
11	1330	OV	0.2	2.1	56353	152323	0.024405479
12	1341	AF	0.3	26.25	772	54663	0.022710703
13	1345	SP	0.16	15.98	775	89074	0.021309238
14	1346	SP	0.16	29.82	776	775	0.021309238
15	1358	OV	0.3	61.81	863	868	0.024405479
16	1363	SP	0.23	33.2	901	902	0.023549279
17	4956	OV	0.16	23.56	459	153455	0.023396565
18	4970	OV	0.2	23.83	1812	1813	0.021807907

Figure 2-14. Sewers' attributes CSV file

To join a CSV file with a sewer pipe, the user needs to do the following steps:

- *Step 1:* Add a CSV file into QGIS as a layer (**Figure 2-15**a). In this step, the CSV file can be loaded as a simple table (the *"No geometry"* option is selected) (**Figure 2-15**b). Select the *"Add"* button to add the CSV file as a layer.

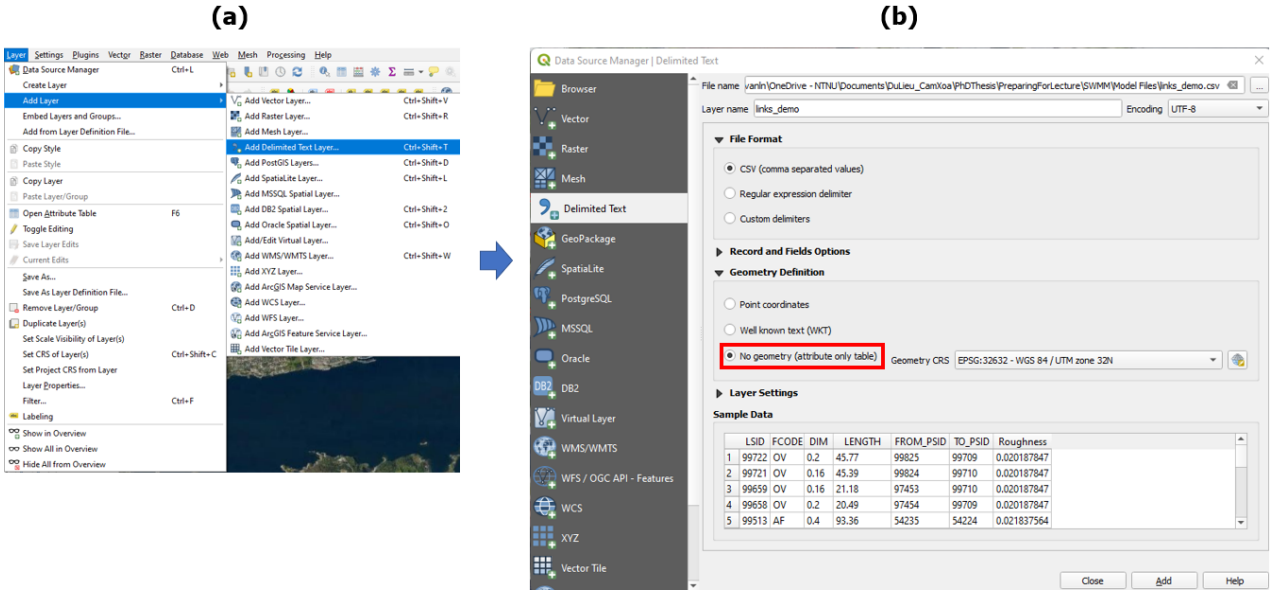


Figure 2-15. Adding a CSV file to QGIS (no geometry)

- *Step 2:* Open the *"Layer Properties"* dialog (step 1 in **Figure 2-16**), add the above CSV layer (steps 2 and 3 in **Figure 2-16**), select the attribute to join (step 4 in **Figure 2-16**), and fields to join (step 5 in **Figure 2-16**).

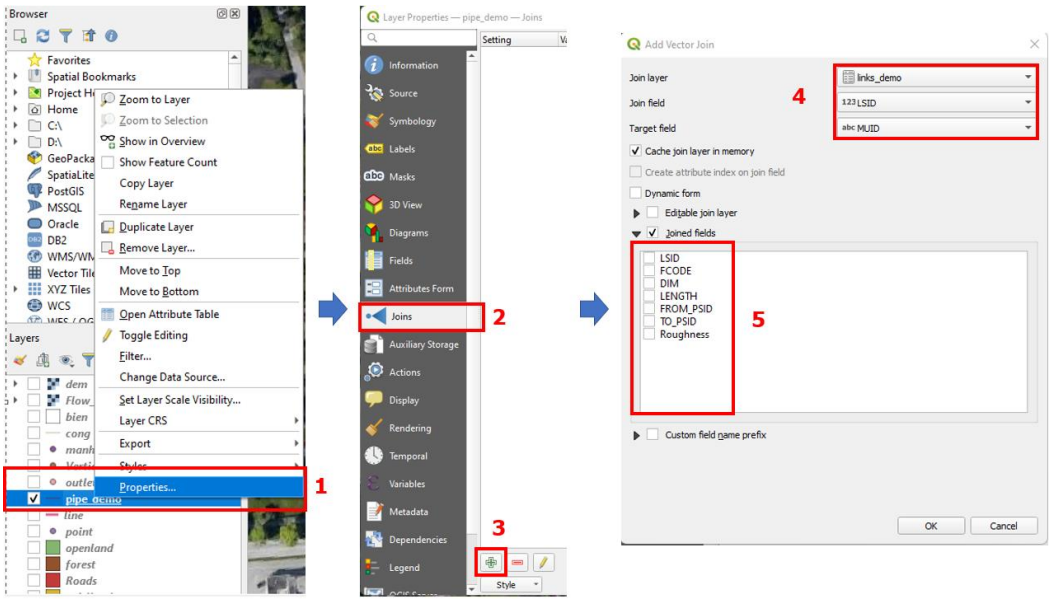
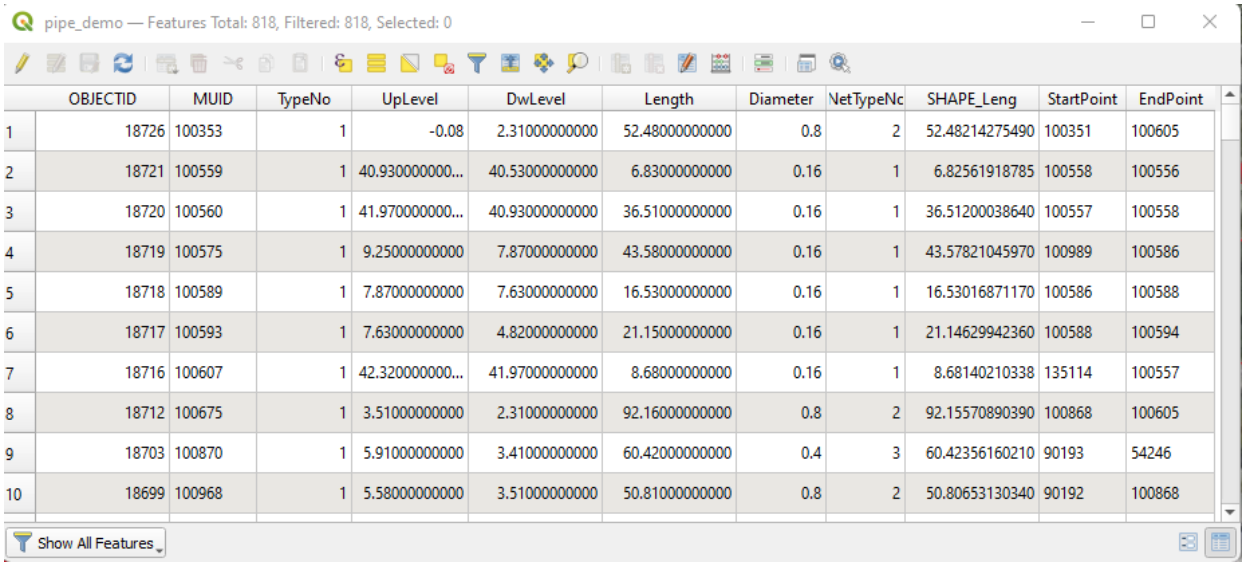


Figure 2-16. Joining attributes in QGIS

- *Step 3:* Open the attribute table to check the attributes of sewers (**Figure 2-17**).



	OBJECTID	MUID	TypeNo	UpLevel	DwLevel	Length	Diameter	NetTypeNc	SHAPE_Leng	StartPoint	EndPoint
1	18726	100353	1	-0.08	2.31000000000	52.48000000000	0.8	2	52.48214275490	100351	100605
2	18721	100559	1	40.930000000...	40.53000000000	6.83000000000	0.16	1	6.82561918785	100558	100556
3	18720	100560	1	41.970000000...	40.93000000000	36.51000000000	0.16	1	36.51200038640	100557	100558
4	18719	100575	1	9.25000000000	7.87000000000	43.58000000000	0.16	1	43.57821045970	100989	100586
5	18718	100589	1	7.87000000000	7.63000000000	16.53000000000	0.16	1	16.53016871170	100586	100588
6	18717	100593	1	7.63000000000	4.82000000000	21.15000000000	0.16	1	21.14629942360	100588	100594
7	18716	100607	1	42.320000000...	41.97000000000	8.68000000000	0.16	1	8.68140210338	135114	100557
8	18712	100675	1	3.51000000000	2.31000000000	92.16000000000	0.8	2	92.15570890390	100868	100605
9	18703	100870	1	5.91000000000	3.41000000000	60.42000000000	0.4	3	60.42356160210	90193	54246
10	18699	100968	1	5.58000000000	3.51000000000	50.81000000000	0.8	2	50.80653130340	90192	100868

Figure 2-17. Sewers' attribute table

The user can further calculate some geometry-related properties such as coordinates of points, area of polygons, length of pipes, etc., and store corresponding values in the attribute table.

2.2. Physical Factors Computation

Physical factors included, but not limited to, age, diameter, depth, slope, length, pipe type, material, network type, pipe form, and connection type. Among them, the depth and slope of each sewer pipe need to be computed, other remaining factors are identified for each pipe and they can be easily assigned for each pipe as the same process in **Figure 2-15** and **Figure 2-16**. In this tutorial, the age of the sewer pipes was calculated as the difference between the installation year and the inspection year.

a. Downloading high-resolution Digital Elevation Model (DEM)

The depth and slope of the sewer pipe can be computed from a Digital Elevation Model (DEM) that is obtained from the ***Høydedata*** portal (https://hoydedata.no/LaserInnsyn/) with pixels of 1 m×1 m spatial resolution. The steps for downloading are presented as follows:

- *Step 1:* Access the address https://hoydedata.no/LaserInnsyn/. Select the "*Download*" tab to set up parameters (step 1 in **Figure 2-18**).

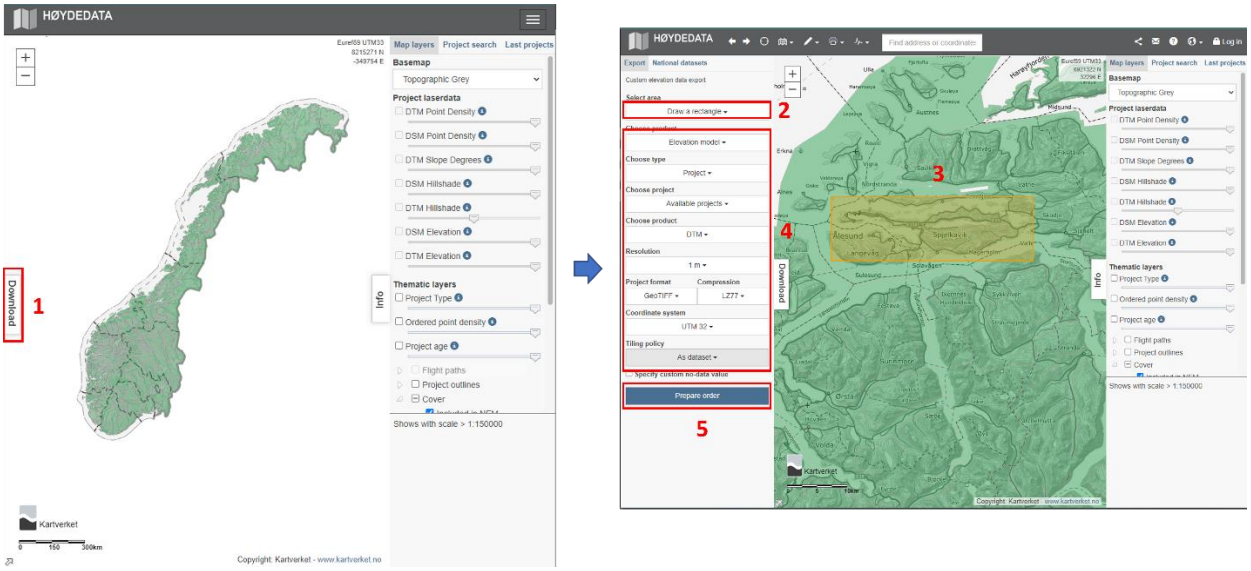


Figure 2-18. Downloading the DEM

- *Step 2:* Select the interested area (steps 2 and 4 in **Figure 2-18**), set up parameters (type of data, spatial resolution, and coordinate system) (step 4 in **Figure 2-18**), and download the data (step 5 in **Figure 2-18**).
- *Step 3:* Import downloaded data by simply dragging and dropping it into QGIS.

b. Computing slope

Because the slope of a particular pipe is changing along its length. In this tutorial, the slope of each pipe is an average change of the height in degree between the start and endpoint of the corresponding pipe. The slope can be computed as follows:

- *Step 1:* Extract the start point and end point of each pipe by using the function "*Extract specific vertices*" in QGIS (step 1 **Figure 2-19**).
- *Step 2:* Select the wanted pipe layer to extract (step 2 in **Figure 2-19**), and set up values of "0" and "-1" to extract the start point and end point of the pipe, respectively (step 3 in **Figure 2-19**). Select the location of the output file (step 4 in **Figure 2-19**). Click *"Run"* to execute the function.

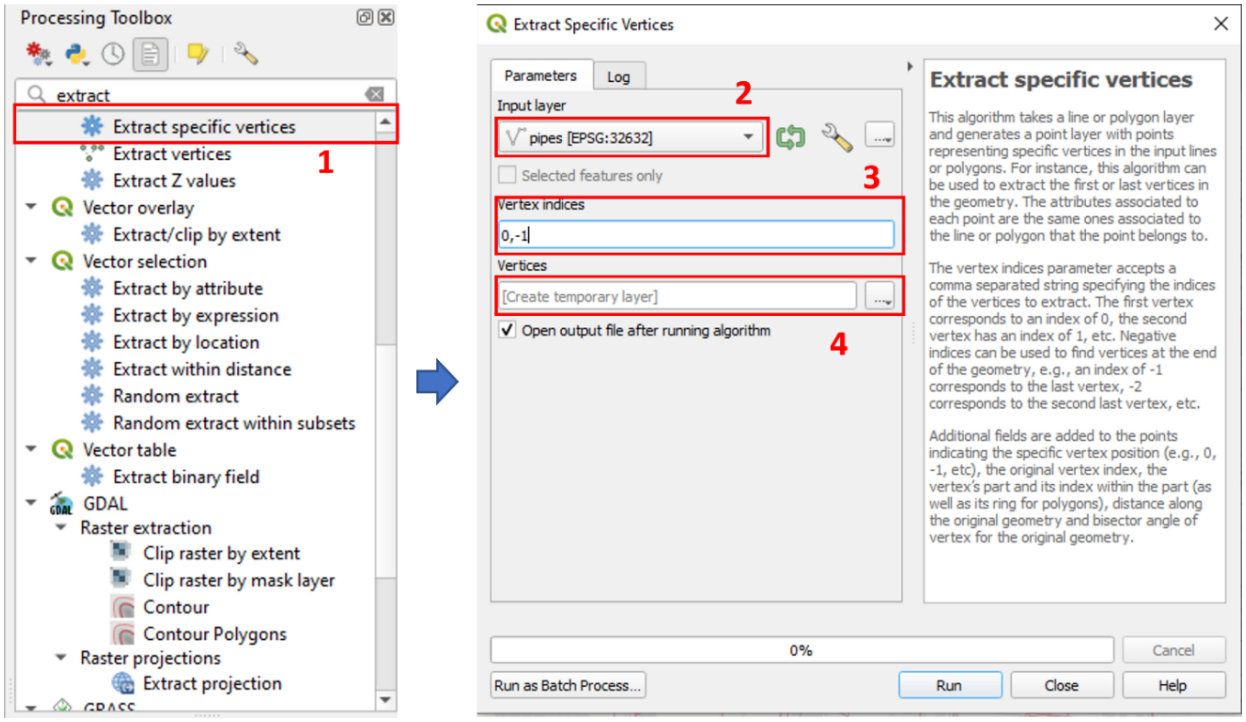


Figure 2-19. Extracting specific vertices of pipe

- *Step 3:* Check the start point and end point with the values of "*vertex_pos*" which are "0" and "-1", respectively (**Figure 2-20**).

Vertices — Features Total: 23790, Filtered: 23790, Selected: 0

	PipeID	vertex_pos	vertex_index	vertex_part	vertex_part_index	distance	angle
1	976	0	0	0	0	0	328.5309970189...
2	976	-1	2	0	2	42.74641821646...	312.7040533792...
3	981	0	0	0	0	0	127.6166338235...
4	981	-1	2	0	2	46.92670907235...	111.8145685058...
5	1187	0	0	0	0	0	169.3985624625...
6	1187	-1	1	0	1	25.05764258267...	169.3985624625...
7	1221	0	0	0	0	0	289.8514696440...
8	1221	-1	1	0	1	11.51417563426...	289.8514696440...
9	1222	0	0	0	0	0	228.0464779877...
10	1222	-1	3	0	3	66.6067424128188	214.9459476018...
11	1229	0	0	0	0	0	179.090604055478

Show All Features

Figure 2-20. The start and end vertices of pipe

- *Step 4:* Assign the height of the surface of these vertices from the DEM obtained in the previous section using the function "*Sample raster values*" (**Figure 2-21**).

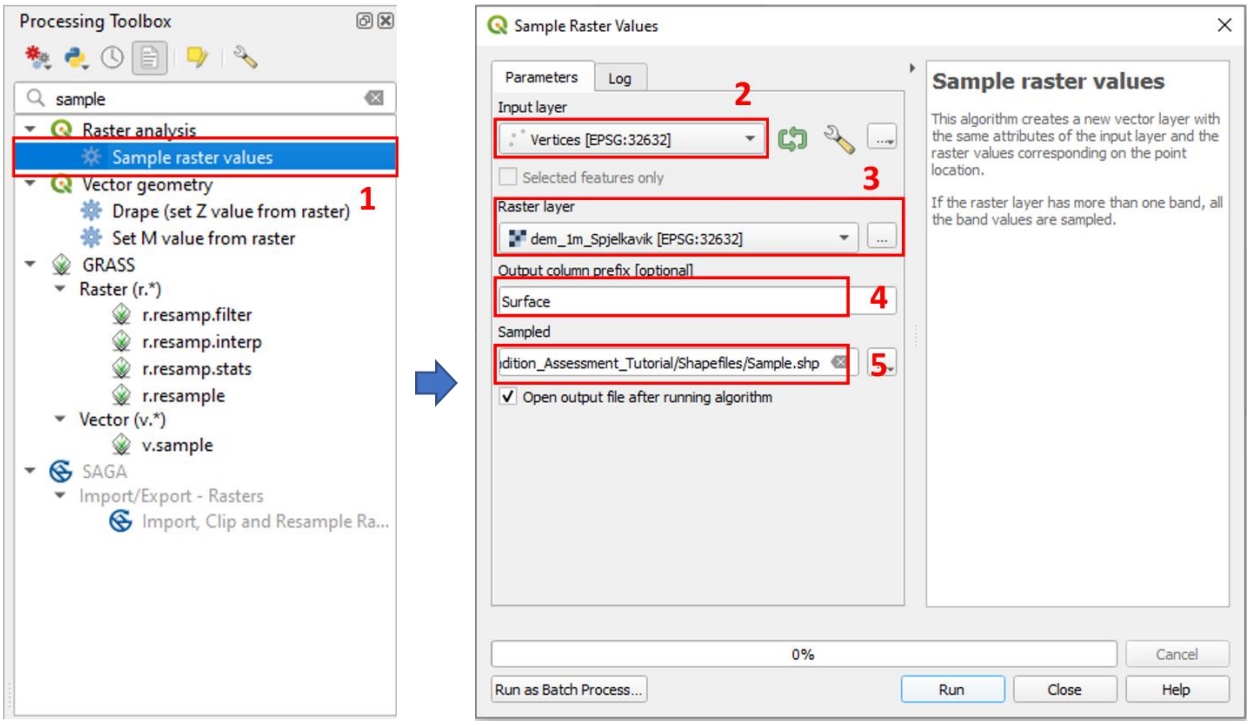


Figure 2-21. Assigning the DEM value for the vertices of the pipe

- *Step 5:* We can easily see that the value in column "*Surface1*" in **Figure 2-22** is the surface height of manholes at the start and end points of a particular pipe. By subtracting the height of the manhole, we can get the inverted elevation of these vertices at the start and end points.

Sample — Features Total: 23790, Filtered: 23790, Selected: 0

123 PipeID = 123 Update All Update Selected

	PipeID	vertex_pos	vertex_ind	vertex_par	vertex_p_1	distance	angle	Surface1
1	976	0	0	0	0	0	328.5309970189...	11.26128482818...
2	976	-1	2	0	2	42.74641821646...	312.7040533792...	4.694990158081...
3	981	0	0	0	0	0	127.6166338235...	7.489274501800...
4	981	-1	2	0	2	46.92670907235...	111.8145685058...	9.785973548889...
5	1187	0	0	0	0	0	169.3985624625...	28.01593971252...
6	1187	-1	1	0	1	25.05764258267...	169.3985624625...	25.86190795898...
7	1221	0	0	0	0	0	289.8514696440...	23.88335990905...
8	1221	-1	1	0	1	11.51417563426...	289.8514696440...	23.85103988647...
9	1222	0	0	0	0	0	228.0464779877...	26.15661811828...
10	1222	-1	3	0	3	66.60674241281...	214.9459476018...	26.32640457153...

Show All Features

Figure 2-22. Assigning the surface height for the vertices of the pipe

- *Step 6:* Average slope of a particular pipe between two manholes is computed as in

Figure 2-23.

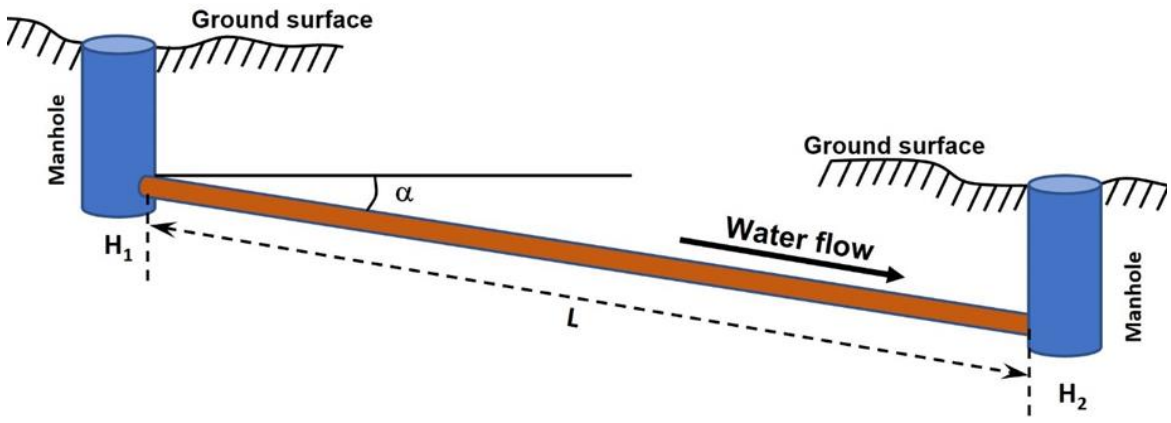


Figure 2-23. A simulation of the sewer's slope

The sewer's slope α (in degree) is computed as followed:

$$\alpha = arctg\left(\frac{H_1 - H_2}{L}\right) \times \frac{180}{\pi} \quad (1)$$

where H_1, H_2, and L are the inverted elevations of the start manhole, end manhole, and sewer's length, respectively. The $arctg(x)$ function is the inverse tangent function of x.

- *Step 7:* Separate the layer in *Step 5* into two separated layers that contained the heights of the start and end points of each pipe. The steps for this implementation are described as follows:
  - *Step 7a:* Select all points that have the value of "*vertex_pos*" equal to "0" (start point) in **Figure 2-20**. The steps for this work are shown in **Figure 2-24**.

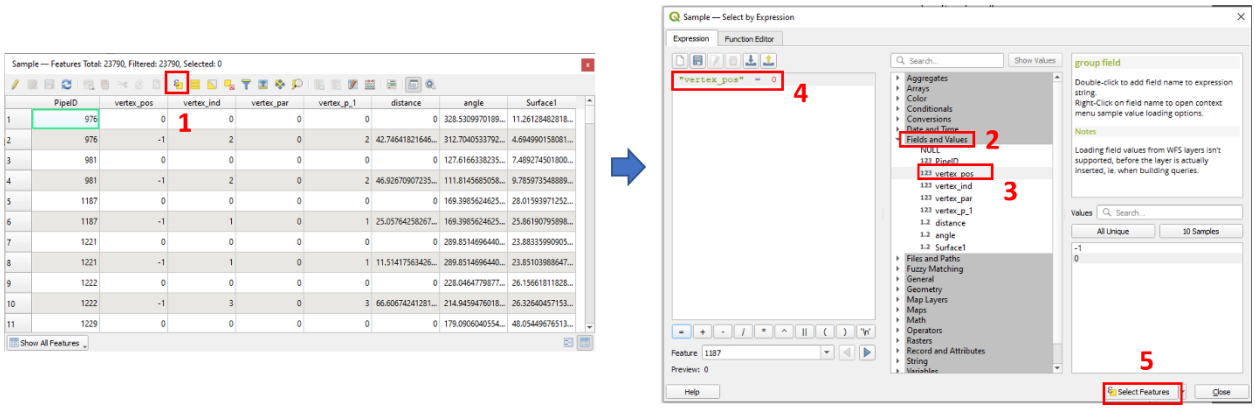


Figure 2-24. Separating layers in QGIS

  - *Step 7b:* Save all selected points into a new layer, named "*Sample_Start*" (**Figure 2-25**).

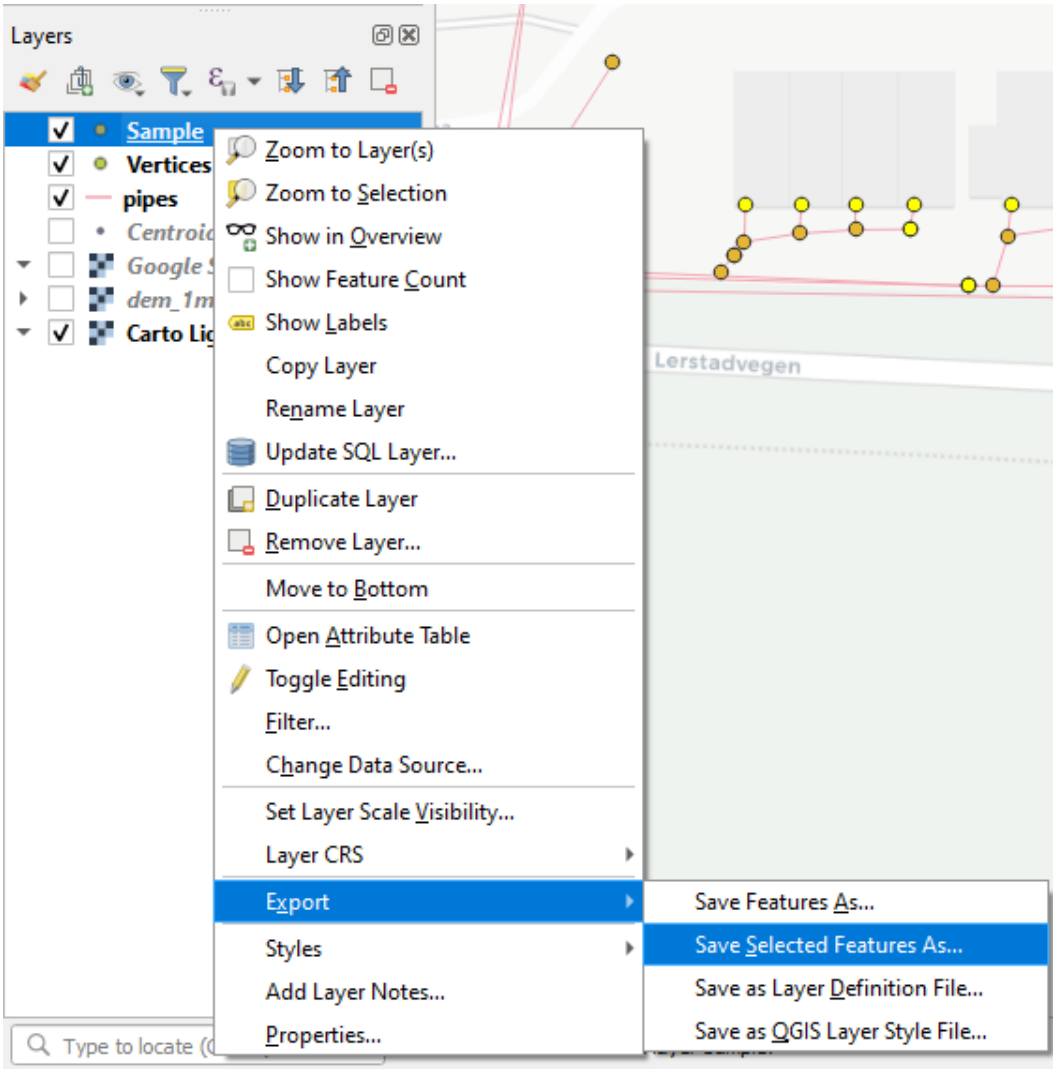
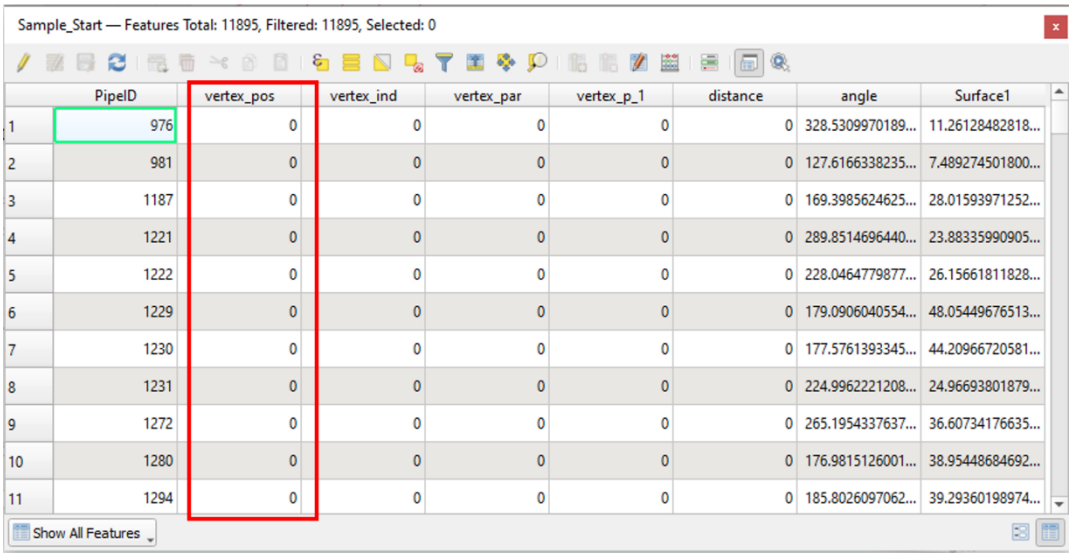


Figure 2-25. Saving the start point layer in QGIS

- *Step 7c:* Check the result (**Figure 2-26**).



	PipeID	vertex_pos	vertex_ind	vertex_par	vertex_p_1	distance	angle	Surface1
1	976	0	0	0	0	0	328.5309970189...	11.26128482818...
2	981	0	0	0	0	0	127.6166338235...	7.489274501800...
3	1187	0	0	0	0	0	169.3985624625...	28.01593971252...
4	1221	0	0	0	0	0	289.8514696440...	23.88335990905...
5	1222	0	0	0	0	0	228.0464779877...	26.15661811828...
6	1229	0	0	0	0	0	179.0906040554...	48.05449676513...
7	1230	0	0	0	0	0	177.5761393345...	44.20966720581...
8	1231	0	0	0	0	0	224.9962221208...	24.96693801879...
9	1272	0	0	0	0	0	265.1954337637...	36.60734176635...
10	1280	0	0	0	0	0	176.9815126001...	38.95448684692...
11	1294	0	0	0	0	0	185.8026097062...	39.29360198974...

Figure 2-26. The start point layer in QGIS

- *Step 7d:* Repeat steps 7a and 7b but replaces the value of "*vertex_pos*" from "0" to "-1" (endpoint) and specify the new layer with the name "*Sample_End*" (**Figure 2-27**).

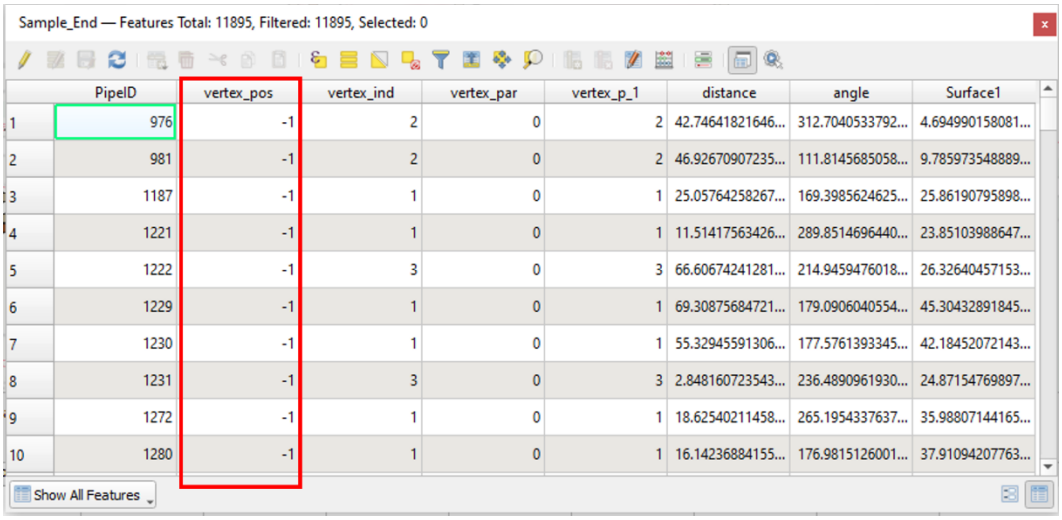


Figure 2-27. The endpoint layer in QGIS

- *Step 8:* Compute the inverted elevation for the start and end points in the layers "*Sample_Start*" and "*Sample_End*".

- *Step 9:* Merge processed elevations in the layers "*Sample_Start*" and "*Sample_End*" into the pipe layer and compute the difference between the start and end points of each pipe. The steps for doing this work are presented as follows:

  - *Step 9a:* Create two new columns in the pipe layer with the names "*Start_Poin*" and "*End_Poin*" to store computed values in *Step 8* (**Figure 2-28**).

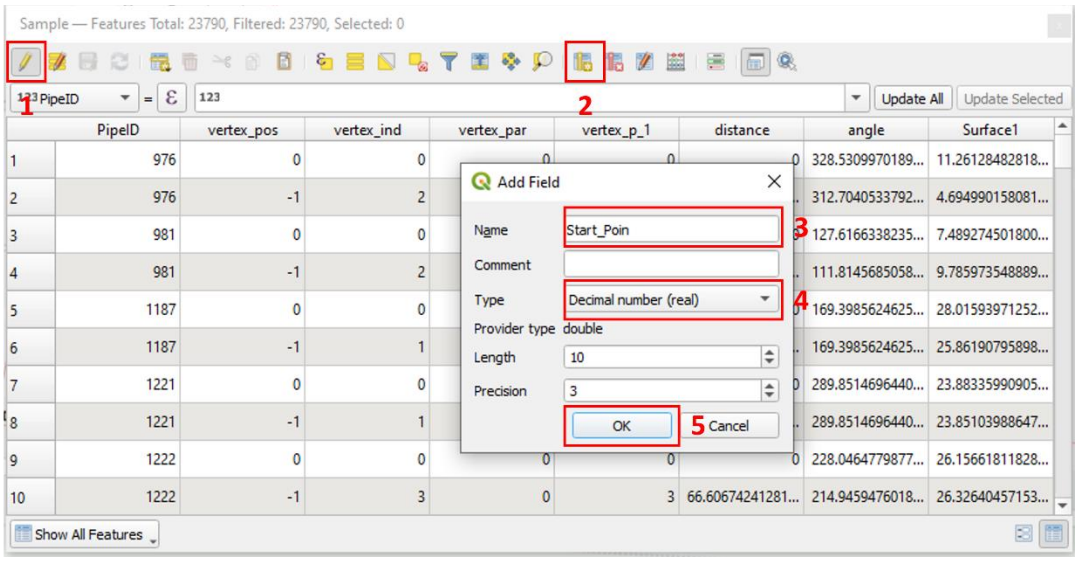


Figure 2-28. Creating a new field in QGIS

- *Step 9b:* Join the layers "*Sample_Start*" and "*Sample_End*" with the pipe layer using a unique ID (**Figure 2-29**).

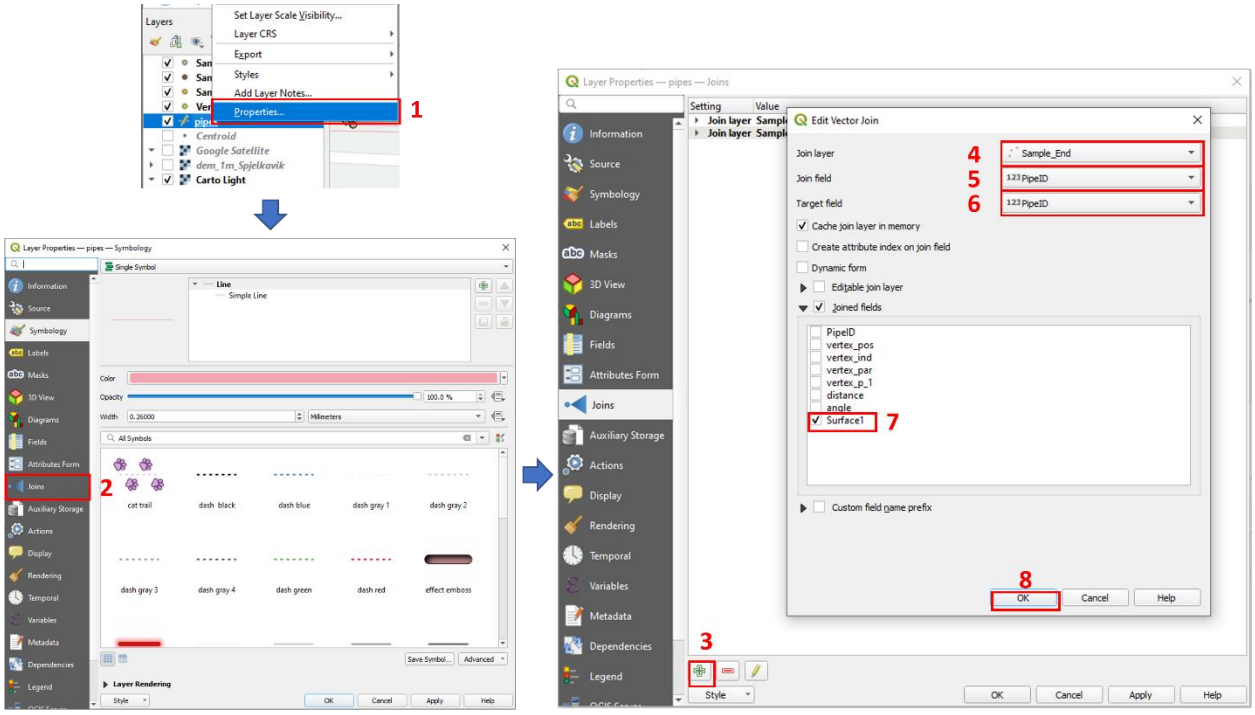
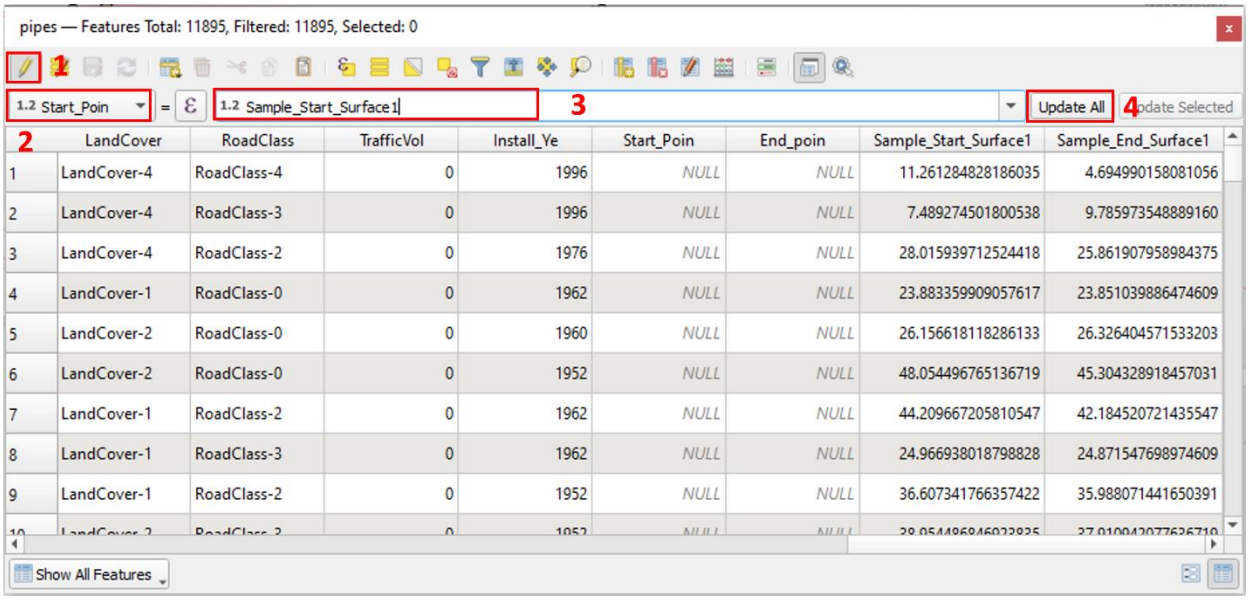


Figure 2-29. Merging layers in QGIS

- *Step 9c:* Assign values for the columns "*Start_Poin*" and "*End_Poin*" by values of the columns "*Surface1*" in the layers "*Sample_Start*" and "*Sample_End*" (**Figure 2-30**).



	LandCover	RoadClass	TrafficVol	Install_Ye	Start_Poin	End_poin	Sample_Start_Surface1	Sample_End_Surface1
1	LandCover-4	RoadClass-4	0	1996	NULL	NULL	11.261284828186035	4.694990158081056
2	LandCover-4	RoadClass-3	0	1996	NULL	NULL	7.489274501800538	9.785973548889160
3	LandCover-4	RoadClass-2	0	1976	NULL	NULL	28.015939712524418	25.861907958984375
4	LandCover-1	RoadClass-0	0	1962	NULL	NULL	23.883359909057617	23.851039886474609
5	LandCover-2	RoadClass-0	0	1960	NULL	NULL	26.156618118286133	26.326404571533203
6	LandCover-2	RoadClass-0	0	1952	NULL	NULL	48.054496765136719	45.304328918457031
7	LandCover-1	RoadClass-2	0	1962	NULL	NULL	44.209667205810547	42.184520721435547
8	LandCover-1	RoadClass-3	0	1962	NULL	NULL	24.966938018798828	24.871547698974609
9	LandCover-1	RoadClass-2	0	1952	NULL	NULL	36.607341766357422	35.988071441650391

Figure 2-30. Assigning values for a column in QGIS

- *Step 9d:* Check the results and disjoin the layers "*Sample_Start*" and

"*Sample_End*" (**Figure 2-31**).

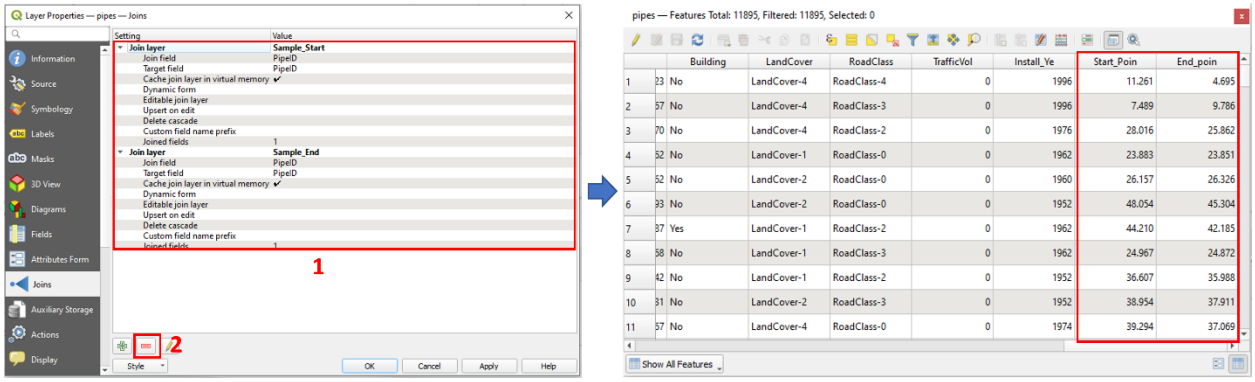


Figure 2-31. Disjoining layers in QGIS

- *Step 9e:* Create a new column, named "*Diff*", and compute the difference (**Figure 2-32**).

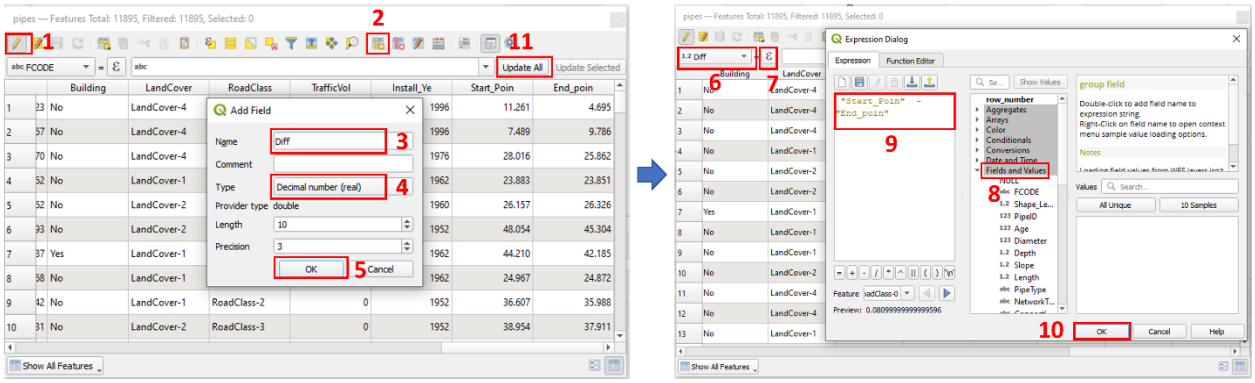


Figure 2-32. Computing the difference between two columns in QGIS

- *Step 9f:* Create a new column, named "*Slope*", and compute the slope based on equation (1) (**Figure 2-33**).

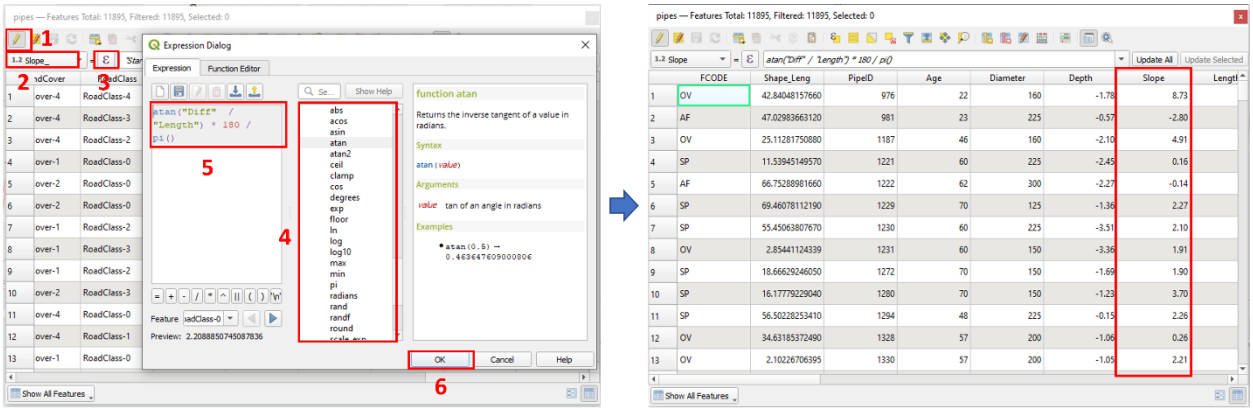


Figure 2-33. Computing the slope of sewer pipes

c. Computing depth

In this tutorial, the depth of pipes was computed as the distance from the ground surface to the centroid of the pipe, therefore, the negative values represent that pipes are lower than the surface.

The steps for computing the centroid of sewer pipes in QGIS are described as follows:

- *Step 1:* Activate the function for computing the centroid from the *"Processing Toolbox"* dialog (steps 1 and 2 in **Figure 2-34**).

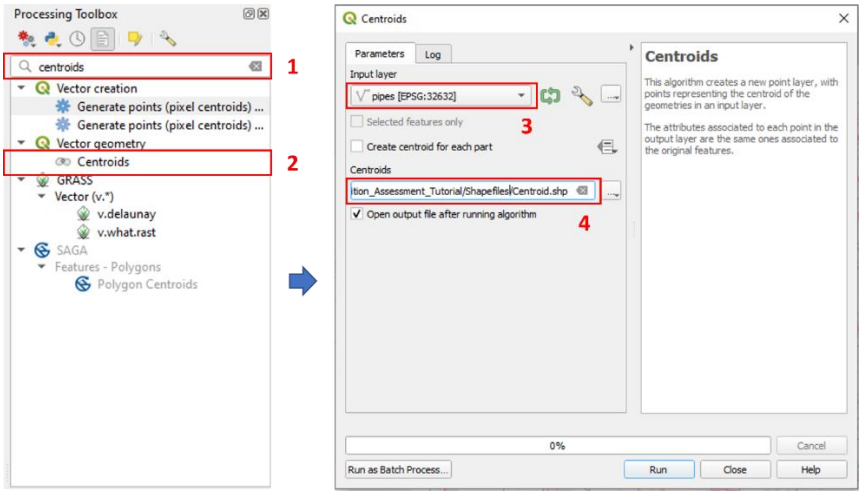


Figure 2-34. Computation of the centroid point of a line

- *Step 2:* Select the layer that contains the sewer pipe (step 3 in **Figure 2-34**) and specify the output destination (step 4 in **Figure 2-34**). Click *"Run"* to execute the function.

- *Step 3:* Check the result (**Figure 2-35**).

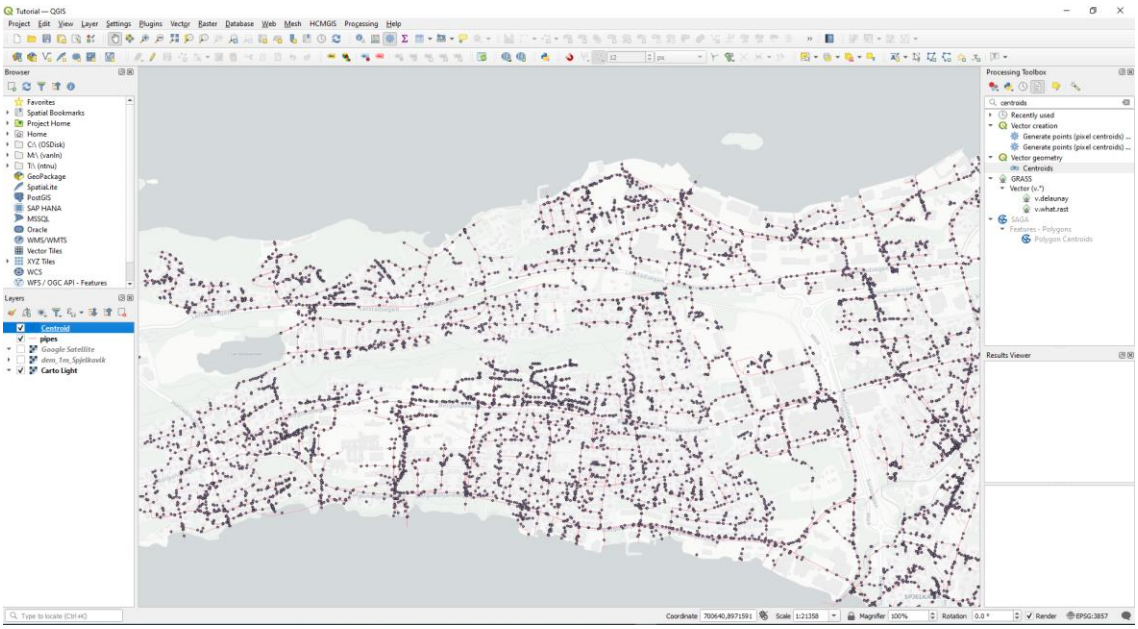


Figure 2-35. Centroid points in QGIS

- *Step 4:* Assign the height of the surface for each centroid point. The steps are described in **Figure 2-21**. A new layer is created, named *"Centroid_Surface"*, that contains the ground surface height of each centroid point (**Figure 2-36**).

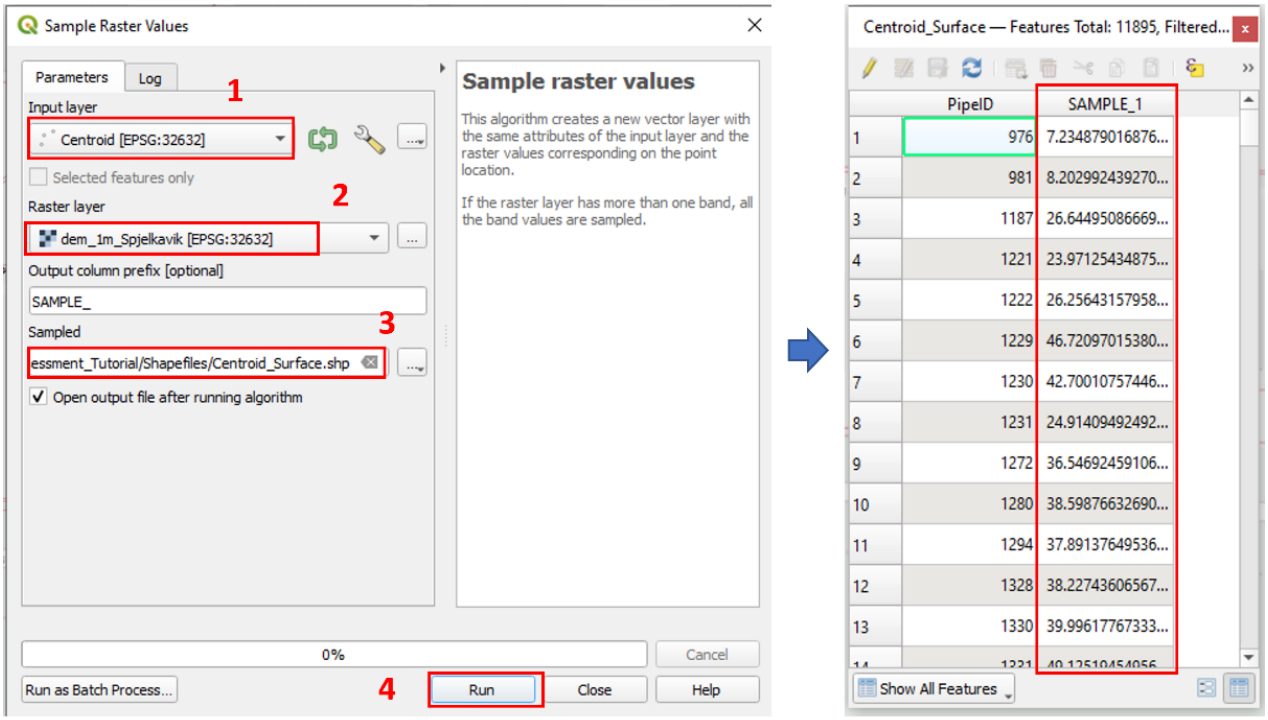


Figure 2-36. Assigning ground surface height for centroid point

- *Step 5:* In this tutorial, the average depth of the centroid point of a particular pipe was computed as the mean value of the depth of the start and end points of the corresponding pipe. Create a new column in the pipe layer named *"Average"* and compute the average value (**Figure 2-37**).

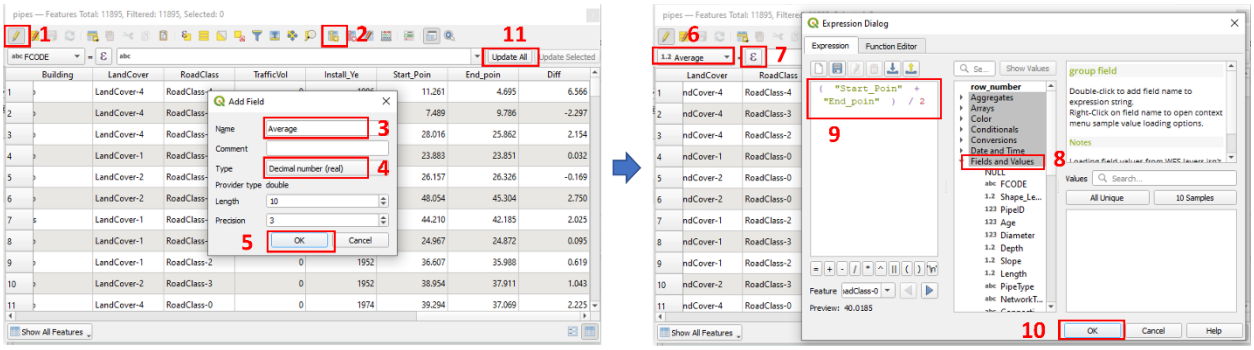


Figure 2-37. Computing the average depth of the pipes from the start and end points

- *Step 6:* Join the column *"SAMPLE_1"* in the layer *"Centroid_Surface"* with the pipe layer based on a unique ID following the steps in **Figure 2-29**. The result is shown in **Figure 2-38**.

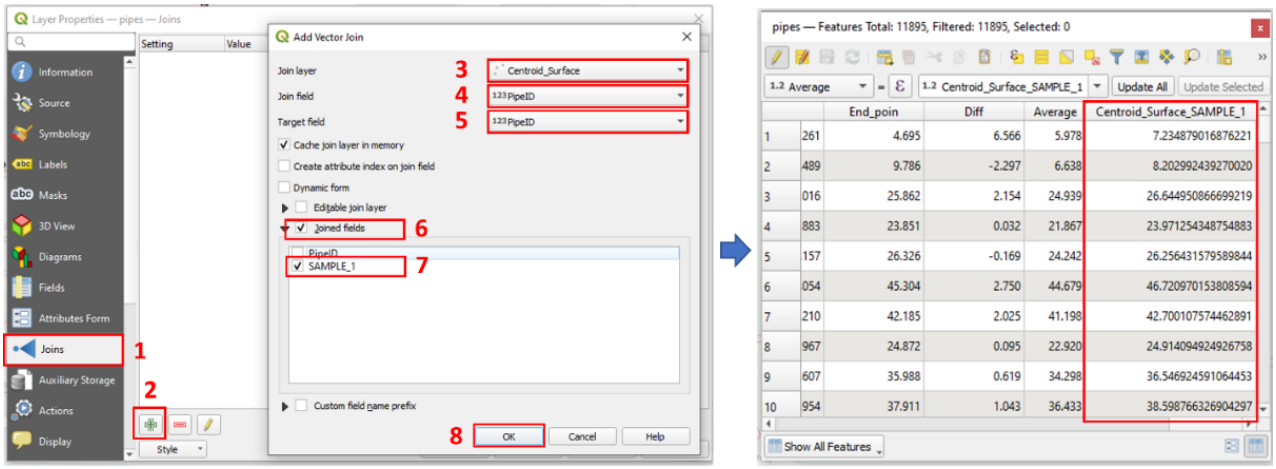


Figure 2-38. Assigning the ground surface height for the pipe layer

➢ *Step 7:* Compute the depth of the pipe using the equation below:

$$D^i = H^i_{avg} - H^i_s \qquad (2)$$

where D^i is the average depth of the i^{th} sewer pipe, H^i_{avg} is the average height computed from the start and end points of the i^{th} sewer pipe, and H^i_s is the height of the ground surface at the i^{th} sewer pipe. Create a new column, named *"Depth"*, in the pipe layer and compute the average depth of the sewer pipe. The steps for the computation of this value are shown in **Figure 2-39**.

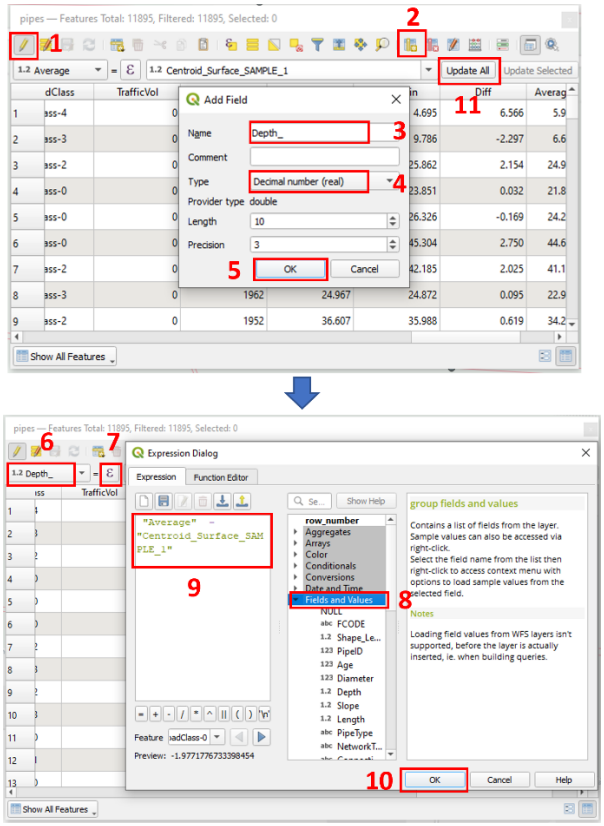
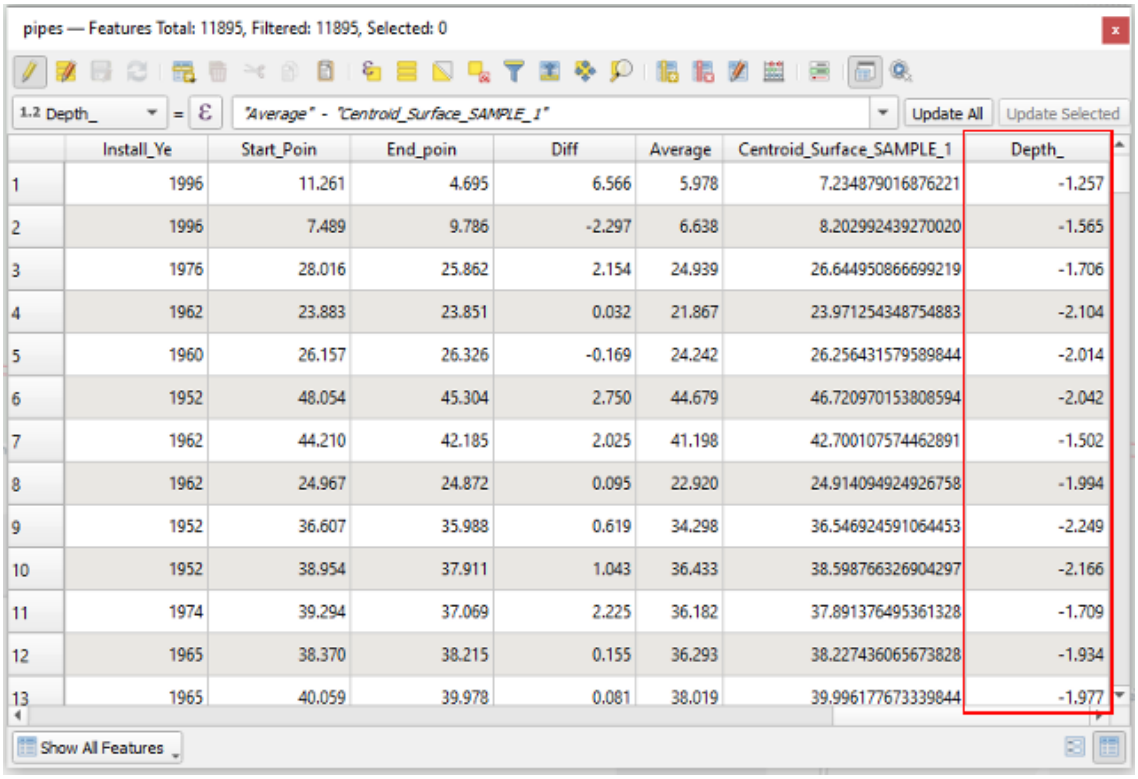


Figure 2-39. Computing the average depth of the pipes

➢ *Step 8:* Check the results (**Figure 2-40**).



	Install_Ye	Start_Poin	End_poin	Diff	Average	Centroid_Surface_SAMPLE_1	Depth_
1	1996	11.261	4.695	6.566	5.978	7.234879016876221	-1.257
2	1996	7.489	9.786	-2.297	6.638	8.202992439270020	-1.565
3	1976	28.016	25.862	2.154	24.939	26.644950866699219	-1.706
4	1962	23.883	23.851	0.032	21.867	23.971254348754883	-2.104
5	1960	26.157	26.326	-0.169	24.242	26.256431579589844	-2.014
6	1952	48.054	45.304	2.750	44.679	46.720970153808594	-2.042
7	1962	44.210	42.185	2.025	41.198	42.700107574462891	-1.502
8	1962	24.967	24.872	0.095	22.920	24.914094924926758	-1.994
9	1952	36.607	35.988	0.619	34.298	36.546924591064453	-2.249
10	1952	38.954	37.911	1.043	36.433	38.598766326904297	-2.166
11	1974	39.294	37.069	2.225	36.182	37.891376495361328	-1.709
12	1965	38.370	38.215	0.155	36.293	38.227436065673828	-1.934
13	1965	40.059	39.978	0.081	38.019	39.996177673339844	-1.977

Figure 2-40. The average depth of the pipes

2.3. Environmental Factors Computation

In this tutorial, environmental factors include, but not limited to, rainfall, geology, population, land use, building area, groundwater, traffic volume, distance to road, and soil type. Other environmental factors are not considered in this tutorial because of unavailable data in the study area. The influence of these factors on the sewer deterioration process was described in the study of Nguyen et al. (2022). The sources of these environmental factors are provided in **Table 2-1**.

Table 2-1. Environmental factors in this analysis

Environmental Factor	Spatial Resolution	GIS Type	Source	Assess Link
Rainfall	-	Point	NCSC	https://klimaservicesenter.no
Geology	1:50,000	Polygon	NMA	https://www.kartverket.no
Population	250 m × 250 m	Grid	NMA	https://www.kartverket.no
Land use	1:5000	Grid	NMA	https://www.kartverket.no
Building area	1:5000	Polygon	NMA	https://www.kartverket.no
Groundwater	-	Point	NGS	https://www.ngu.no
Traffic volume	5 m × 5 m	Grid	NPRA	https://www.vegvesen.no/en
Distance to road	5 m × 5 m	Grid	NMA	https://www.kartverket.no
Soil type	1:50,000	Point	NCSC	https://www.kartverket.no

Abbreviations: NCSC - Norwegian Climate Service Center; NMA - Norwegian Mapping Authority; COAH - Copernicus Open Access Hub; NGS - Norwegian Geological Survey; NPRA - Norwegian Public Roads Administration.

a. Rainfall

Rainfall results in rising groundwater which leads sewer pipes to deteriorate more quickly (Kwak et al., 2020). Rainfall data were obtained from annual average rainfall over several years at nine weather stations near the study area (**Figure 2-41**).

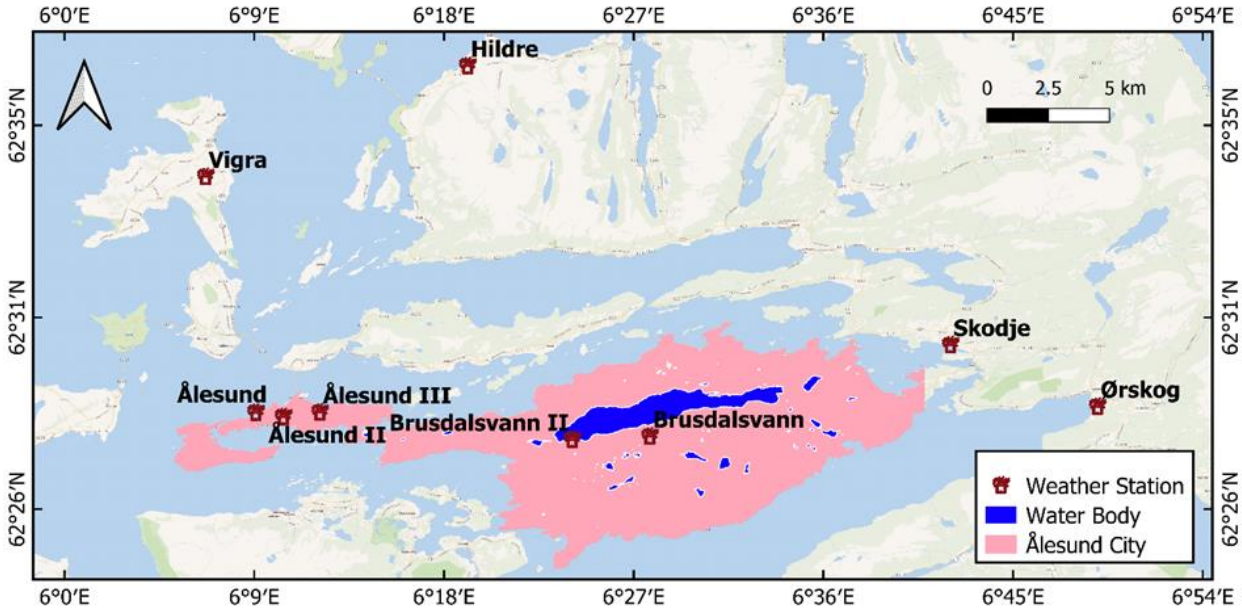


Figure 2-41. Location of the weather stations around the study area

The weather station data provided by the Norwegian Climate Service Center (https://klimaservicesenter.no/) is provided in **Table 2-2**.

Table 2-2. The hydrological stations used for rainfall interpolation

Weather station name	Latitude (°)	Longitude (°)	Avg. rainfall (mm)	Period
Brusdalsvann	62.4666	6.4626	157.0	01.1907 - 12.1972
Brusdalsvann II	62.4654	6.4013	152.1	01.1973 - 12.2014
Skodje	62.5000	6.7004	139.8	01.1961 - 12.1979
Ålesund	62.4753	6.1511	105.8	01.1895 - 12.1930
Ålesund II	62.4737	6.1729	95.5	01.1908 - 12.1954
Ålesund III	62.4754	6.2017	125.9	01.1955 - 12.2004
Ørskog	62.4775	6.8167	130.7	01.1896 - 12.2019
Hildre	62.6016	6.3186	125.5	01.1970 - 12.2018
Vigra	62.5613	6.1113	113.7	01.1959 - 12.2019

The rainfall of each weather station at a specific period can be calculated using the interpolation approach (**Figure 2-42**).

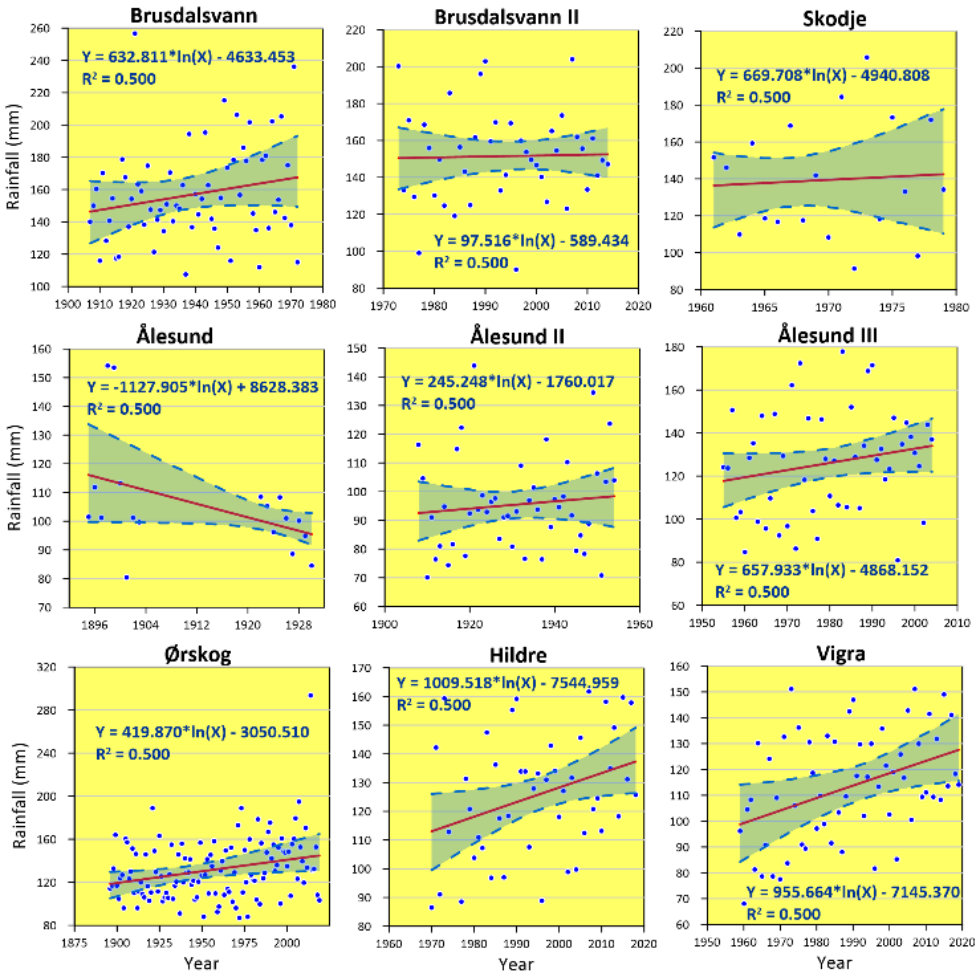


Figure 2-42. Fitting functions for rainfall interpolation

After the rainfall at a specific year is calculated, a rainfall map can be established using the Inverse Distance Weighting (IDW) method. The steps for creating a rainfall map are presented as follows:

- *Step 1:* Create a CSV file containing the rainfall at the weather stations in the wanted period. An example of the rainfall interpolated for the year 2022 is presented in **Figure 2-43**.

	A	B	C	D
1	**Station Name**	**Latitude**	**Longitude**	**Rainfall (2022)**
2	Brusdalsvann	62.4666	6.4626	183.4
3	Brusdalsvann II	62.4654	6.4013	152.8
4	Skodje	62.5	6.7004	156.9
5	Ålesund	62.4753	6.1511	42.9
6	Ålesund II	62.4737	6.1729	106.8
7	Ålesund III	62.4754	6.2017	139.9
8	Ørskog	62.4775	6.8167	145.5
9	Hildre	62.6016	6.3186	139.3
10	Vigra	62.5613	6.1113	129

Figure 2-43. Rainfall data at the weather stations in 2022

➢ *Step 2:* Import the CSV containing the geographic location of the weather stations in QGIS (**Figure 2-44**).

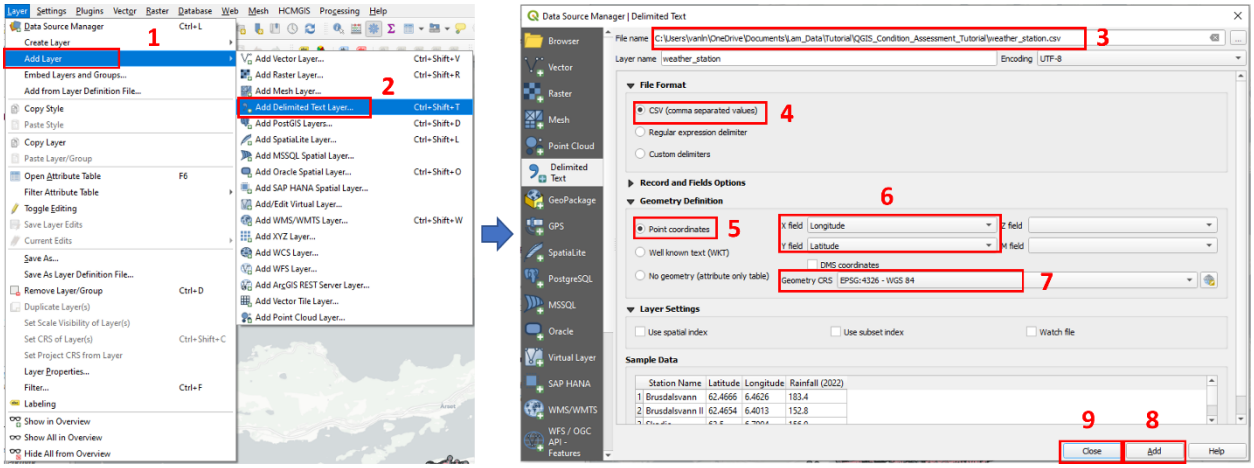


Figure 2-44. Adding a CSV file to QGIS (with geometry)

➢ *Step 3:* Save as a new vector layer, named *"Weather_Station"* in the same coordinate system with the pipe layer (WGS84-UTM32T (EPSG:32632)).

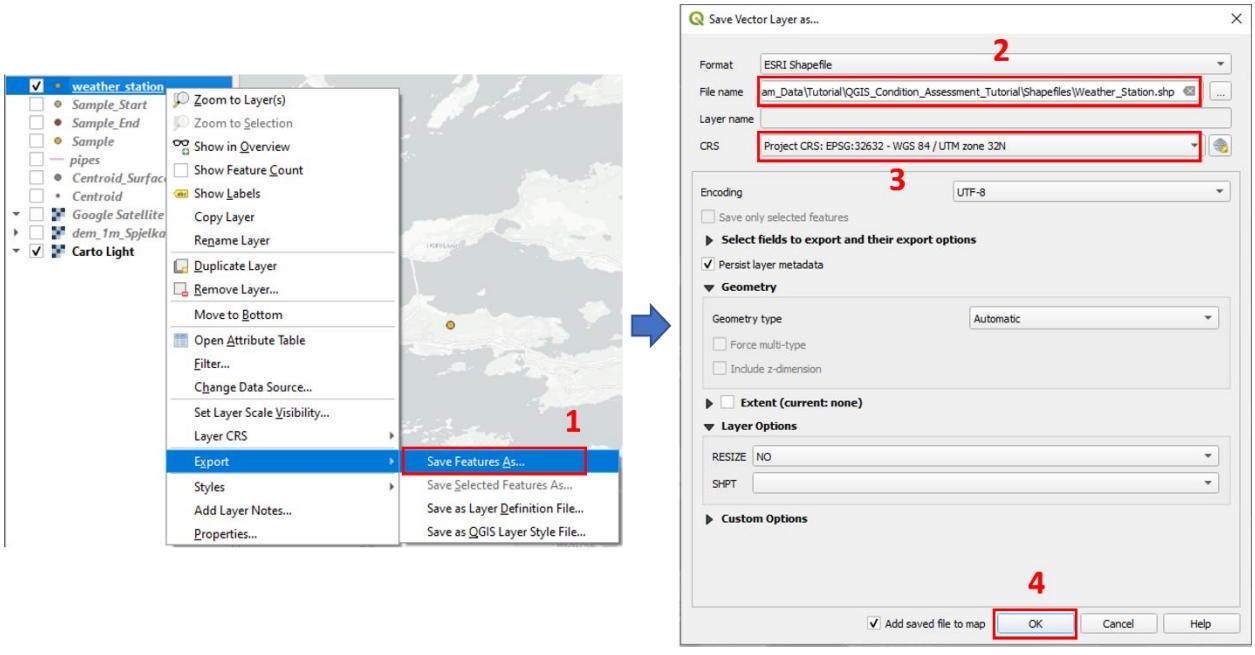


Figure 2-45. Saving a layer in QGIS

➢ *Step 4:* Interpolate rainfall for the entire Ålesund city using the IDW method (**Figure 2-46**). In step 5 in **Figure 2-46**, the extent area should be focused on the interesting study area. Interpolating for a huge area while the study area is quite small will waste the time and computational resources. In addition, by setting up the CRS as WGS84-UTM32T (EPSG:32632), the unit of pixel size in step 6 in **Figure 2-46** will be the

meter.

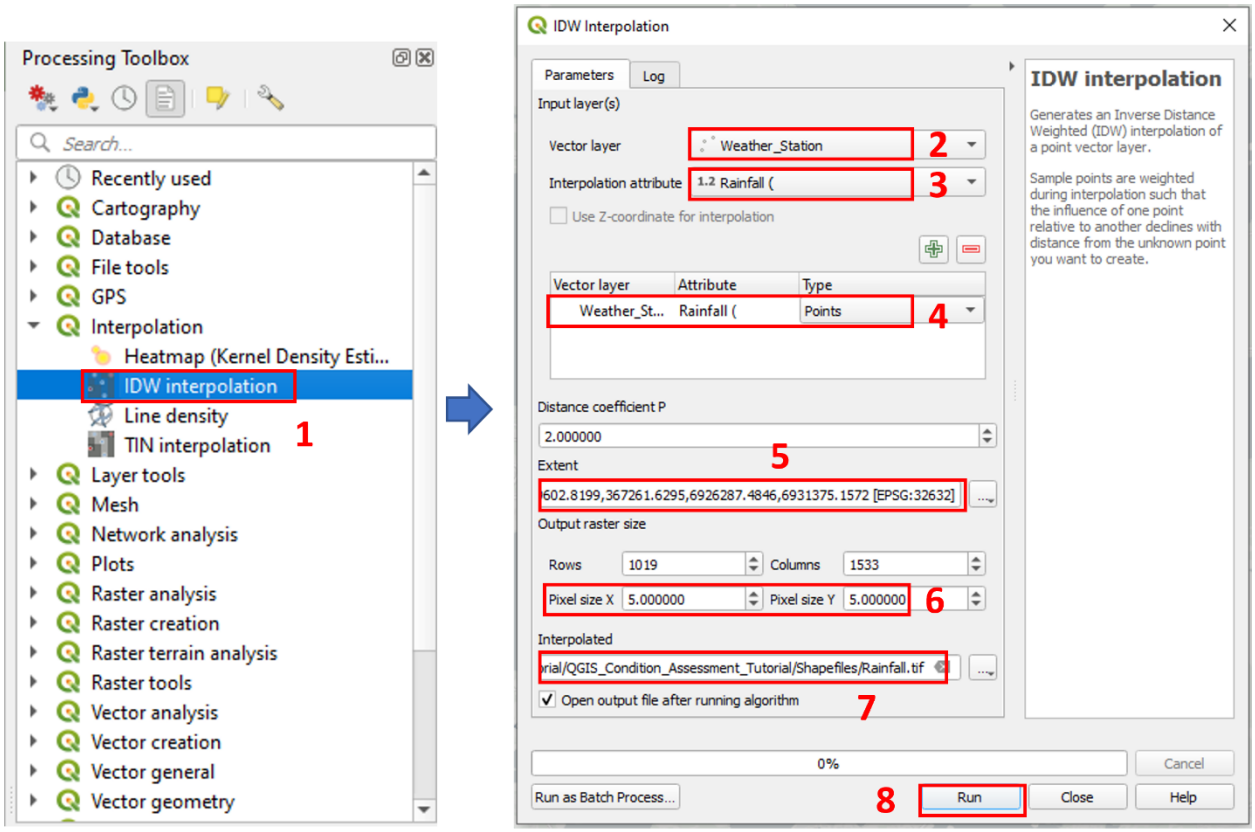


Figure 2-46. Rainfall interpolation

➢ *Step 5:* Check the result (**Figure 2-47**).

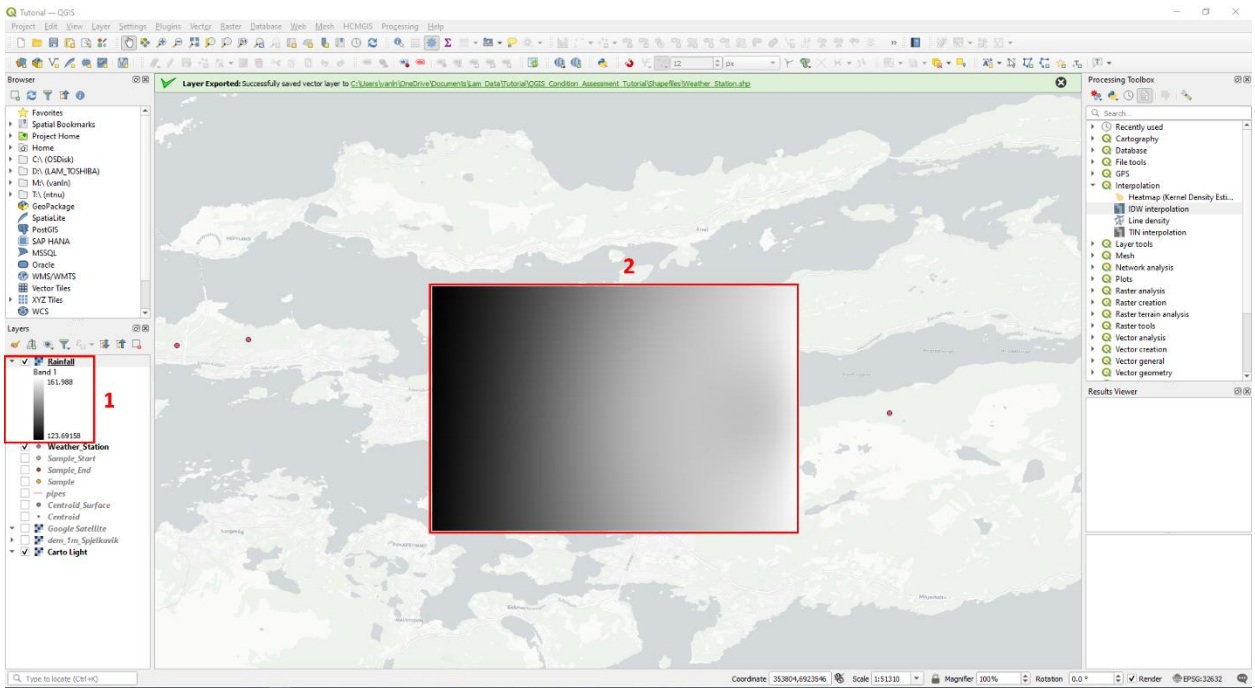


Figure 2-47. Map of interpolated rainfall

➢ *Step 6:* Assign the value of rainfall created from the previous step based on steps in **Figure 2-36** to the centroid layer. Save a new layer with the name *"Centroid_Rainfall"*. The result is shown in **Figure 2-48**.

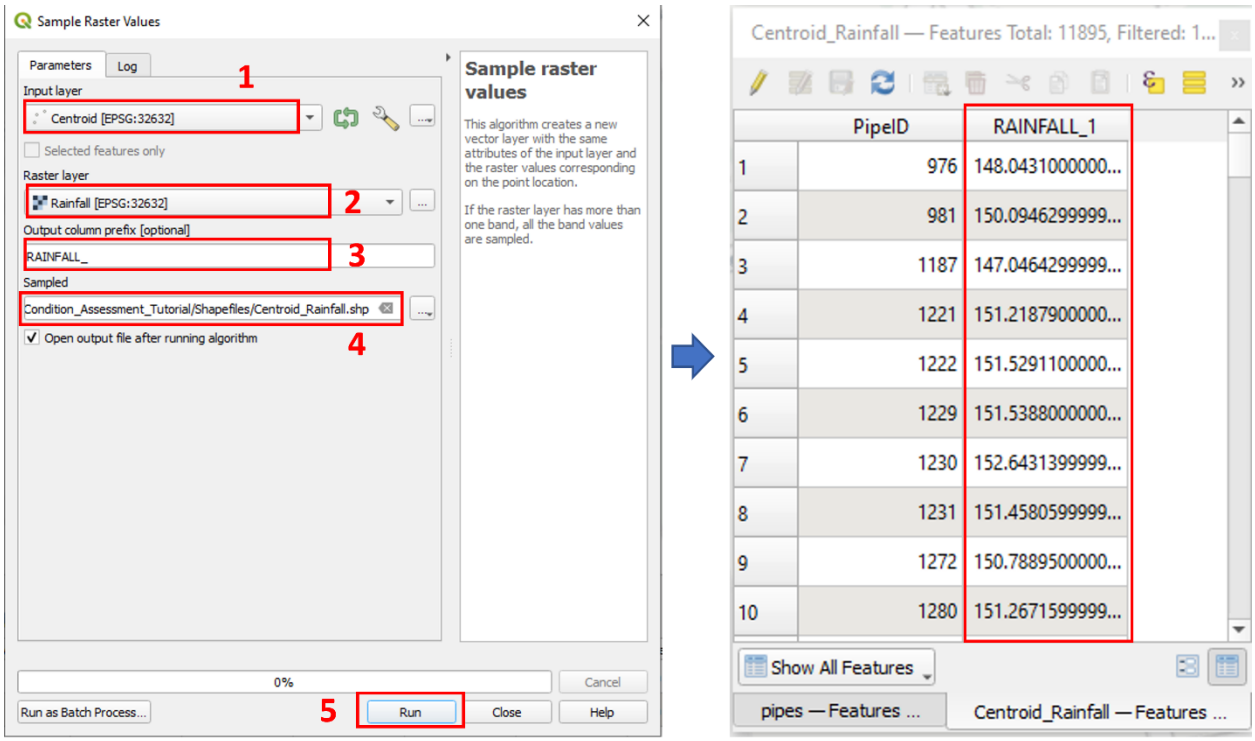


Figure 2-48. Assigning rainfall value for centroid point

➢ *Step 7:* Join the column *"RAINFALL_1"* in the layer *"Centroid_Rainfall"* with the pipe layer based on a unique ID following the steps in **Figure 2-29**. The result is shown in **Figure 2-49**.

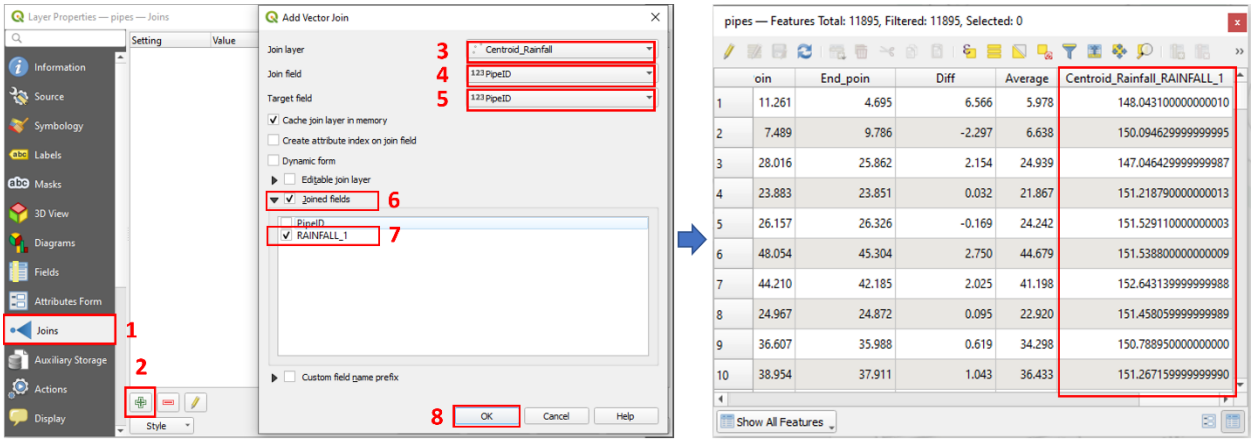


Figure 2-49. Assigning the rainfall value for the pipe layer

➢ *Step 8:* Create a new column, named *"Rainfall"* in the pipe layer and assign the value from the above step to this column (**Figure 2-50**).

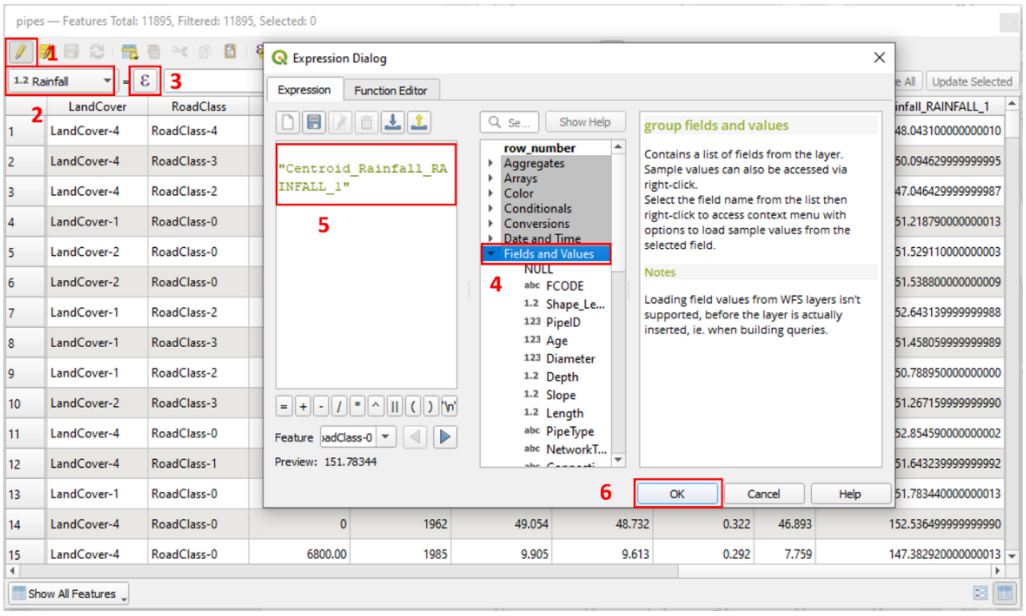


Figure 2-50. Assigning value for the column

b. Geology

The geological characteristics around a sewer pipe can affect its condition processes. For instance, it has been shown that changes in geological structures affect infiltration and groundwater in coastal urban areas, resulting in sewer deterioration (Su et al., 2020). Additionally, hydraulic conductivity in different geological types can affect sewer deterioration differently (Liu et al., 2018).

The geological map was provided as polygons in which each polygon represents each type of geological characteristic. A clipped geological map for the study area is shown in **Figure 2-51**.

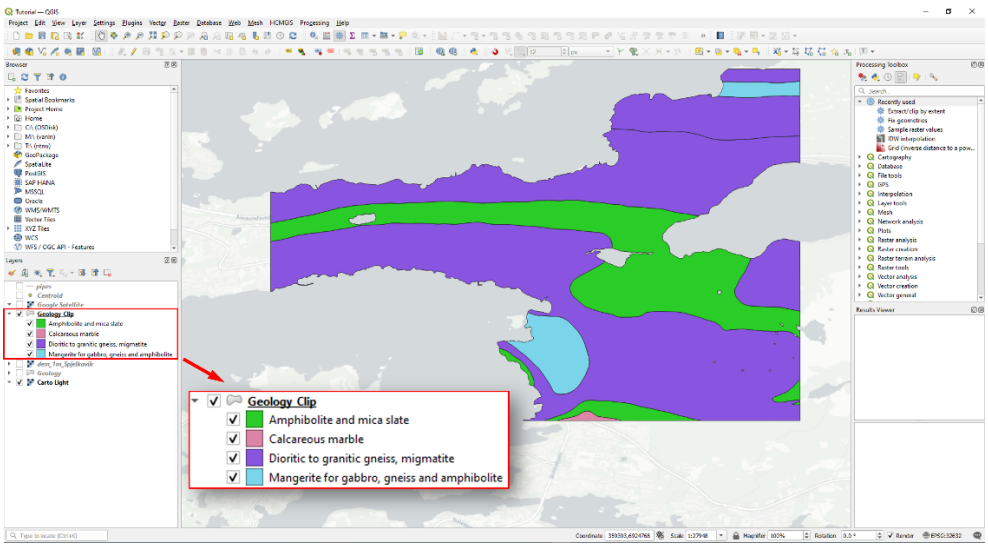


Figure 2-51. Geological map in the study area

The steps for assigning geological characteristics are described as follows:

- *Step 1:* Convert the geology from the vector type (polygon) to the raster type based on the corresponding characteristics of each type (**Figure 2-52**). Because rasterizing a vector to raster requires the rasterized field is numeric. Therefore, the type of geology must be coded into integers based on its type. A new column, named *"Geology_Co"*, was created to store coded values (step 3 in **Figure 2-52**). The output resolution at step 4 in **Figure 2-52** is similar to the output resolution of the rainfall map at step 6 in **Figure 2-46**.

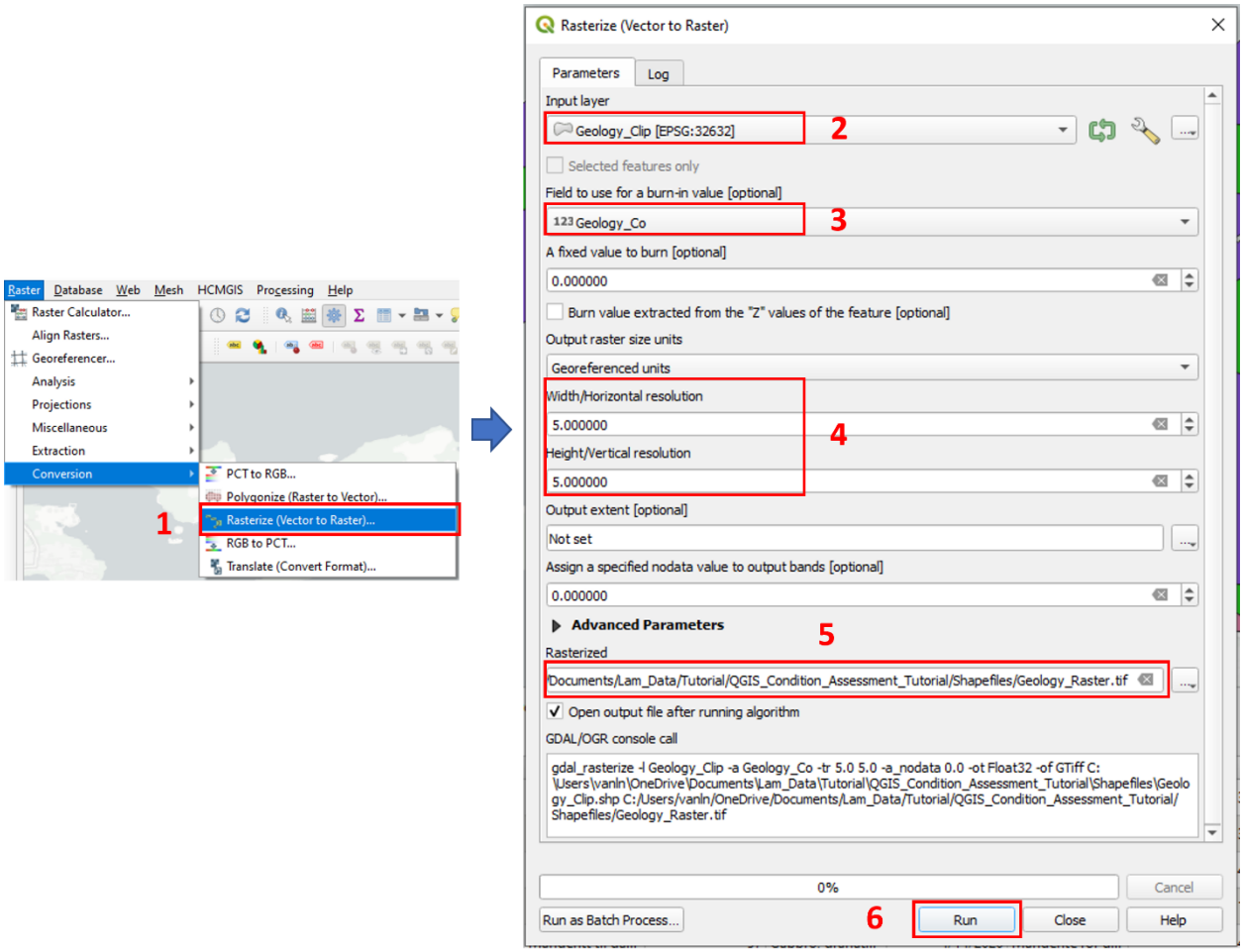


Figure 2-52. Converting vector to raster in QGIS

- *Step 2:* Check the geological raster map (**Figure 2-53**).

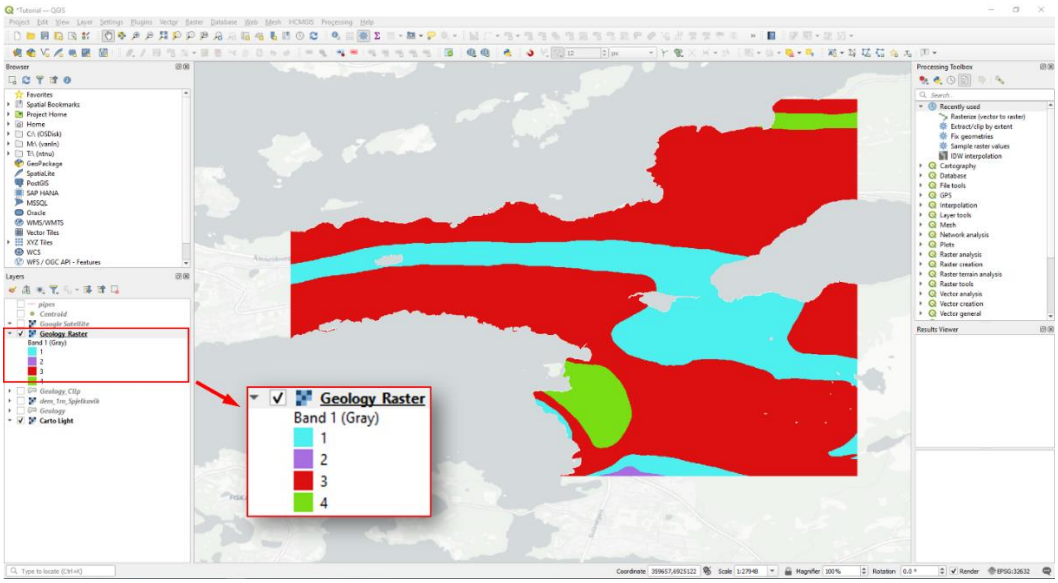


Figure 2-53. Geological raster map

- *Step 3:* Assign the values of the geological raster map for the centroid point of the pipe layer. The implementation steps are similar to assigning rainfall data (the steps from **Figure 2-48** to **Figure 2-50**). The result is shown in **Figure 2-54**.

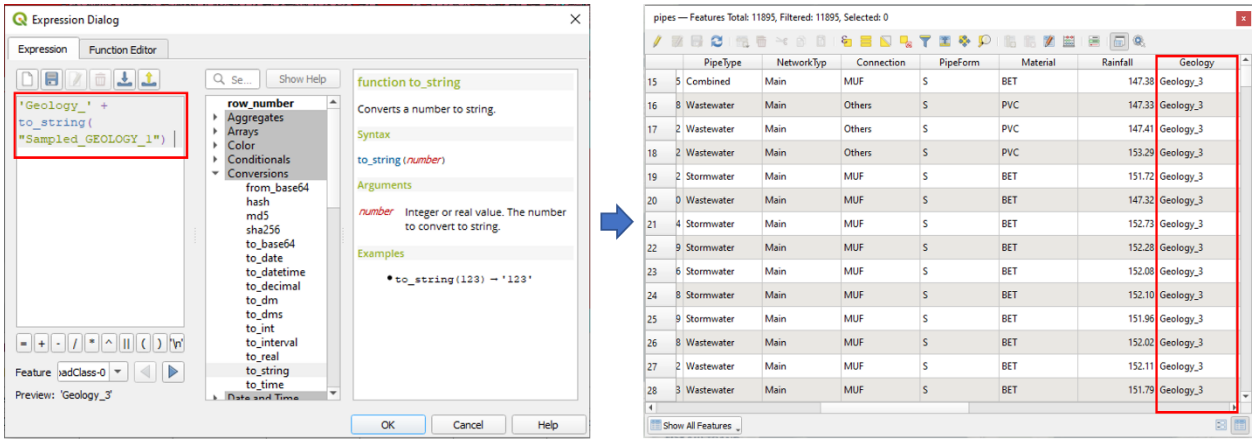


Figure 2-54. Assigning the geological value for the pipe layer

c. Population

Population density is considered a critical factor for sewer deterioration. For example, a large population may lead to a huge volume of wastewater discharge into the wastewater collection network, resulting in the deterioration of the system (Zamanian et al., 2020).

The population map provided by the NMA was already in raster type, therefore rasterizing is not necessary. The population-related data from NMA at the time of this tutorial only provides the map of the population in 2018. Additionally, the population is a dynamic component depending on time and the population in a specific year can be calculated using the interpolation

method. According to Worldometer (2022), the annual population change (APC) in Ålesund city is 0.62% (2020-2021). To calculate the population in the specific year t, we assumed that the population change is directly proportional to the APC. Then, the value of pixel i in population density maps in the years t were calculated as follows:

$$P_t^i = P_{2018}^i[1 + 0.62\% \times (t - 2018)] \quad (3)$$

where P_{2018}^i and P_t^i are the population density in the year 2018 and at the calculated time t.

The steps for assigning the population density to the sewer pipe are described as follows:

- *Step 1:* Import the population raster map of the entire Ålesund city in QGIS (**Figure 2-55**).

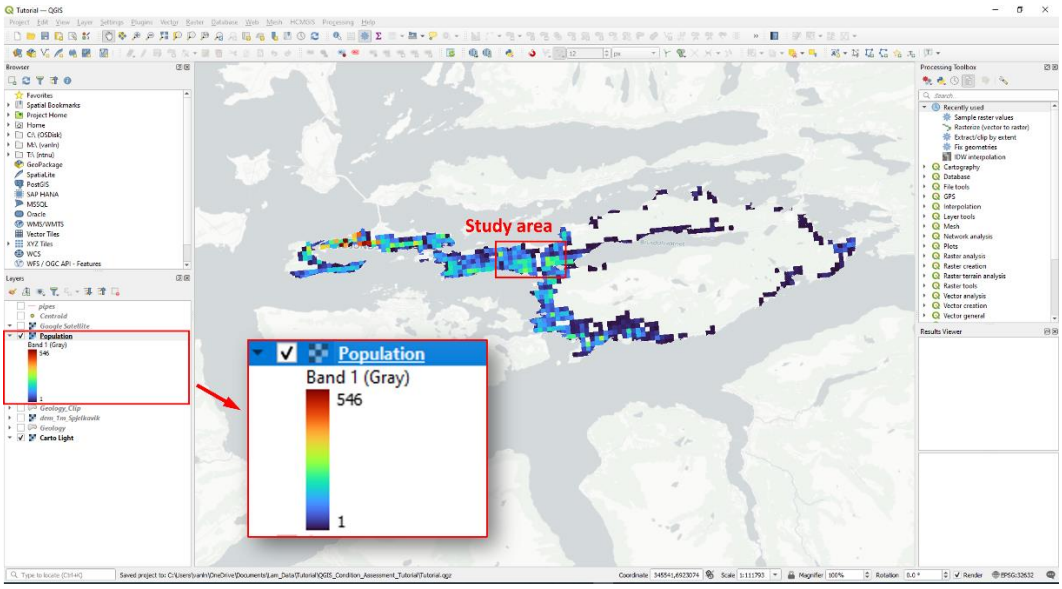


Figure 2-55. Importing the population map

- *Step 2:* Clip the population raster map from the entire Ålesund area to the study area (**Figure 2-56**).

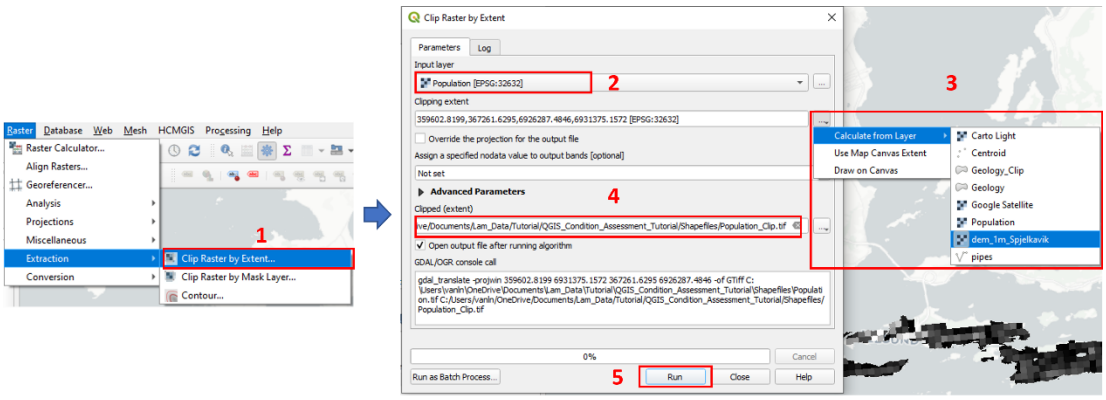


Figure 2-56. Clipping the population map depending on the study area

➢ *Step 3:* Check the result (**Figure 2-57**).

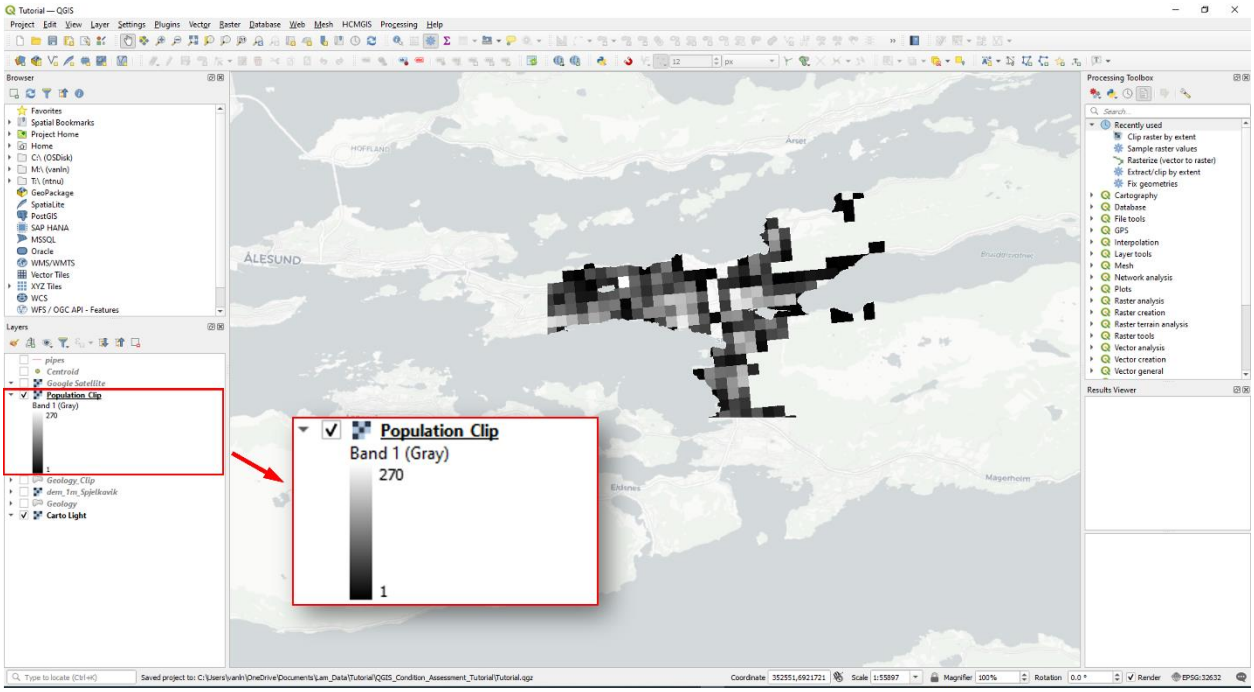


Figure 2-57. The population map in the study area

➢ *Step 4:* Calculate the population density for a specific year (2022) from the data in the year 2018 using the raster calculation tool in QGIS based on equation (3) (**Figure 2-58**).

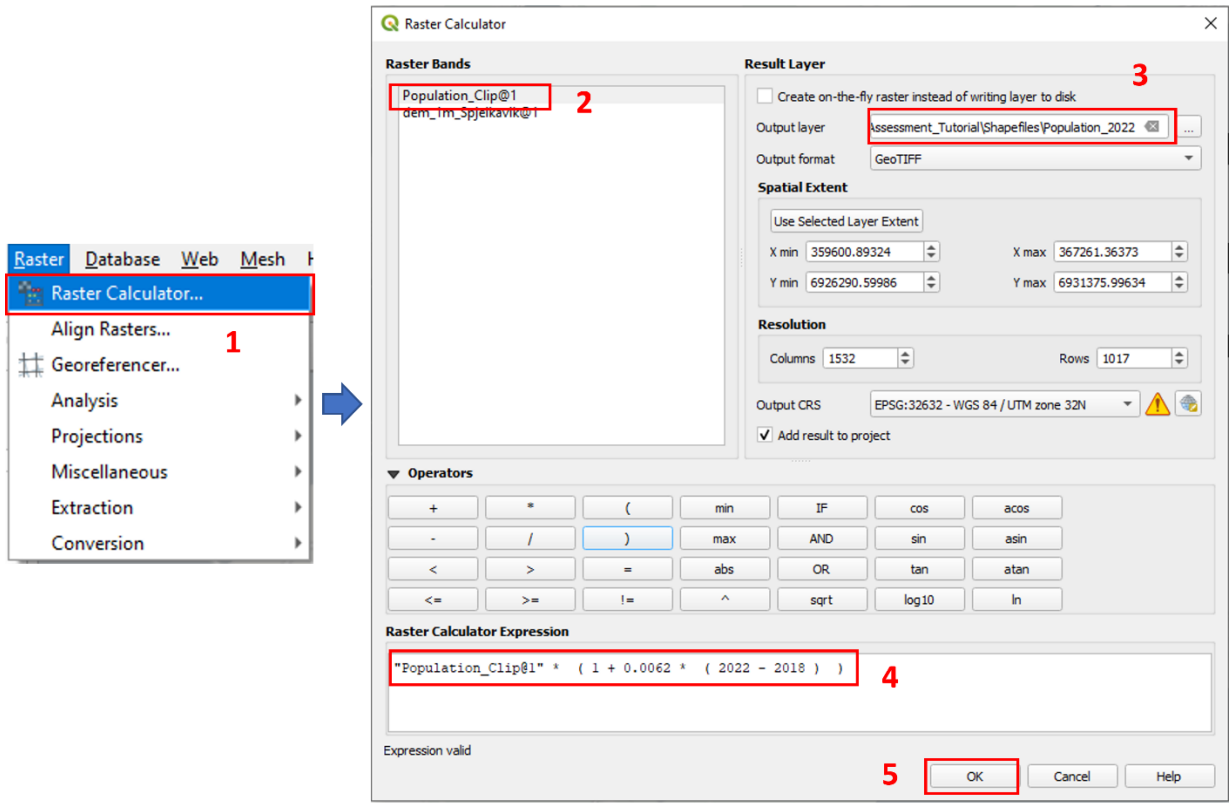


Figure 2-58. Calculating the population using the GIS tool

➢ *Step 5:* Check the processed population density map (**Figure 2-59**).

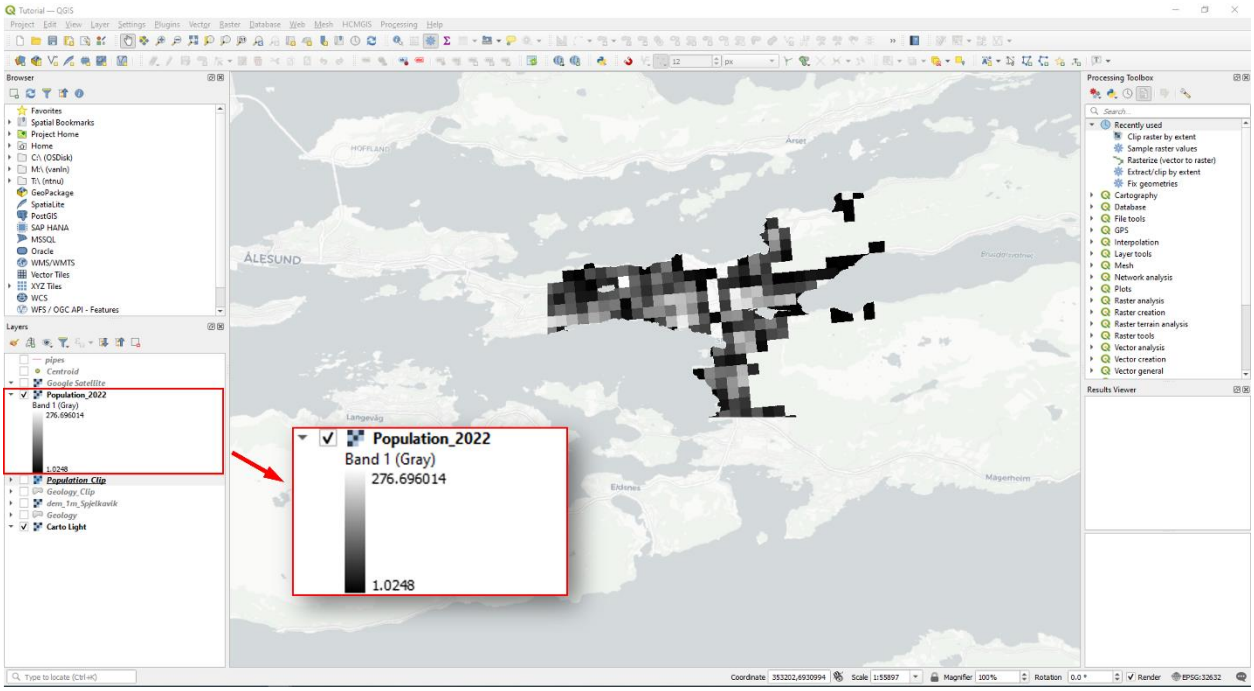


Figure 2-59. Processed population map

➢ *Step 6:* Assign the values of the population raster map for the centroid point of the pipe layer (the steps from **Figure 2-48** to **Figure 2-50**). The result is shown in **Figure 2-60**.

pipes — Features Total: 11895, Filtered: 11895, Selected: 0

	Install_Ye	Start_Poin	End_poin	Diff	Average	Population
1	1996	11.261	4.695	6.566	5.978	23.57
2	1996	7.489	9.786	-2.297	6.638	68.66
3	1976	28.016	25.862	2.154	24.939	71.74
4	1962	23.883	23.851	0.032	21.867	63.54
5	1960	26.157	26.326	-0.169	24.242	63.54
6	1952	48.054	45.304	2.750	44.679	197.79
7	1962	44.210	42.185	2.025	41.198	89.16
8	1962	24.967	24.872	0.095	22.920	69.69
9	1952	36.607	35.988	0.619	34.298	248.00
10	1952	38.954	37.911	1.043	36.433	236.73
11	1974	39.294	37.069	2.225	36.182	171.14
12	1965	38.370	38.215	0.155	36.293	236.73
13	1965	40.059	39.978	0.081	38.019	236.73

Show All Features

Figure 2-60. Assigning the population density for the pipe layer

d. Groundwater

Groundwater is considered an essential factor that influences sewer pipes (Su et al., 2020),

because groundwater at or above sewer pipes leads to water infiltration into the pipe, facilitating the deterioration processes. In addition, the availability of groundwater around the sewer pipe can destabilize the soil around the sewers leading to failures or collapses.

In this tutorial, the groundwater map was prepared using the IDW method and 31 drills data around the study area. The locations of these drills are shown in **Figure 2-61**.

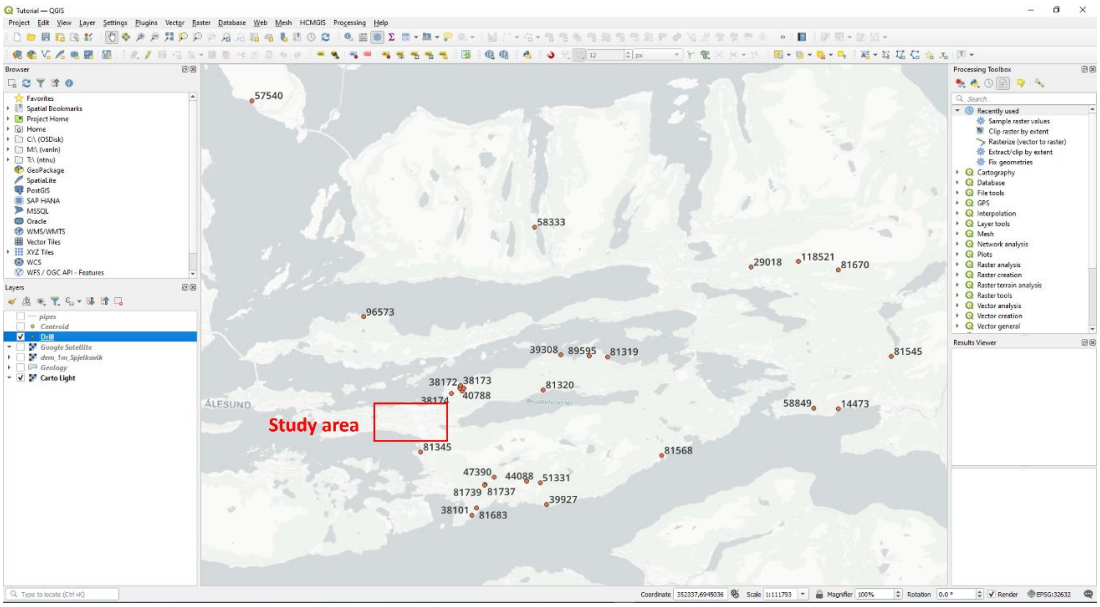


Figure 2-61. Location of the drills near the study area

The steps for calculating the groundwater map are described as follows.

- *Step 1:* Create an interpolated map for groundwater from the drills. The process was similarly implemented in creating a map for rainfall (**Figure 2-62**).

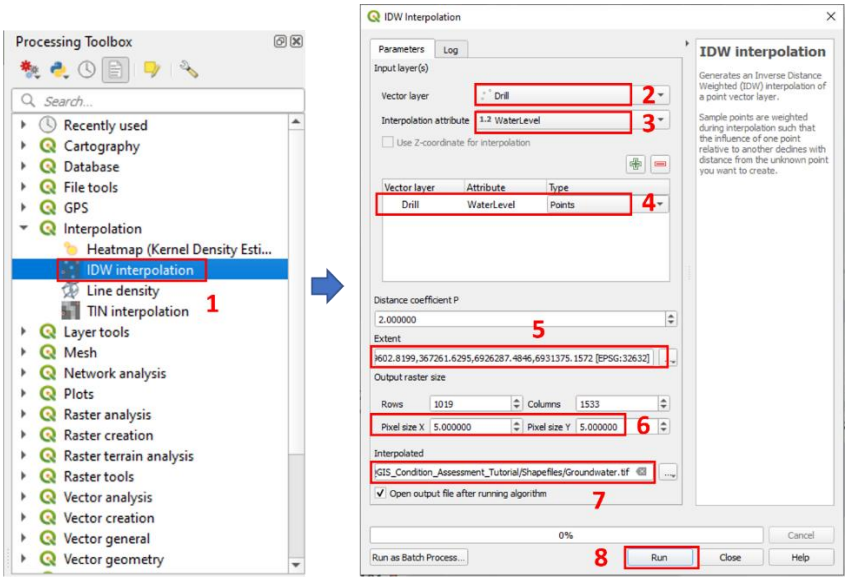


Figure 2-62. Calculating groundwater from rainfall

➢ *Step 2:* Check the result (**Figure 2-63**).

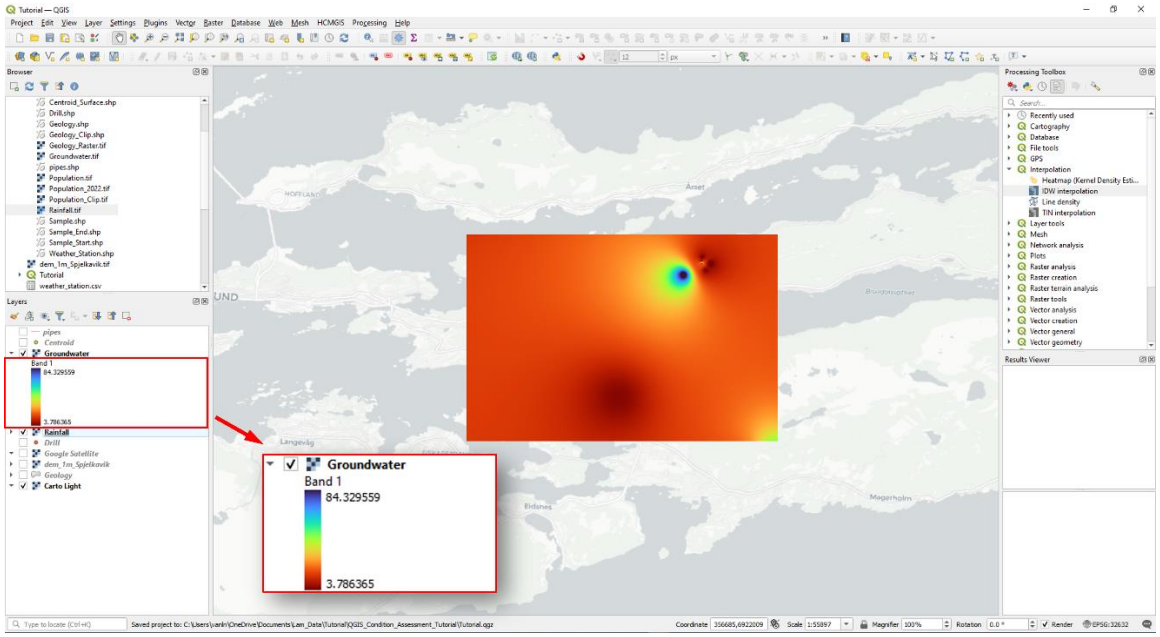


Figure 2-63. The groundwater map

➢ *Step 3:* Assign the values of the groundwater map for the centroid point of the pipe layer (the steps from **Figure 2-48** to **Figure 2-50**). The result is shown in (**Figure 2-64**).

pipes — Features Total: 11895, Filtered: 11895, Selected: 0

	ỳp	Connection	PipeForm	Material	Rainfall	Geology	GroundWate
1		Others	S	PVC	148.04	Geology_3	9.96
2		MUF	S	BET	150.09	Geology_3	11.28
3		MUF	S	PVC	147.05	Geology_3	15.13
4		MUF	S	BET	151.22	Geology_3	19.02
5		MUF	S	BET	151.53	Geology_3	22.36
6		MUF	S	BET	151.54	Geology_1	26.18
7		MUF	S	BET	152.64	Geology_1	26.51
8		MUF	S	BET	151.46	Geology_3	20.04
9		MUF	S	BET	150.79	Geology_3	23.54
10		MUF	S	BET	151.27	Geology_3	24.00
11		MUF	S	BET	152.85	Geology_1	25.75
12		MUF	S	BET	151.64	Geology_3	24.36

Show All Features

Figure 2-64. Assigning the groundwater value for the pipe layer

e. Soil type

Soil type is one of the significant factors in the deterioration models because it affects runoff

generation and groundwater and the influence of soil on sewers with larger sizes or buried deeper is more significant than the others (Beheshti et al., 2015). The soil type in the study area is shown in **Figure 2-65**.

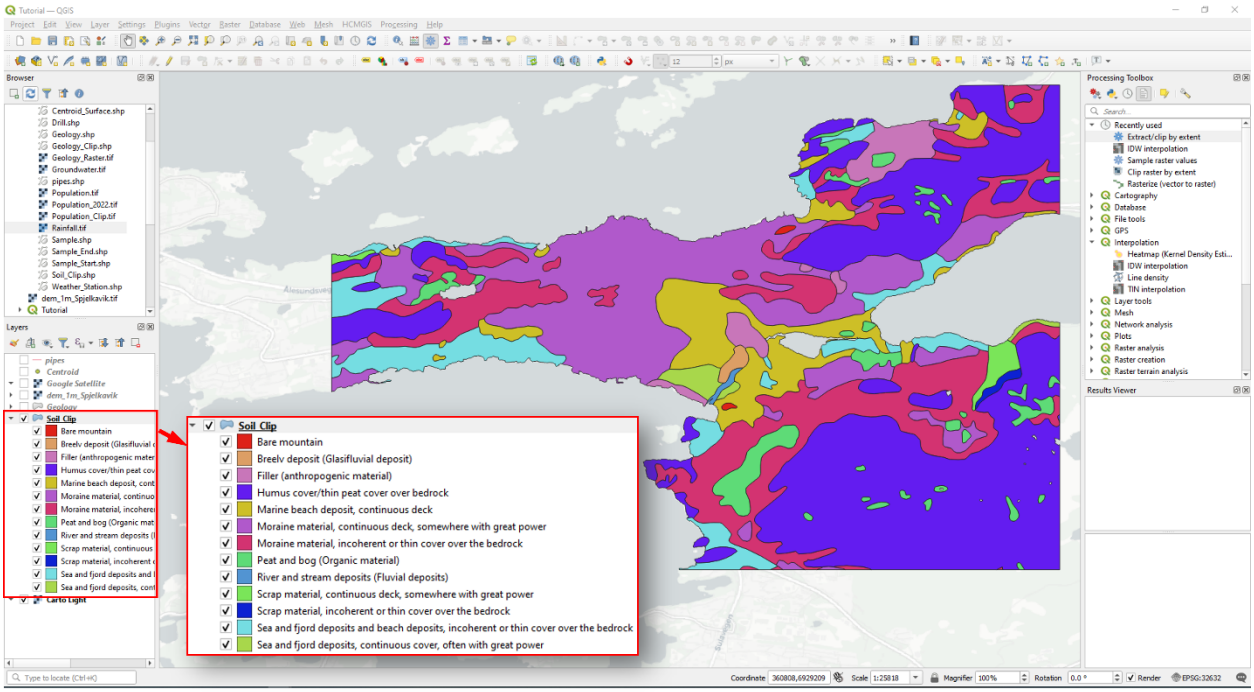


Figure 2-65. The soil type in the study area

The soil type was provided as a vector map (similar to the geological map). Therefore, the process for calculating and assigning soil type is similar to the geological map (steps from **Figure 2-51** to **Figure 2-54**). The result is shown in **Figure 2-66**.

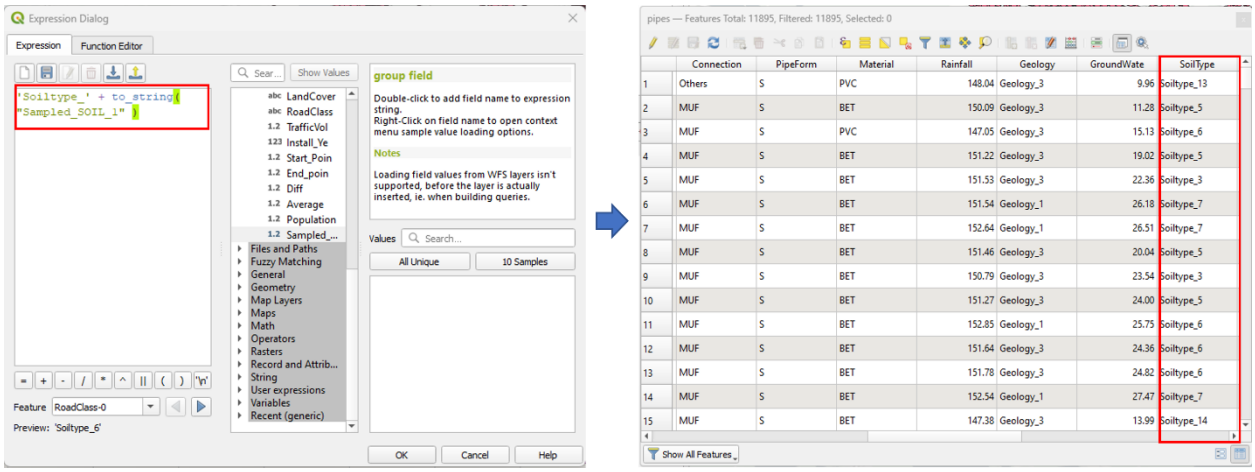


Figure 2-66. Assigning the soil type value for the pipe layer

f. Building area

Sewer pipes under building areas are more vulnerable to deterioration than those found in

non-built areas (Hawari et al., 2020). In this tutorial, the attribute of sewer pipe will be assigned as *"Yes"* for pipes that are under the building, and *"No"* is assigned for remaining sewer pipes. The steps for assigning pipes under the building are shown as follows:

- *Step 1:* Import the building layer into QGIS (**Figure 2-67**).

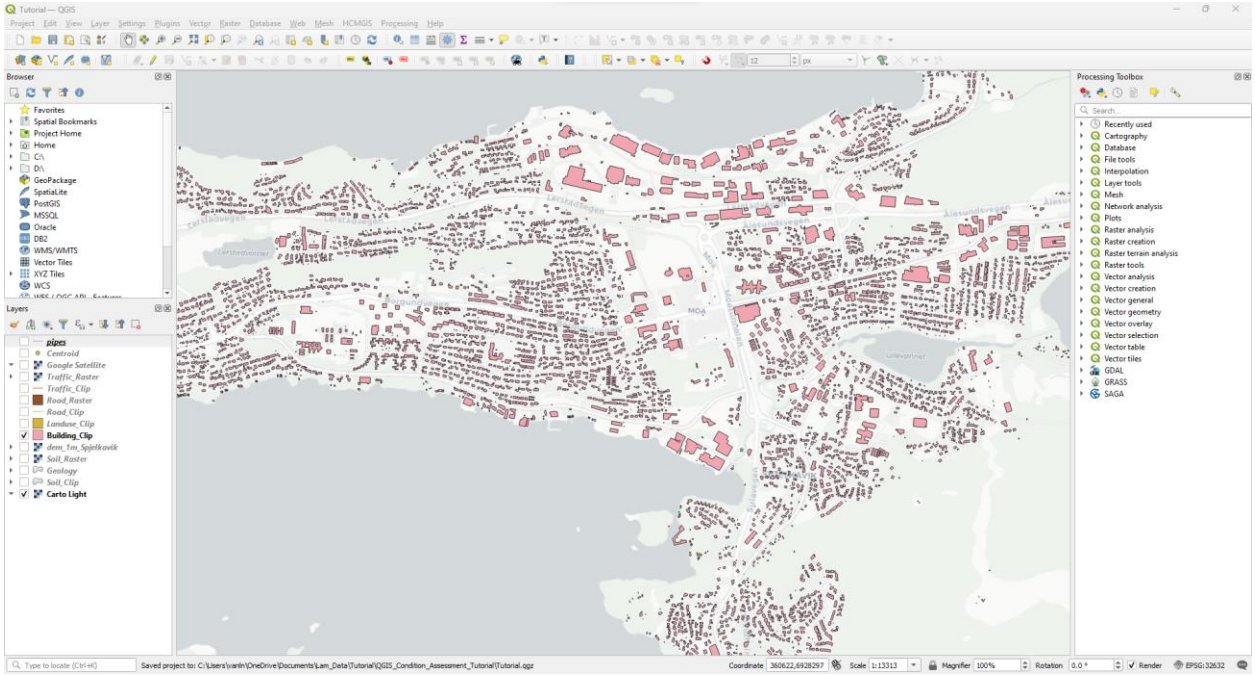


Figure 2-67. Importing building layer

- *Step 2:* Find pipes that intersect with the building layer (**Figure 2-68**).

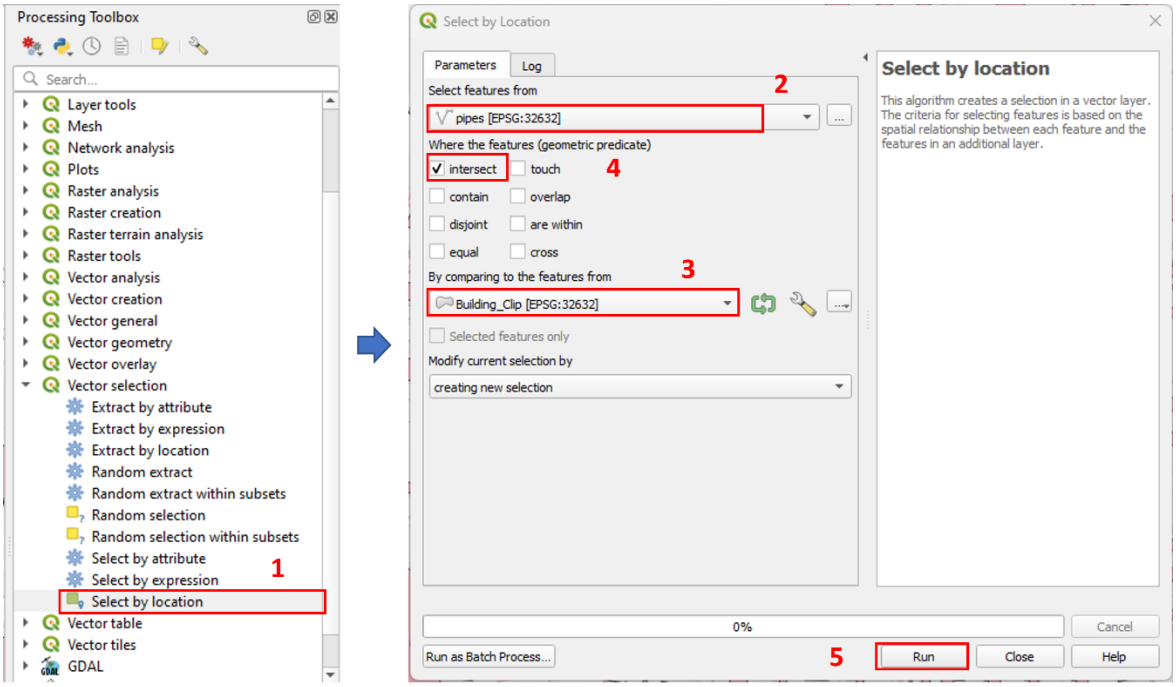


Figure 2-68. Selecting a layer based on the location

➢ *Step 3:* Create a new column, called *"Building"*, and assign the value *"Yes"* for the pipes that intersect with the building layer (**Figure 2-69**).

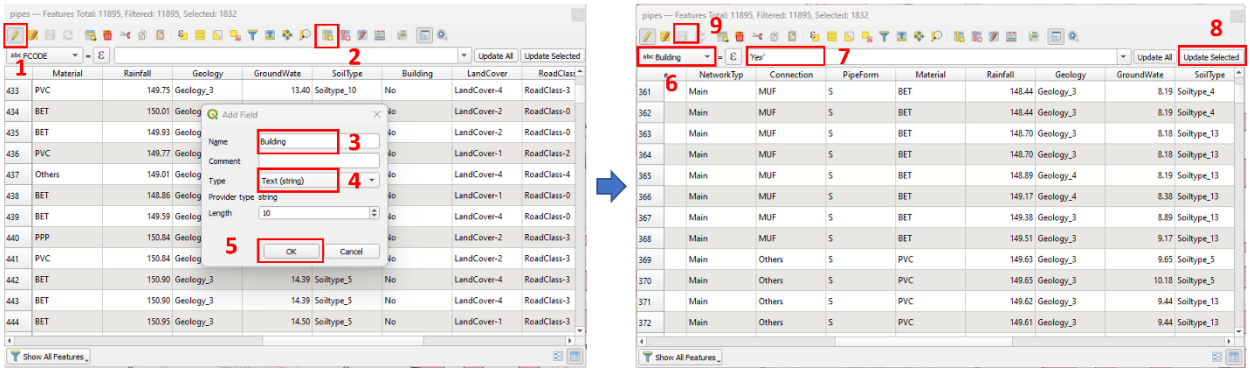
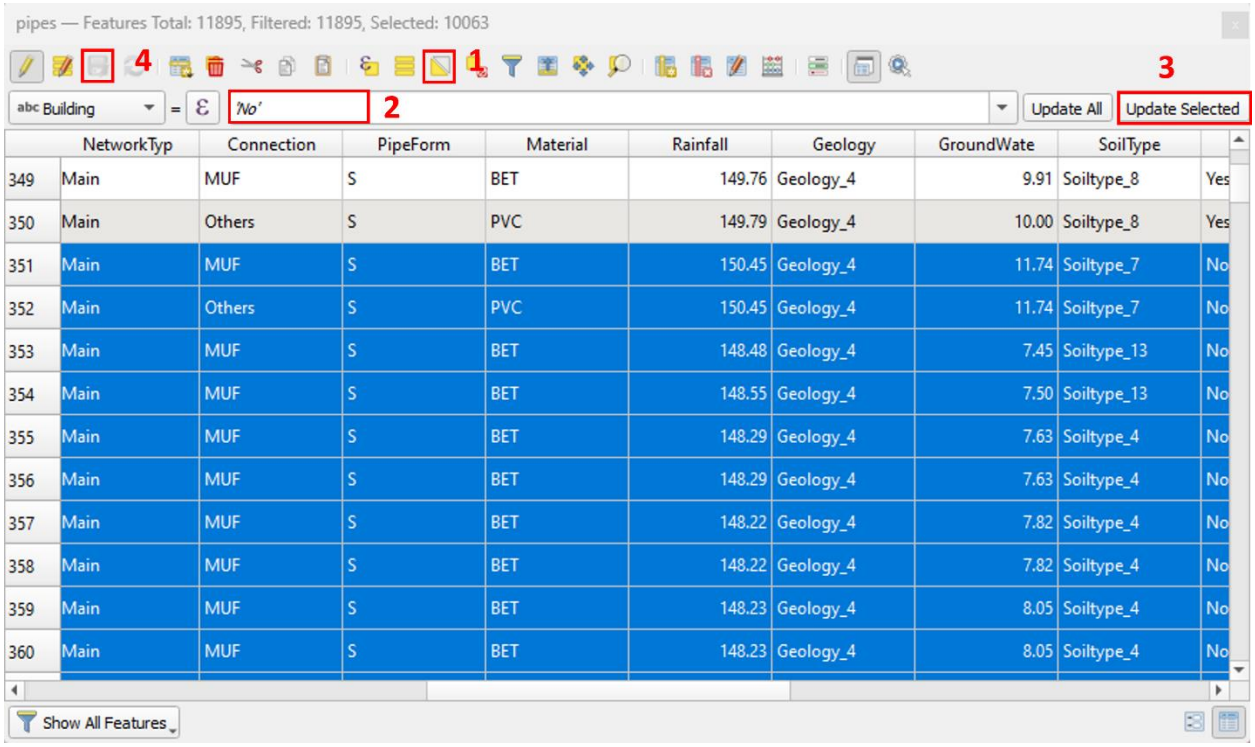


Figure 2-69. Assigning for sewer pipes under the building

➢ *Step 4:* Select the remaining pipes and assign the value *"No"* to them (**Figure 2-70**).



	NetworkTyp	Connection	PipeForm	Material	Rainfall	Geology	GroundWate	SoilType	
349	Main	MUF	S	BET	149.76	Geology_4	9.91	Soiltype_8	Yes
350	Main	Others	S	PVC	149.79	Geology_4	10.00	Soiltype_8	Yes
351	Main	MUF	S	BET	150.45	Geology_4	11.74	Soiltype_7	No
352	Main	Others	S	PVC	150.45	Geology_4	11.74	Soiltype_7	No
353	Main	MUF	S	BET	148.48	Geology_4	7.45	Soiltype_13	No
354	Main	MUF	S	BET	148.55	Geology_4	7.50	Soiltype_13	No
355	Main	MUF	S	BET	148.29	Geology_4	7.63	Soiltype_4	No
356	Main	MUF	S	BET	148.29	Geology_4	7.63	Soiltype_4	No
357	Main	MUF	S	BET	148.22	Geology_4	7.82	Soiltype_4	No
358	Main	MUF	S	BET	148.22	Geology_4	7.82	Soiltype_4	No
359	Main	MUF	S	BET	148.23	Geology_4	8.05	Soiltype_4	No
360	Main	MUF	S	BET	148.23	Geology_4	8.05	Soiltype_4	No

Figure 2-70. Assigning for remaining sewer pipes

g. Land use

Land use affects soil infiltration rate, evapotranspiration, or surface runoff and has been considered a variable in water quality change that has a strong correlation with the current condition of sewer pipes (de Oliveira et al., 2017). The land use map in the study area is shown in **Figure 2-71**.

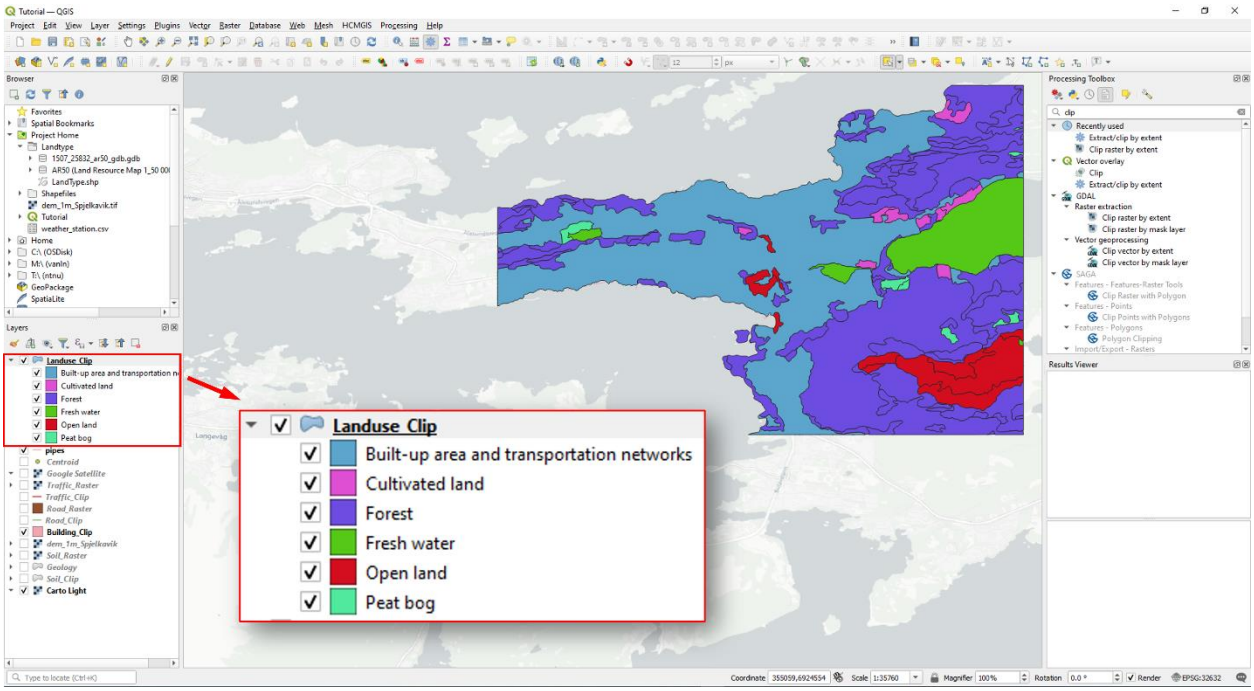


Figure 2-71. The land use map in the study area

The land use was provided as a vector map, and the processing procedure is similar to processing geological data (steps from **Figure 2-51** to **Figure 2-54**). The steps are shown in **Figure 2-72**.

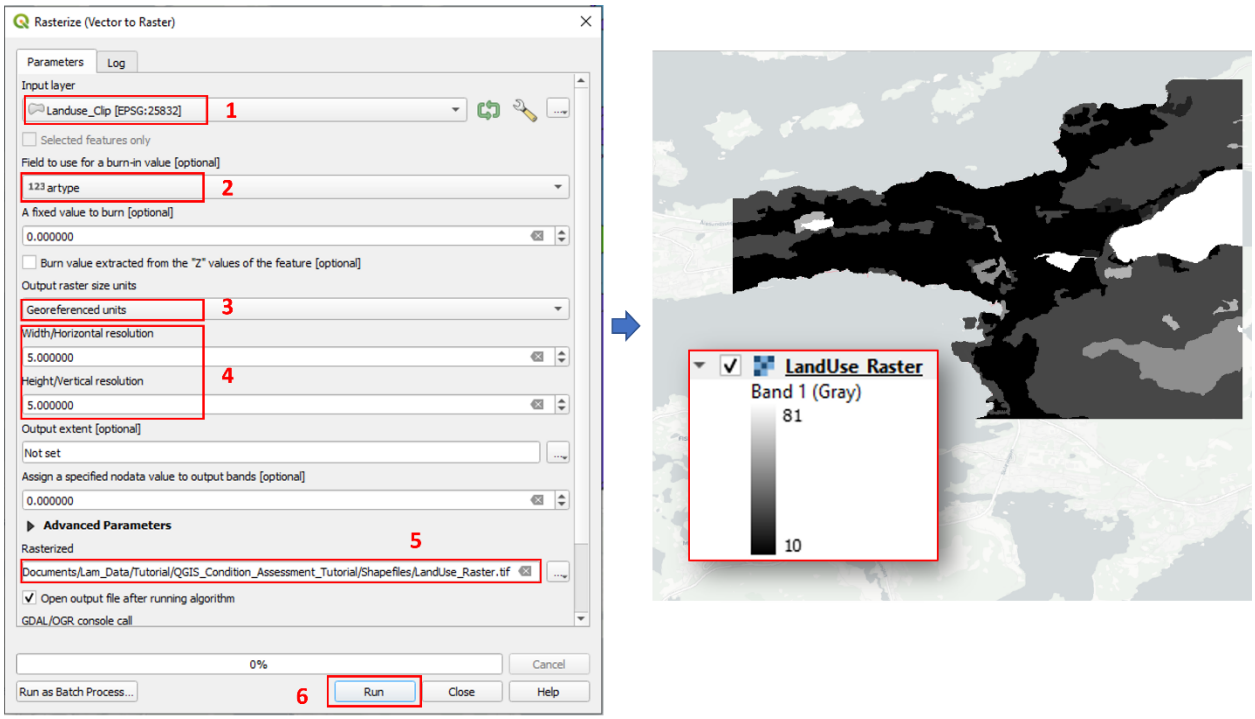


Figure 2-72. Converting vector to raster

The result is shown in **Figure 2-73**.

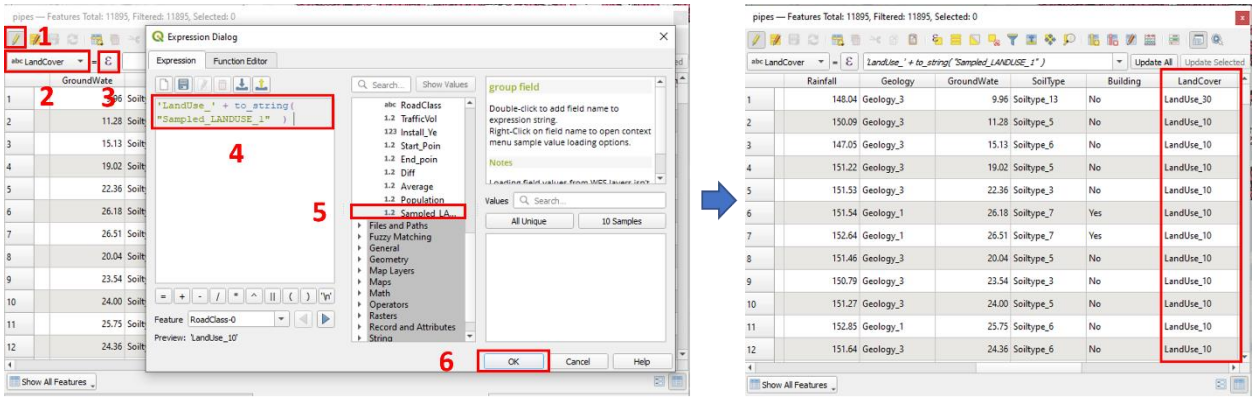


Figure 2-73. Assigning the land use value for the pipe layer

h. Distance to road

There is no universal guideline for selecting the distance to the road in modeling the sewer deterioration process. For instance, while Ahmadi et al. (2014) only considered pipes located under roads, the ratio of pipe length along the road was counted in the study by Yin et al. (2020). Remarkably, Laakso et al. (2018) emphasized the pipes close to the tree (about 5 m) had a higher deteriorated degree compared to further ones, and the pipes far from roads will suffer from less influence compared to near ones.

In this tutorial, we consider 5m-range road distances for the first road class; therefore, larger distances can be accepted for classifying further pipes into different road classes. Finally, five ordinal road classes were used based on the road's buffers of 0-5 m, 5-10 m, 10-20 m, 20-50 m, and >50 m. The steps for buffering the road are presented as follows:

- *Step 1:* Import the road network into the QGIS (**Figure 2-74**).

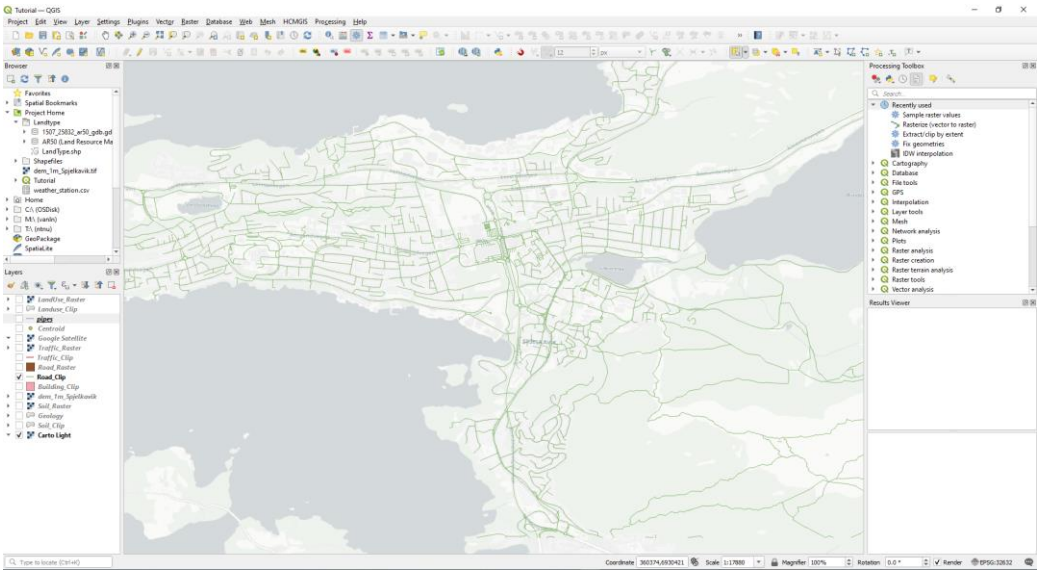


Figure 2-74. Importing the road network

➢ *Step 2:* Buffer the road network with a distance of 5 m (**Figure 2-75**).

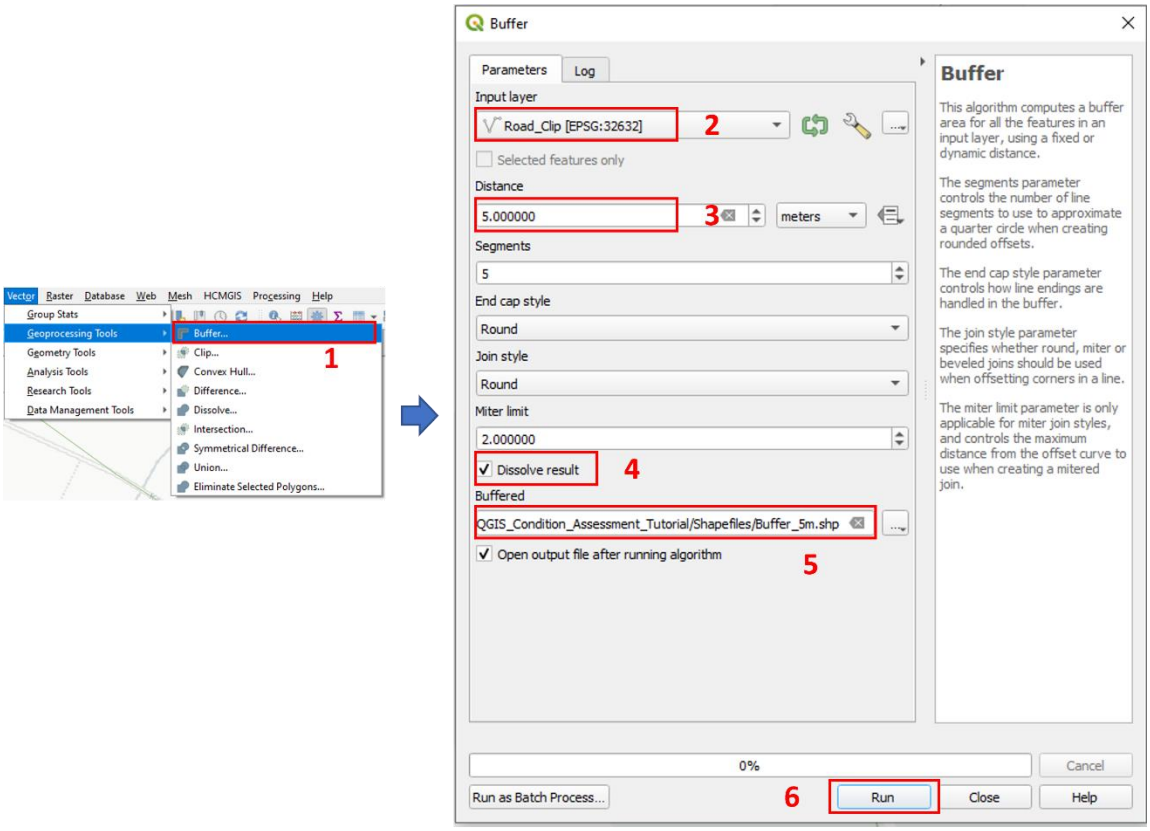


Figure 2-75. Buffering the road network in QGIS

➢ *Step 3:* Check the result and do the same process with the distances of 5-10 m, 10-20 m, 20-50 m, and >50 m (**Figure 2-76**).

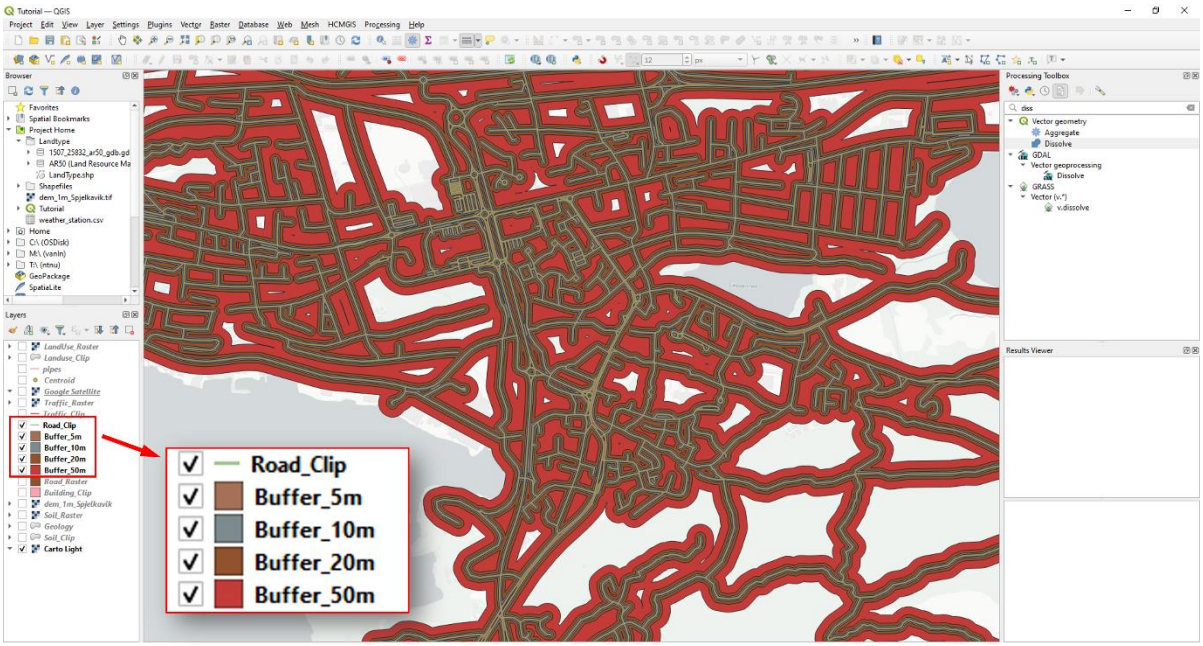


Figure 2-76. Road buffering in QGIS

➢ *Step 4:* Get the different parts between buffer classes (**Figure 2-77**).

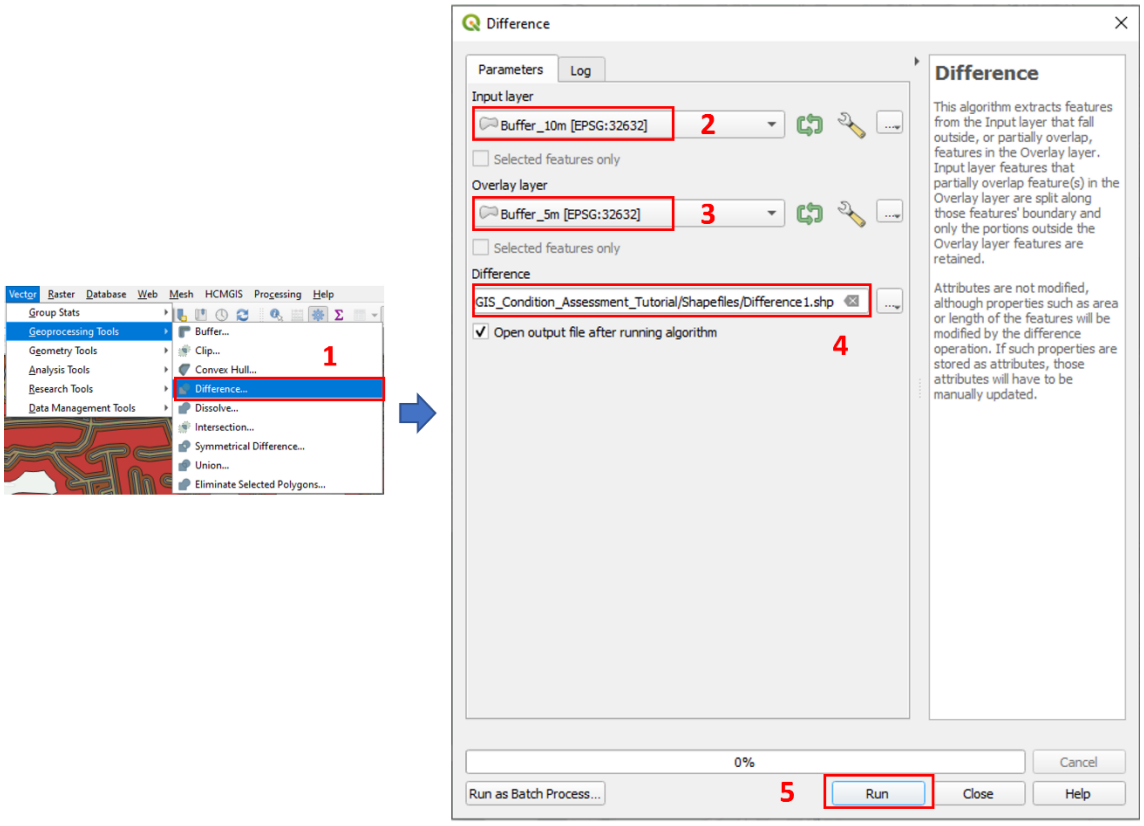


Figure 2-77. Getting the difference between vector classes

➢ *Step 5:* Check the result of the difference between the 5m-buffer and the 10m-buffer classes (**Figure 2-78**).

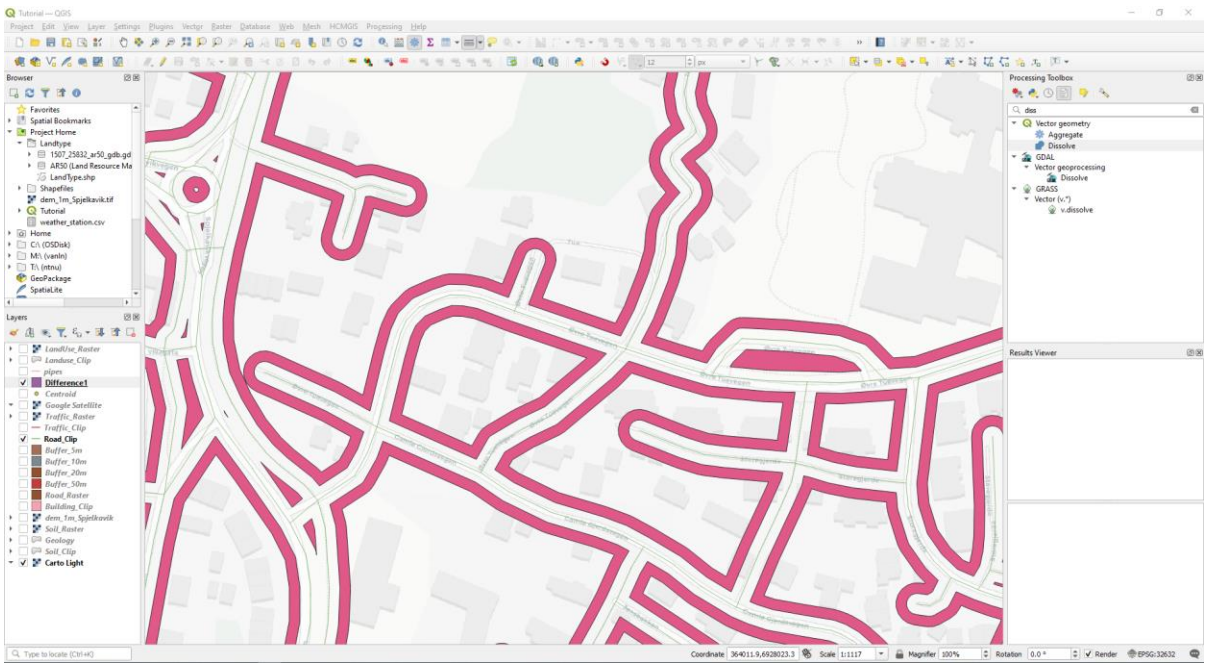


Figure 2-78. The difference between the 5m-buffer and the 10m-buffer classes

➢ *Step 6:* Implement the same process with 20m-buffer and 50m-buffer classes (**Figure 2-79**).

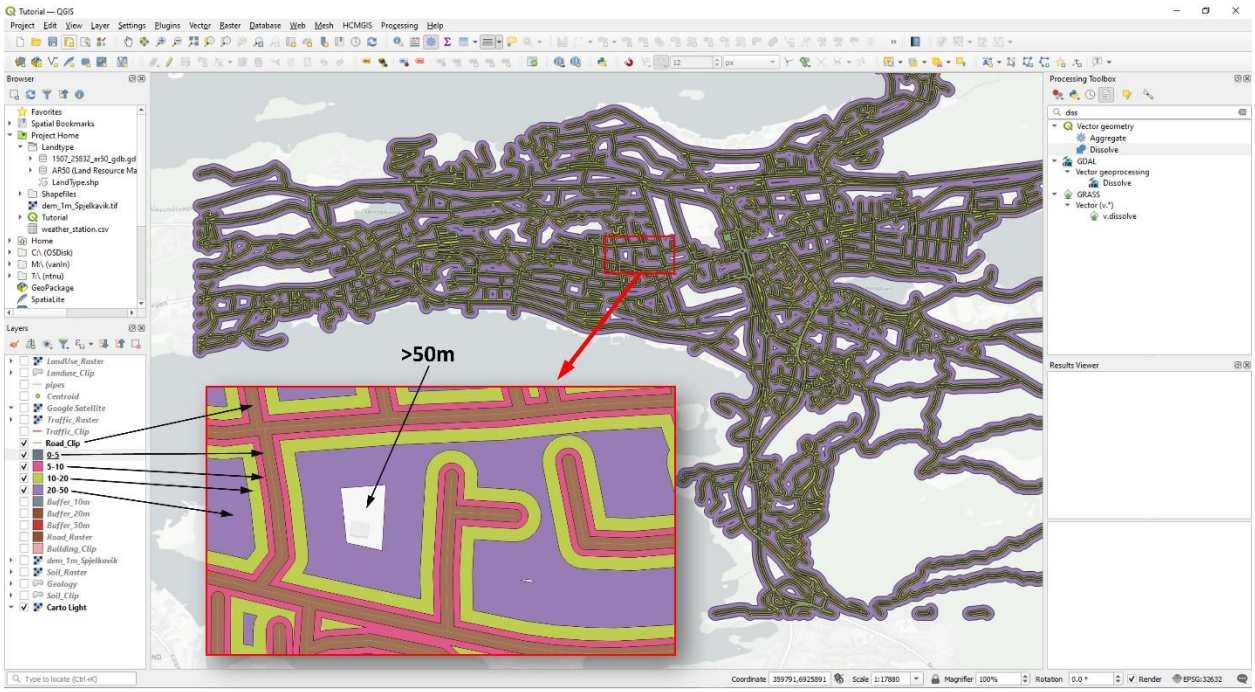


Figure 2-79. Distance to the road classes in QGIS

➢ *Step 7:* Merge the buffer classes (**Figure 2-80**).

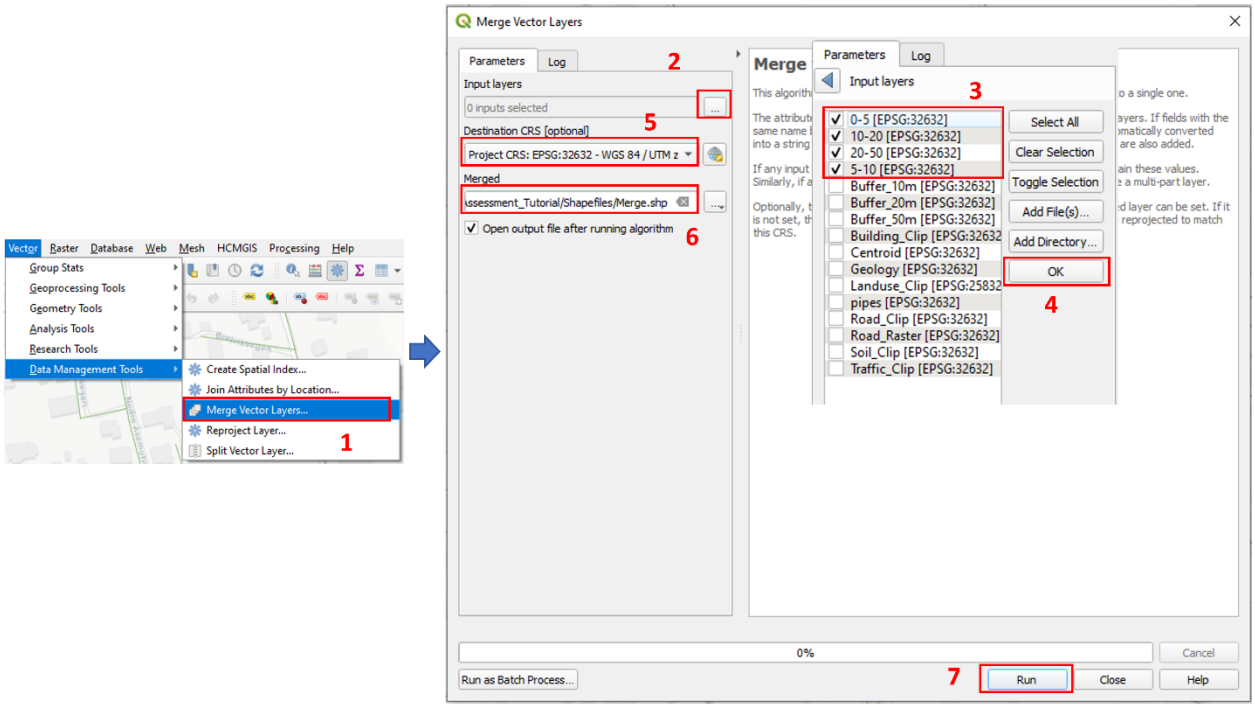


Figure 2-80. Merging classes in QGIS

➢ *Step 8:* Assign values for the buffer classes before converting them to raster type

(**Figure 2-81**).

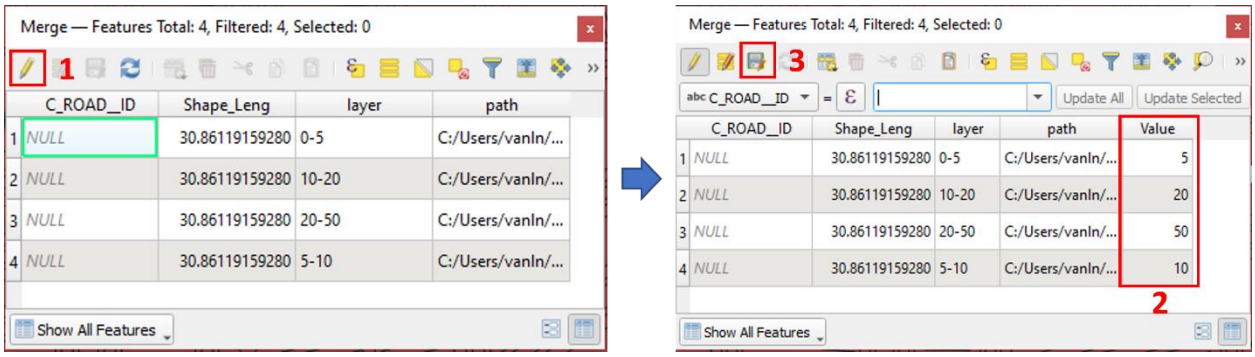


Figure 2-81. Assigning value for buffer classes

- *Step 9:* Convert the buffer vector layer to raster based on the assigned attribute in the previous step (**Figure 2-82**).

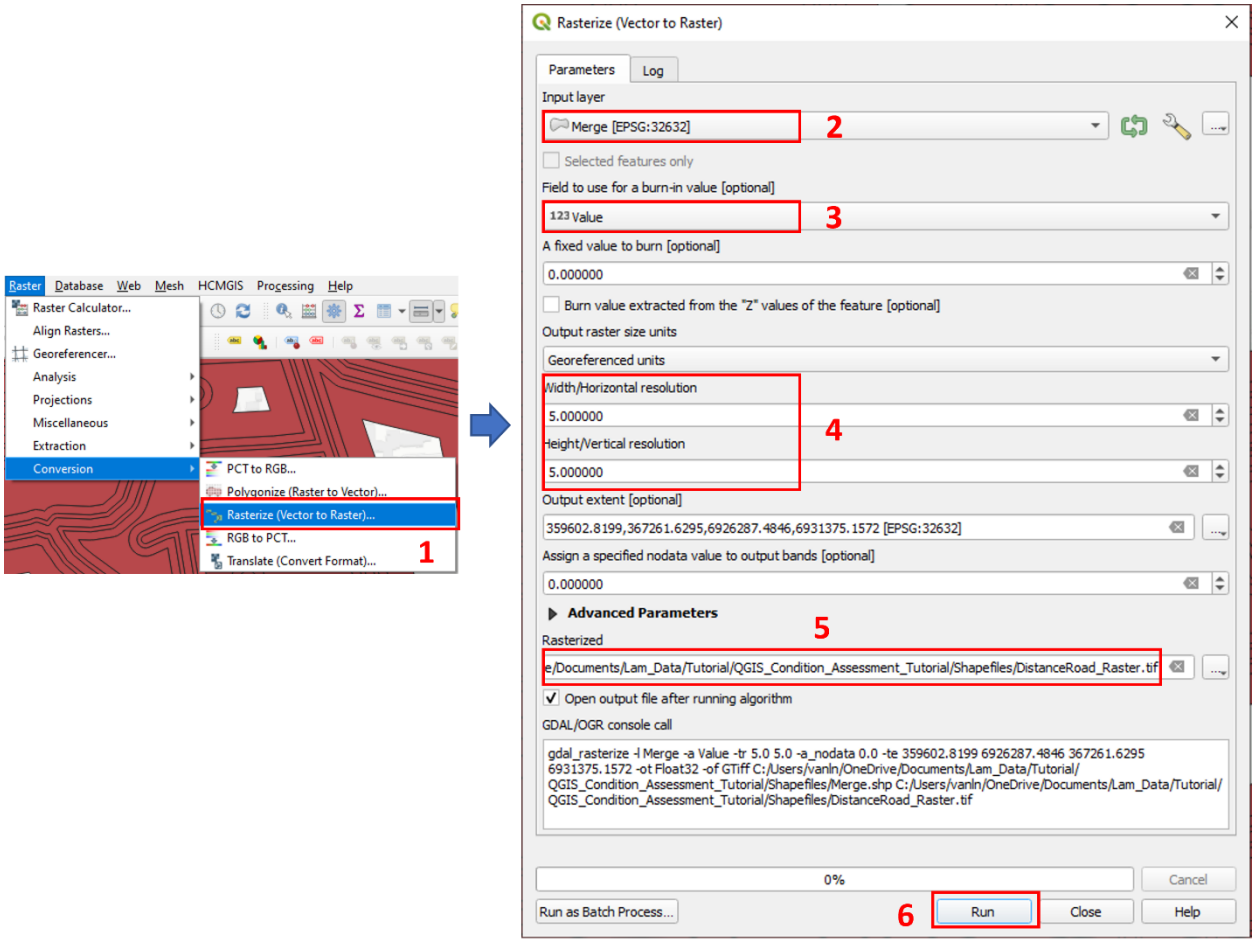


Figure 2-82. Converting the buffer class to raster

- *Step 10:* Join the value of the above raster class to the centroid point of the pipe layer (**Figure 2-83**).

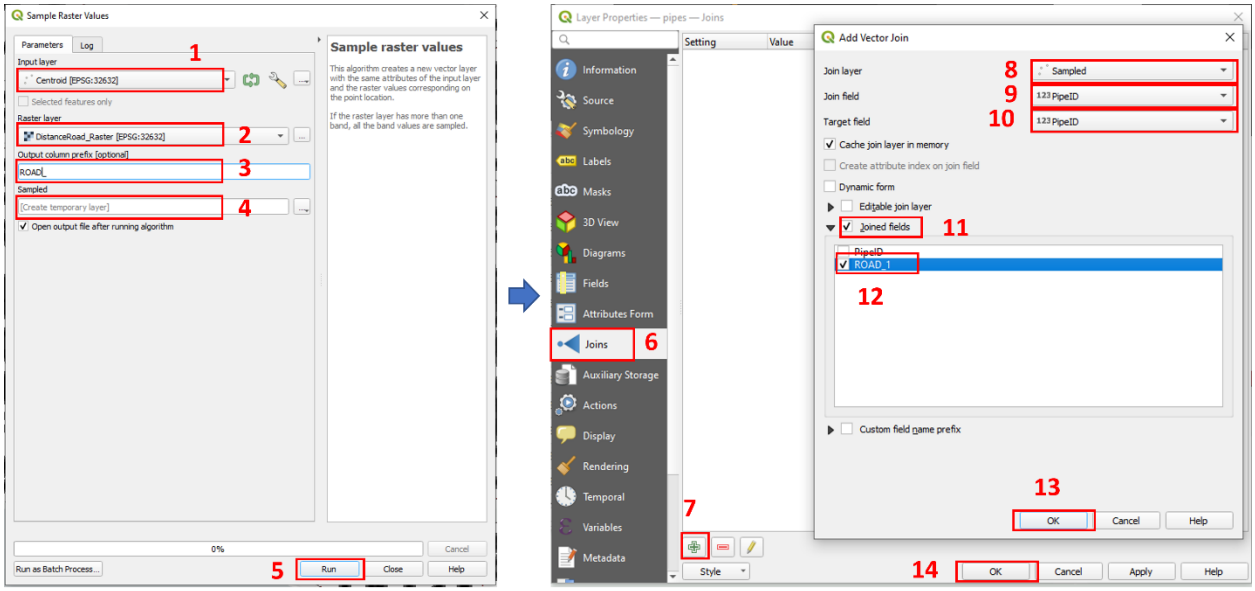


Figure 2-83. Joining the road class for the pipe layer

➢ *Step 11:* Assign the value of the road class for the pipe layer (**Figure 2-84**).

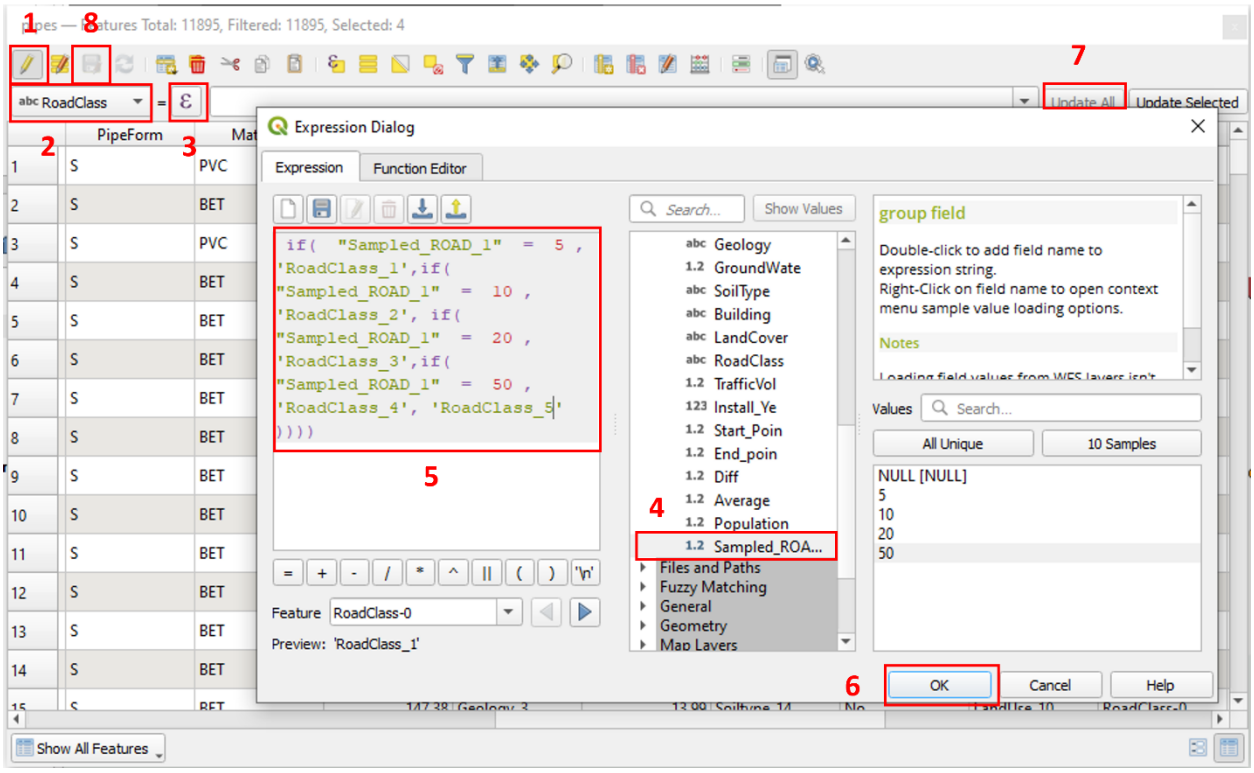
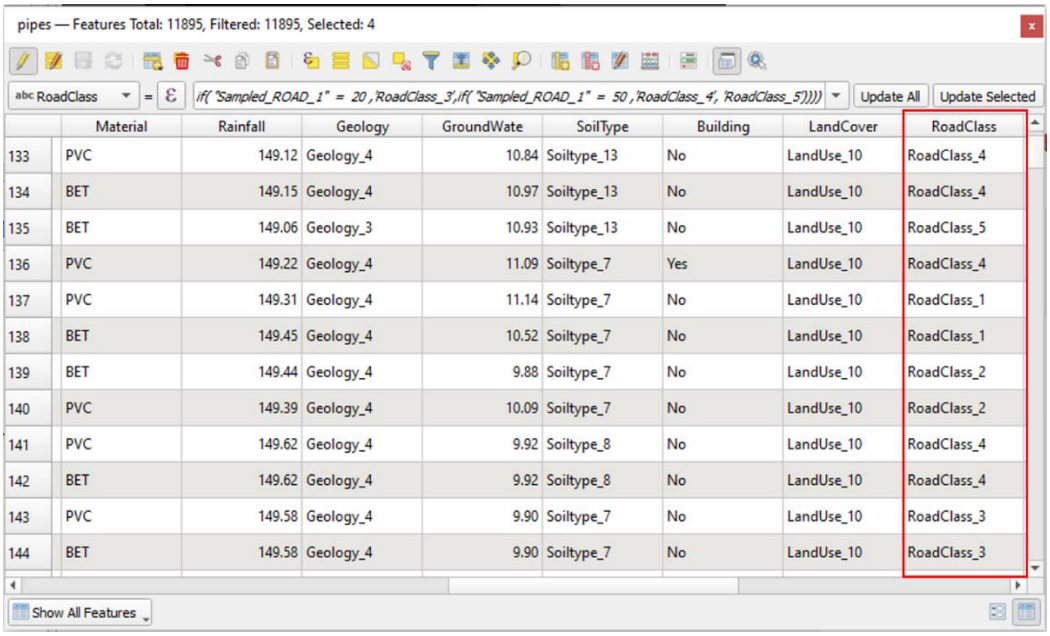


Figure 2-84. Assigning the road class for the pipe layer

➢ *Step 12:* Check the result (**Figure 2-85**).

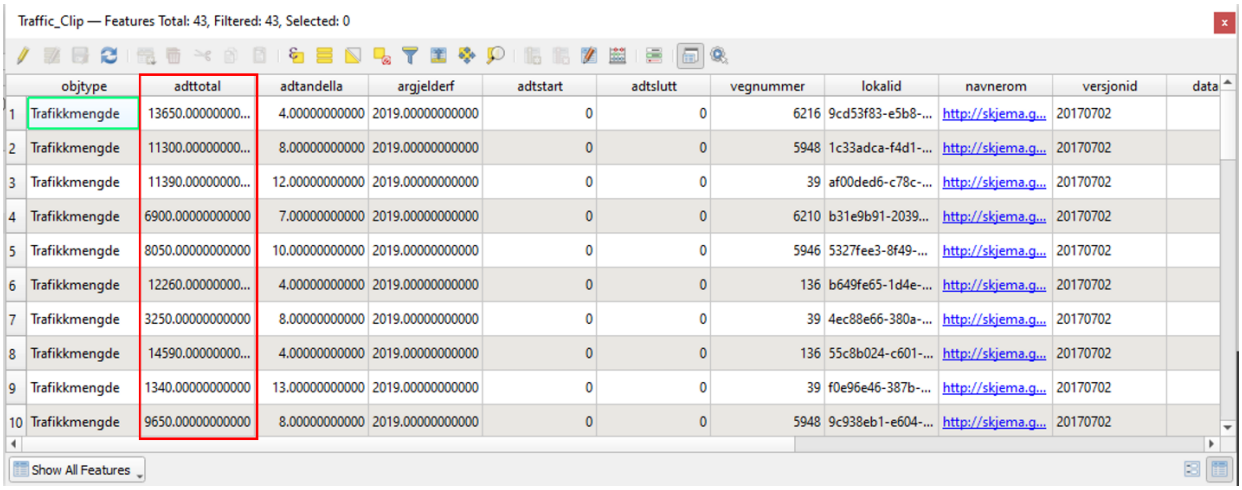


	Material	Rainfall	Geology	GroundWate	SoilType	Building	LandCover	RoadClass
133	PVC	149.12	Geology_4	10.84	Soiltype_13	No	LandUse_10	RoadClass_4
134	BET	149.15	Geology_4	10.97	Soiltype_13	No	LandUse_10	RoadClass_4
135	BET	149.06	Geology_3	10.93	Soiltype_13	No	LandUse_10	RoadClass_5
136	PVC	149.22	Geology_4	11.09	Soiltype_7	Yes	LandUse_10	RoadClass_4
137	PVC	149.31	Geology_4	11.14	Soiltype_7	No	LandUse_10	RoadClass_1
138	BET	149.45	Geology_4	10.52	Soiltype_7	No	LandUse_10	RoadClass_1
139	BET	149.44	Geology_4	9.88	Soiltype_7	No	LandUse_10	RoadClass_2
140	PVC	149.39	Geology_4	10.09	Soiltype_7	No	LandUse_10	RoadClass_2
141	PVC	149.62	Geology_4	9.92	Soiltype_8	No	LandUse_10	RoadClass_4
142	BET	149.62	Geology_4	9.92	Soiltype_8	No	LandUse_10	RoadClass_4
143	PVC	149.58	Geology_4	9.90	Soiltype_7	No	LandUse_10	RoadClass_3
144	BET	149.58	Geology_4	9.90	Soiltype_7	No	LandUse_10	RoadClass_3

Figure 2-85. Road class for sewer pipe

i. Traffic volume

Road traffic has been shown to have an impact on the deterioration process of sewers. Studies have shown that the condition of sewers located under roads as well as those close to roads are significantly affected (Salman & Salem, 2012). The traffic volume provided by the NMA is the traffic flow on Norwegian major roads, this is annually averaged data with an annual average daily traffic count, and the ratio of large vehicles (**Figure 2-86**).



	objtype	adttotal	adtandella	argjelderf	adtstart	adtslutt	vegnummer	lokalid	navnerom	versjonid	data
1	Trafikkmengde	13650.00000000...	4.00000000000	2019.00000000000	0	0	6216	9cd53f83-e5b8-...	http://skjema.g...	20170702	
2	Trafikkmengde	11300.00000000...	8.00000000000	2019.00000000000	0	0	5948	1c33adca-f4d1-...	http://skjema.g...	20170702	
3	Trafikkmengde	11390.00000000...	12.00000000000	2019.00000000000	0	0	39	af00ded6-c78c-...	http://skjema.g...	20170702	
4	Trafikkmengde	6900.00000000000	7.00000000000	2019.00000000000	0	0	6210	b31e9b91-2039...	http://skjema.g...	20170702	
5	Trafikkmengde	8050.00000000000	10.00000000000	2019.00000000000	0	0	5946	5327fee3-8f49-...	http://skjema.g...	20170702	
6	Trafikkmengde	12260.00000000...	4.00000000000	2019.00000000000	0	0	136	b649fe65-1d4e-...	http://skjema.g...	20170702	
7	Trafikkmengde	3250.00000000000	8.00000000000	2019.00000000000	0	0	39	4ec88e66-380a-...	http://skjema.g...	20170702	
8	Trafikkmengde	14590.00000000...	4.00000000000	2019.00000000000	0	0	136	55c8b024-c601-...	http://skjema.g...	20170702	
9	Trafikkmengde	1340.00000000000	13.00000000000	2019.00000000000	0	0	39	f0e96e46-387b-...	http://skjema.g...	20170702	
10	Trafikkmengde	9650.00000000000	8.00000000000	2019.00000000000	0	0	5948	9c938eb1-e604-...	http://skjema.g...	20170702	

Figure 2-86. Traffic volume in the study area

The steps for processing traffic data are presented as follows:

- *Step 1:* Import the layer containing traffic volume data into QGIS.
- *Step 2:* Buffer the traffic volume with the distance value of 5m (**Figure 2-75**).

➢ *Step 3:* Convert traffic volume from the vector layer to the raster layer based on their volume (**Figure 2-87**).

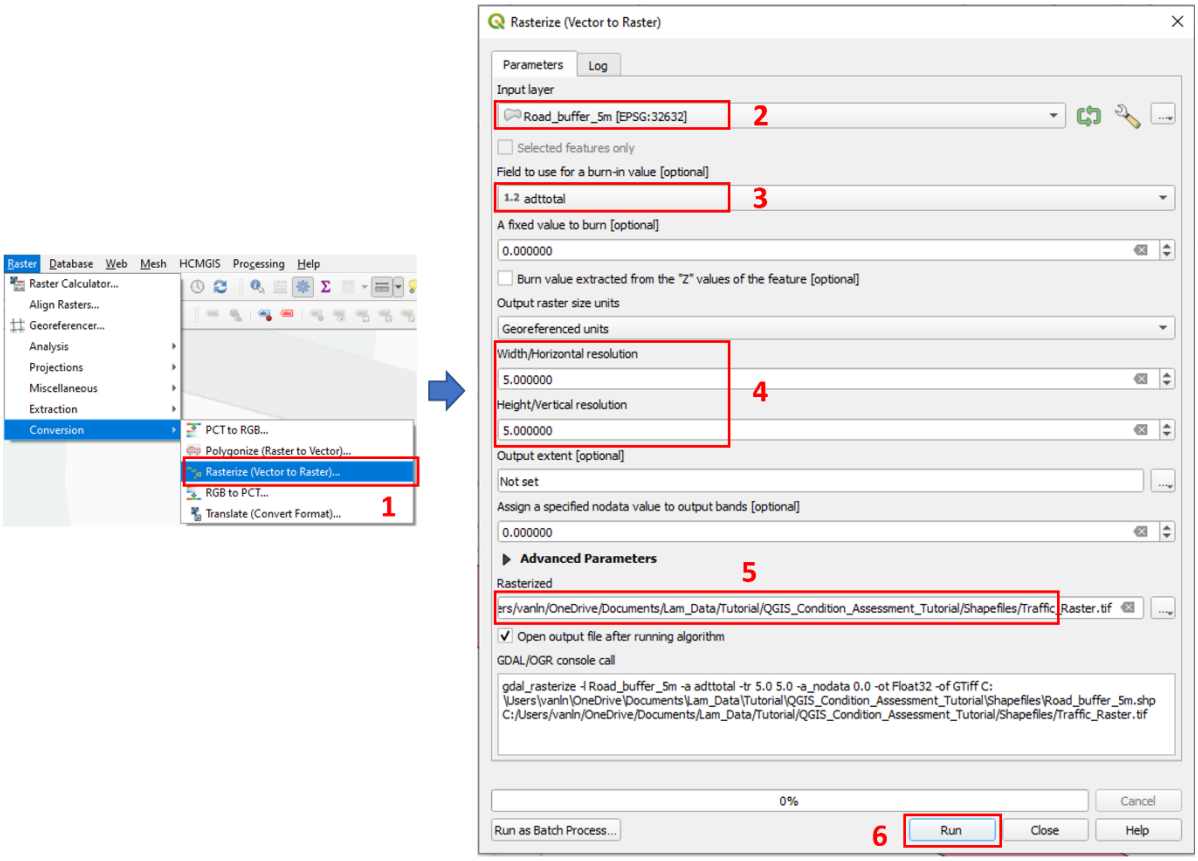


Figure 2-87. Converting the traffic flow to raster type

➢ *Step 3:* Assign the traffic volume for the centroid point of the pipe layer (**Figure 2-88**).

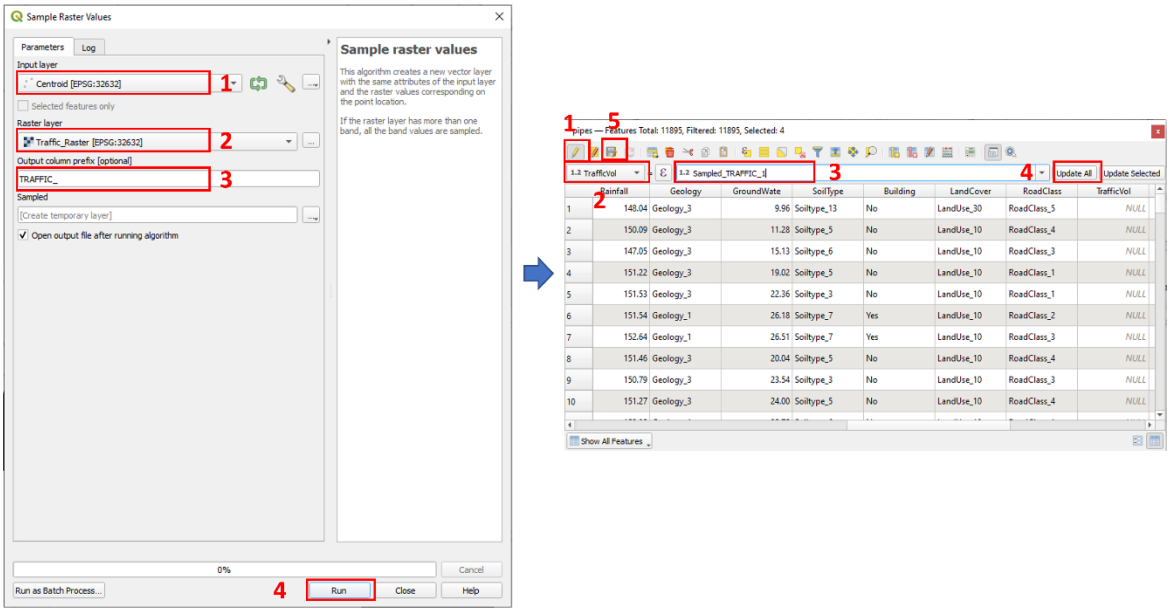


Figure 2-88. Assigning the traffic volume for the pipe layer

➢ *Step 4:* Assign the value equal to 0 for the sewer pipes that do not have the traffic volume value (**Figure 2-89**).

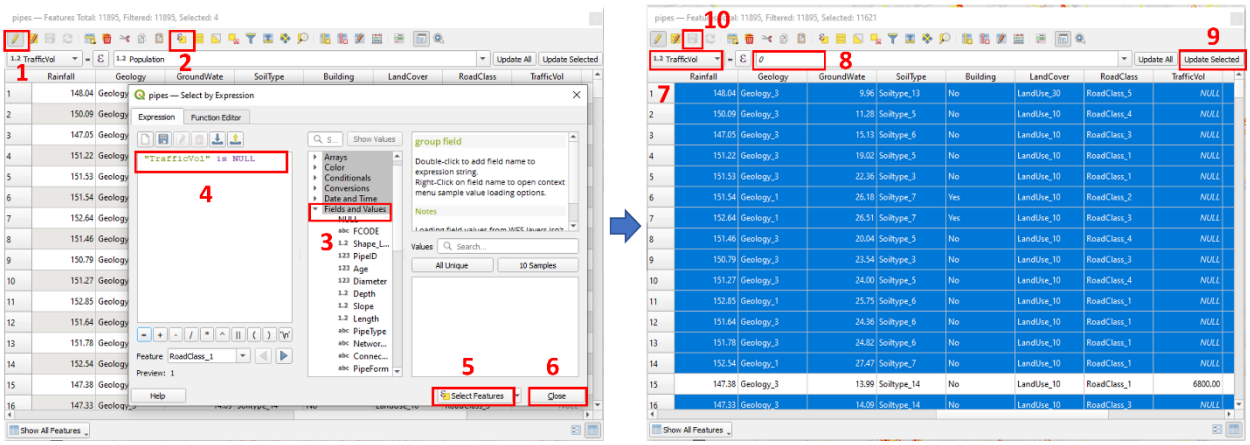
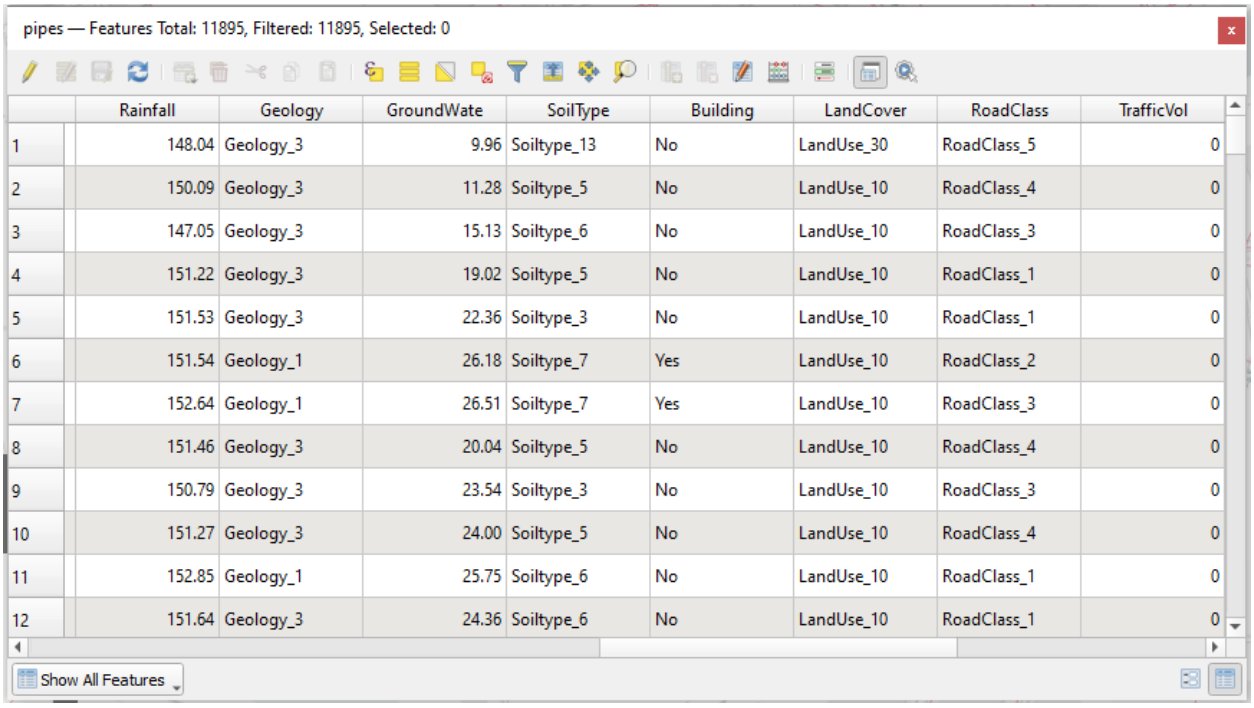


Figure 2-89. Processing null value

➢ *Step 5:* Check the result (**Figure 2-90**).



pipes — Features Total: 11895, Filtered: 11895, Selected: 0

	Rainfall	Geology	GroundWate	SoilType	Building	LandCover	RoadClass	TrafficVol
1	148.04	Geology_3	9.96	Soiltype_13	No	LandUse_30	RoadClass_5	0
2	150.09	Geology_3	11.28	Soiltype_5	No	LandUse_10	RoadClass_4	0
3	147.05	Geology_3	15.13	Soiltype_6	No	LandUse_10	RoadClass_3	0
4	151.22	Geology_3	19.02	Soiltype_5	No	LandUse_10	RoadClass_1	0
5	151.53	Geology_3	22.36	Soiltype_3	No	LandUse_10	RoadClass_1	0
6	151.54	Geology_1	26.18	Soiltype_7	Yes	LandUse_10	RoadClass_2	0
7	152.64	Geology_1	26.51	Soiltype_7	Yes	LandUse_10	RoadClass_3	0
8	151.46	Geology_3	20.04	Soiltype_5	No	LandUse_10	RoadClass_4	0
9	150.79	Geology_3	23.54	Soiltype_3	No	LandUse_10	RoadClass_3	0
10	151.27	Geology_3	24.00	Soiltype_5	No	LandUse_10	RoadClass_4	0
11	152.85	Geology_1	25.75	Soiltype_6	No	LandUse_10	RoadClass_1	0
12	151.64	Geology_3	24.36	Soiltype_6	No	LandUse_10	RoadClass_1	0

Show All Features

Figure 2-90. Traffic volume for sewer pipe

2.4. Sewer Database

In this study, the inspected grades were used as dependent variables for modeling the condition of the sewer pipes. The current conditions of the sewer pipes were assigned using damage scores obtained through the closed-circuit television (CCTV) method. Next, these damage scores were coded into damage classes representing the sewer conditions. According to

Haugen and Viak (2018), the conditions of sewer pipes in Norway are classified into five-grade scales based on their damage scores (**Table 2-3**).

Table 2-3. The condition classes of pipe

Damage class	Damage score	Sewer condition
Class 1	0 – 5	Very good status
Class 2	6 – 10	Good status
Class 3	11 – 20	Questionable status
Class 4	21 – 50	Bad status
Class 5	>50	Very bad status

The data in **Table 2-3** indicates that damage score or sewer condition can be used as the output for building condition assessment models. By using the damage score as the output, we will solve the regression problem, and by using the sewer condition as the output, the classification problem is defined. In this tutorial, we will deal with both types of problems.

In the study area, there are a total of 11855 sewer pipes including 5870 wastewater pipes, 5317 stormwater pipes, and 668 combined pipes. Among them, there are 465 pipes were inspected with the number of pipes in each condition represented in **Table 2-4**.

Table 2-4. The condition of inspected sewer pipelines

Sewer condition	Number of inspections
Very good status	231
Good status	24
Questionable status	29
Bad status	51
Very bad status	130

To build the sewer condition assessment models using machine learning algorithms, 80% of this data (equivalent to 372 sewers) is selected, and the remaining sewers (93 pipes) are used to validate the developed machine learning models.

2.5. Data Storage

After the sewer pipe layer is created and modified in QGIS, it can be exported and stored as a shapefile (*.shp). The shapefile format is a geospatial vector data format for GIS software. It is developed and regulated by Esri as a mostly open specification for data interoperability among Esri and other GIS software products. The shapefile format can spatially describe vector features: points, lines, and polygons.

The steps for exporting a layer to a shapefile are illustrated in **Figure 2-91**. Firstly, the user right-clicks on the exported layer, selects *Export* → *Save Feature As*… (step 1 in **Figure 2-91**),

selects data format and location (step 2 in **Figure 2-91**), and selects wanted CRS (step 3 in **Figure 2-91**).

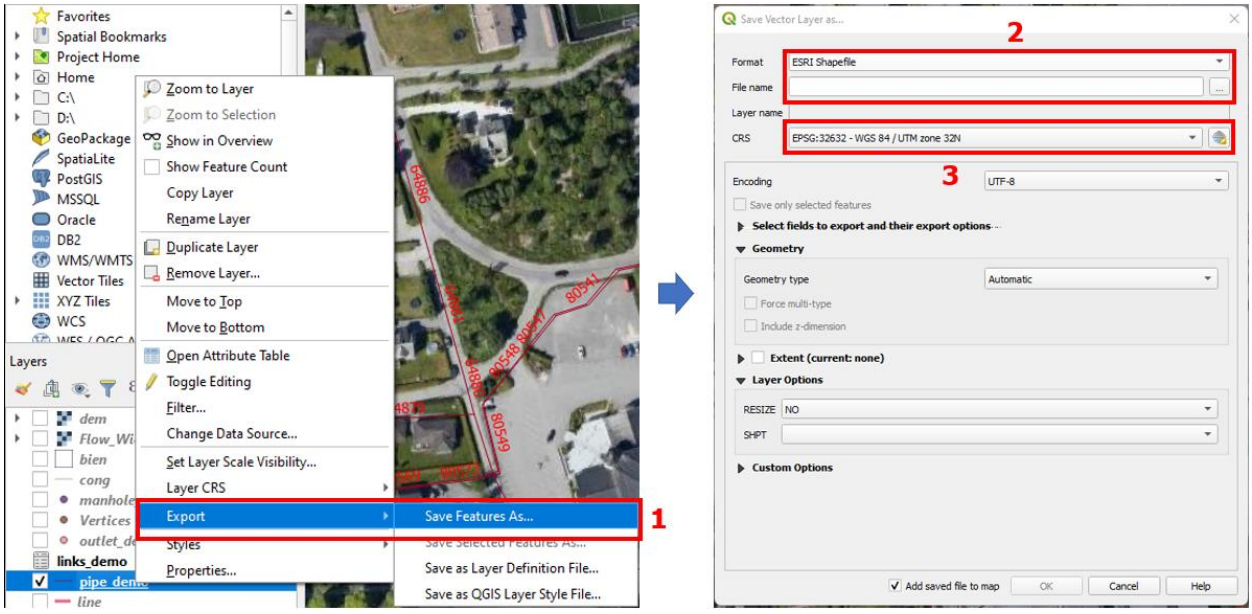


Figure 2-91. Export layer to a shapefile

The processed data can be exported into CSV format which is used as input data for implementing machine learning models. Steps for exporting a layer to a CSV file are shown in **Figure 2-92**.

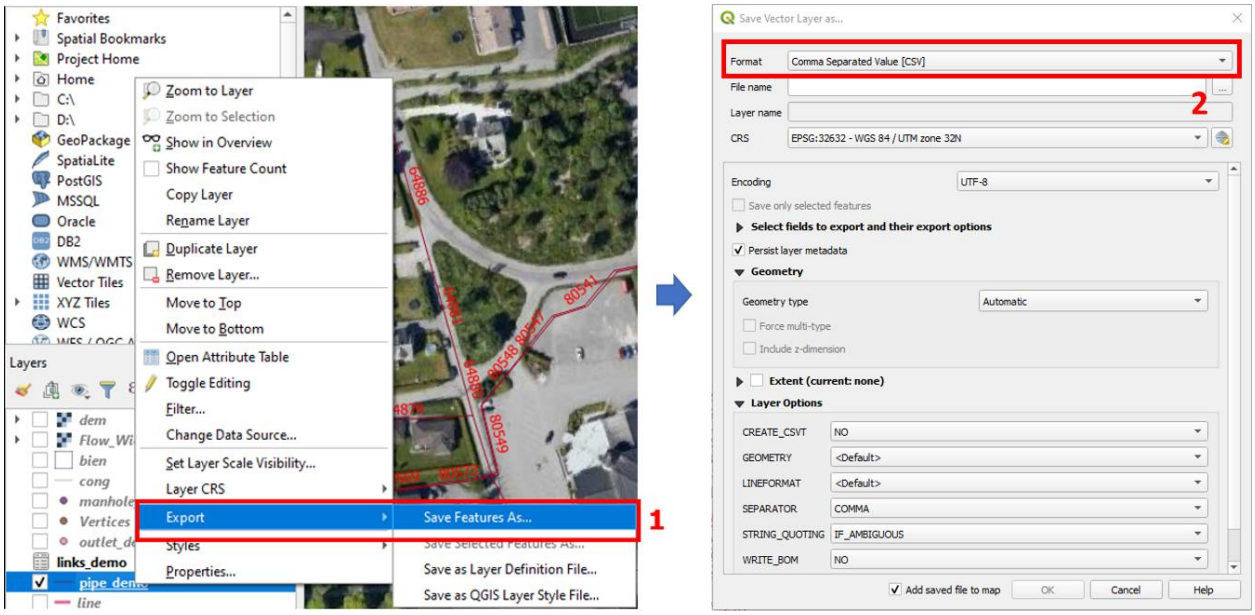


Figure 2-92. Export layer to a CSV file

The structure of data used for building the sewer condition assessment models is shown in **Figure 2-93**.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	PipeID	Age	Diameter	Depth	Slope	Length	PipeType	NetworkT	Connectic	PipeForm	Material	Rainfall	Geology	GroundW	SoilType	Building	LandCove	RoadClass	TrafficVol	Populatio	Score	Class
2	4839	48	225	-2.15	0.61	54.53	Combinec	Main	MUF	S	BET	153.11	Geology_	24.6	Soiltype_	No	LandUse_	RoadClass	0	213.16	35	Bad
3	4850	48	150	-2.03	0.04	48.94	Combinec	Main	MUF	S	BET	152.92	Geology_	24.55	Soiltype_	No	LandUse_	RoadClass	0	12.3	2222	Very bad
4	4919	67	250	-1.13	0.95	80.24	Wastewat	Main	MUF	S	BET	151.72	Geology_	33.44	Soiltype_	Yes	LandUse_	RoadClass	0	133.22	13	Questionable
5	4931	39	200	-1.91	-5.91	1.67	Combinec	Main	Others	S	PVC	151.13	Geology_	34.72	Soiltype_	No	LandUse_	RoadClass	0	141.42	67	Very bad
6	4956	46	160	-2.1	0.74	23.56	Stormwat	Main	MUF	S	PVC	146.99	Geology_	14.87	Soiltype_	No	LandUse_	RoadClass	0	71.74	176	Very bad
7	5177	55	160	-3.55	2.96	18.02	Combinec	Main	MUF	S	PVC	144.38	Geology_	19.81	Soiltype_	No	LandUse_	RoadClass	0	209.06	0	Very good
8	5236	51	600	-1.6	-0.78	35.58	Stormwat	Main	MUF	S	BET	136.57	Geology_	18.6	Soiltype_	No	LandUse_	RoadClass	0	79.93	145	Very bad
9	53252	43	1000	-2.27	2.53	21.93	Stormwat	Main	MUF	S	BET	148.93	Geology_	10.89	Soiltype_	Yes	LandUse_	RoadClass	0	72.76	57	Very bad
10	53275	43	1000	-2.85	4.94	27.12	Stormwat	Main	MUF	S	BET	149.06	Geology_	10.93	Soiltype_	No	LandUse_	RoadClass	0	72.76	45	Bad
11	53504	22	1000	-2.19	2.94	11.18	Stormwat	Main	MUF	S	BET	149.55	Geology_	11.59	Soiltype_	No	LandUse_	RoadClass	0	72.76	0	Very good
12	53571	43	800	-1.96	2.15	18.46	Stormwat	Main	MUF	S	BET	149.39	Geology_	11.34	Soiltype_	No	LandUse_	RoadClass	0	72.76	0	Very good
13	53898	34	200	-2.04	2.12	21.98	Wastewat	Main	Others	S	PVC	148.63	Geology_	5.81	Soiltype_	No	LandUse_	RoadClass	0	122.98	9	Good
14	53983	23	250	-1.94	0.61	57.49	Stormwat	Main	MUF	S	BET	148.35	Geology_	6.27	Soiltype_	No	LandUse_	RoadClass	0	83.01	25	Bad
15	53984	23	300	-2	1.47	5.92	Stormwat	Main	MUF	S	BET	148.41	Geology_	6.51	Soiltype_	No	LandUse_	RoadClass	0	83.01	113	Very bad
16	54399	28	250	-1.64	2.36	48.81	Wastewat	Main	Others	S	PVC	149.58	Geology_	12.45	Soiltype_	No	LandUse_	RoadClass	0	92.23	0	Very good
17	54433	52	600	-2.03	2.3	25.91	Stormwat	Main	MUF	S	BET	150.55	Geology_	14.23	Soiltype_	No	LandUse_	RoadClass	0	207.01	30	Bad
18	54434	52	200	-2	1.1	7.51	Wastewat	Main	MUF	S	BET	150.52	Geology_	14.14	Soiltype_	No	LandUse_	RoadClass	0	207.01	0	Very good
19	54435	52	200	-2.03	2.34	25.14	Wastewat	Main	MUF	S	BET	150.57	Geology_	14.23	Soiltype_	No	LandUse_	RoadClass	0	207.01	12	Questionable
20	54436	52	600	-2.03	1.84	34.75	Stormwat	Main	MUF	S	BET	150.63	Geology_	14.11	Soiltype_	No	LandUse_	RoadClass	0	207.01	15	Questionable
21	54437	52	200	-2.05	1.73	34.18	Wastewat	Main	MUF	S	BET	150.63	Geology_	14.11	Soiltype_	No	LandUse_	RoadClass	0	207.01	41	Bad

Figure 2-93. Structure data for sewer condition assessment in CSV file

Data used in this tutorial is available on GitHub: https://github.com/Lam-V-Nguyen/Sewer-Conditon-Assessment. Codes were written by python, the user, therefore, can open, modify, and run by using an integrated development environment (IDE) such as Spyder, PyCharm, Sublime Text, etc., In this tutorial, we used Spyder as the default IDE. All Python codes used in this tutorial can be found on the same GitHub link above.

3. Theory of Machine Learning Models Used

3.1. Classification-based Machine Learning Algorithms

a. Multi-Layer Perceptron Neural Network

Multi-layer Perceptron Neural Network (MLP) is a fully connected class of feedforward Artificial Neural Networks (ANN). This network has three sequential layers: the input layer, the hidden layer, and the output layer. The number of neurons in the input layer equals the number of factors, the number of neurons in the output layer represents the expected sewer status, and the number of hidden layers and hidden neurons is generally found by trial and error (Orhan et al., 2011). MLP can be used for both regression and classification problems.

Each neuron j in the hidden layer computes its input signals x_i and produces its output y_j based on the following equation:

$$y_j = f\left(\sum_{j,i=1}^{n} w_{ji}x_i + b_i\right) \quad (1)$$

where n is the number of sewer inspections in the training dataset; f is an activation function; w_{ji} and b are connection weight and bias, respectively.

Before training the MLP model, each factor (i.e., physical, and environmental factors) was

assigned to each neuron and a bias unit was added to the input layer. Then, randomly generated weights were assigned for elements in the input layer, the weighted sums for neurons were calculated and the activation functions were used to transfer the results to the hidden layer. Similar processes were implemented in the hidden layer and the results were driven to the output layer. The error (the difference between the predicted sewer condition status and the measured condition) was calculated and minimized at the output layer. Finally, the derivation of the error function (loss function) with each weight in the network was determined and the model was updated. This was an iterative process over multiple epochs until the ideal weights were determined and the final sewer condition status was predicted based on these weights.

b. Support Vector Machine

Support Vector Machine (SVM) was proposed by Cortes and Vapnik (1995) to distinctly classify the data points using a hyperplane in N-dimensional space (N is the number of features). In the SVM model, the sewer pipe condition status is determined by maximizing the distance from the hyperplane to the data points of both good and bad conditions. The hyperplanes can be computed as follows (Zendehboudi et al., 2018):

$$\begin{cases} y_i(\boldsymbol{w}.\emptyset^T(x_i) + \boldsymbol{b}) \geq 1 - \varepsilon_i \\ \varepsilon_i \geq 0, \qquad \forall i = 1,2,\dots,n \end{cases} \tag{2}$$

where n is the number of inspected pipes, $x_i \in \mathbb{R}^n, y_i \in \mathbb{R}$, x, and y are vectors that contain input factors and sewer's condition status respectively, $\boldsymbol{w}$ is the coefficient vector, $\boldsymbol{b}$ is and bias of the hyperplane in the feature space, $\emptyset$ is the non-linear mapping function, and ε_i are positive slack variables. The predicted condition status of the sewer pipe using the SVM is calculated as follows (Cervantes et al., 2020):

$$\begin{cases} f(x) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right) \\ \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \ \ \forall i = 1,2,\dots,n \end{cases} \tag{3}$$

where auxiliary variables α_i are Lagrange multipliers, C is the regularization parameter, and $K(x, x_i)$ is Kernel function which can be linear SVM (LN-SVM), (polynomial SVM (PL-SVM), radial basis function SVM (RBF-SVM), or sigmoid SVM (SIG-SVM) according to the following equations (Suykens & Vandewalle, 1999; Tien Bui et al., 2012):

$$\begin{cases} K_{\text{LN-SVM}}(x, x_i) = x^T x_i \\ K_{\text{PL-SVM}}(x, x_i) = (\gamma x^T x_i + 1)^d \\ K_{\text{RBF-SVM}}(x, x_i) = e^{-\gamma \|x - x_i\|^2} \\ K_{\text{SIG-SVM}}(x, x_i) = \tanh(\gamma x^T x_i + r) \end{cases} \quad (4)$$

where d is the degree of polynomial kernel function ($d > 0$), γ is the kernel width, and r is the coefficient in kernel projection. Optimizing the parameters $C, \gamma, d,$ and r will lead to optimizing the kernel function and therefore improve the predicted SVM performance. These parameters can be determined by using the cross-validation technique to combine with the grid-search method (Liu & Yang, 2013; Tien Bui et al., 2012).

c. Random Forest for Classification

A Decision Tree (DT) regression creates regression models in the form of a tree structure in which the sewer training dataset is split into smaller and smaller subsets while at the same time an associated decision tree is developed. The decision tree consists of four basic components: root, internal nodes, leaf nodes, and branches. The root node contains all the factors, an internal node can contain two or more branches that are associated with a decision function, and the leaf node indicates the sewer damage score. A decision tree can be constructed via several steps (Syachrani et al., 2013): (1) assigning all observations in the root node; (2) splitting the root node into branches based on the predicted sewer damage score using the decision function; (3) distributing observations on the higher node to the lower nodes; and (4) repeating the process until all sewer pipes have been processed.

Classification And Regression Tree (CART) was first proposed by Breiman et al. (1984) to solve regression and classification problems based on tree-based structures. In this method, the sewer dataset (also called the root node) was divided into binary (good or bad) values at each node using a series of recursive binary splits based on evaluating every possible predictor (Ebrahimy et al., 2020). Finally, the predicted sewer's status was defined based on the most commonly occurring class of the node. The decision trees created by CART have two branches for each decision node. Difference from the decision tree for classification, which uses Gini Impurity or Entropy values as criteria for splitting root/decision nodes, the "goodness" criterion is applied in the CART algorithm to split root/decision nodes and is computed as follows (Larose & Larose, 2014):

$$f(s|t) = 2P_L P_R \sum_{i=1}^{n} |P(i|t_L) - P(i|t_R)| \quad (5)$$

where n is the number of sewer inspections, $f(s|t)$ is a measure of "goodness of fit", t_L and t_R are the left and right children of a candidate split s at node t, respectively, P_L and P_R are the proportions of records at t_L and t_R, respectively, $P(i|t_L)$ and $P(i|t_R)$ are the proportions of class i at t_L and t_R, respectively.

Random Forest for classification (RFC) was developed by Breiman (2001) to significantly improve classification accuracy by creating an ensemble of trees and letting them vote for the most popular class. In the RFC model, the sewer input dataset was randomly split into classification trees, and the model was trained through bagging or bootstrap aggregating. The final sewer's condition status was obtained by aggregating the prediction from each tree. The RFC model was applied for this case using the bootstrap technique to control the sub-sample size and get the average prediction from sub-decision trees to improve the predictive accuracy and control over-fitting.

d. Model Validation

In this study, the efficiency of the developed models was assessed using Geometric Mean (GM), Accuracy (ACC), F-Score, and Matthew's correlation coefficient (MCC).

$$GM = \sqrt{\frac{TP}{TP + FN} + \frac{TN}{TN + FP}} \tag{6}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$F - Score = \frac{2TP}{2TP + FP + FN} \tag{8}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{9}$$

Whereas ACC is the most popular criterion for assessing the classification performance of ML algorithms, GM, F-Score, and MCC are sensitive to imbalanced datasets (Tharwat, 2021). Other values including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are obtained from the confusion matrix for binary classification (**Figure 3-1**).

		Predicted Class	
		Good condition	Bad condition
Actual Class	Good condition	True Positive (TP)	False Negative (FN)
	Bad condition	False Positive (FP)	True Negative (TN)

Figure 3-1. Confusion matrix for binary classification

3.2. Regression-based Machine Learning Algorithms

a. Multi-Layer Perceptron Neural Network

A Multi-layer Perceptron Neural Network (MLP) is a fully connected class of feedforward artificial neural networks. This architecture normally consists of three or more layers (i.e., an input layer, an output layer, and one or more hidden layers) and each layer contains different neurons. In general, the number of neurons in the input layer is equal to the number of input factors, the number of neurons in the output layer is equal to one for the regression problem, and the number of hidden layers and hidden neurons fluctuates depending on the complexity of the MLP architecture. Determining the number of hidden layers and hidden neurons is generally implemented using a trial-and-error approach (Orhan et al., 2011).

In this tutorial, a single-layer MLP architecture was used. The number of hidden neurons, various activation functions in the hidden layer, and several optimization solvers were tuned using the Scikit-learn ML library. The early-stopping technique was used to avoid overfitting while training the model.

b. Support Vector Regression

A Support Vector Regression (SVR) is one type of Support Vector Machine used for regression problems. This algorithm creates and finds the best-fit hyperplane in n-dimensional space that is close to as many of the data points as possible (Trafalis & Ince, 2000). For regression problems, the linear form of the hyperplane can be computed as follows (Wauters & Vanhoucke, 2014):

$$f(x) = wx + b \tag{10}$$

where $f(x)$ is the predicted value, x is the input vector of the data point, w and b are the slope and intercept.

The goal function of the SVR model can be defined as follows (Smola & Schölkopf, 2004):

$$\begin{cases} \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x) + b \\ subject\ to \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0, \qquad \alpha_i, \alpha_i^* \in [0, C] \end{cases} \tag{11}$$

where $f(x)$ is the predicted value, n is the number of sewer inspections, x is the input vector of the data point, w and b are the slope and intercept, respectively, α_i, α_i^* are Lagrange multipliers, the constant $C > 0$ is the trade-off between the flatness of the $f(x)$ and the amount up to which deviations larger than the insensitive loss function, $K(x_i, x)$ is kernel function (e.g., linear function, polynomial function, radial basis function, or sigmoid function).

c. Random Forest for Regression

Random Forest for regression (RFR) is an ensemble learning method that uses multiple decision trees as base learning models for regression problems. The bagging (or bootstrap aggregation) algorithm is generally used to create the RFR model. In this way, each decision tree in the RFR is created from different samples at each node and produces an individual prediction. This model generates hundreds or thousands of regression decision trees and the average sewer status predicted from the individual trees is calculated for the final result (Kumar & Shaikh, 2017). As a result, the RFR model generally has higher performance compared to the DT because it can avoid the correlation of different trees and the final results are obtained from the diversity of the trees (Li et al., 2018).

d. Model Validation

The constructed ML models were compared to select the best model for the sewer condition prediction. In this tutorial, the predictive performance of the regression-based machine learning models was assessed using the coefficient of determination (R^2), mean absolute error (MAE), and root mean square error ($RMSE$) expressed as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i^{act} - y_i^{pred}\right)^2}{\sum_{i=1}^{n}(y_i^{act} - \bar{y})^2} \tag{12}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{act} - y_i^{pred}\right| \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{act} - y_i^{pred}\right)^2} \tag{14}$$

where n is the total number of measurements, $\bar{y}$ is the mean value of the actual measurements, y_i^{act} and y_i^{pred} are the i^{th} actual and predicted measurements.

3.3. Comparison of Machine Learning Algorithms

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) was applied to rank the machine learning models (Vazquezl et al., 2021). This method is a common approach for ranking machine learning algorithms, using multiple criteria on a single dataset by choosing the alternatives that have the shortest distance to the positive-ideal solution and the longest distance to the negative-ideal solution (Behzadian et al., 2012). These distances relate to the alternative weights that are used to compute the overall performance score (Chakraborty, 2022). Interested readers can find more detailed information on the TOPSIS in Behzadian et al. (2012). In this tutorial, the package "TOPSIS" in R was used to implement the TOPSIS method (Ihaka & Gentleman, 1996).

4. Machine Learning Model Implementation

4.1. Importing needed libraries

```
import os
import numpy as np
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.svm import SVR, SVC
from sklearn.neural_network import MLPRegressor, MLPClassifier
from imblearn.metrics import geometric_mean_score
from sklearn.metrics import f1_score, matthews_corrcoef, accuracy_score
```

4.2. Importing dataset and eliminating unnecessary components

```
folder =
r'C:\Users\vanln\OneDrive\Documents\Lam_Data\Tutorial\QGIS_Condition_Asses
sment_Tutorial\Data'
data_path = os.path.join(folder, 'Data.csv')
data = pd.read_csv(data_path)
column_drop = ['PipeID']
data = data.drop(columns=column_drop)
```

4.3. Defining input and output vectors

```
X = data.iloc[:,:-2]
y_reg = data.iloc[:,-2:-1]
y_clas = data.iloc[:,-1:]
```

4.4. Defining categorical columns

```
lb_column = ['PipeType','NetworkTyp','Connection','PipeForm','Material',
             'Geology','SoilType','Building','LandCover','RoadClass']
```

4.5. Machine Learning Implementation for Classification

The hyperparameters used for tuning the machine learning models are shown in **Table 4-1**.

Table 4-1. The hyperparameters used for tuning models

Model	Parameter	Range
MLP	Activation function	*logistic, tanh, and relu*
	Solver	*lbfgs, sgd, and adam*
	Number of hidden neurons	1,2, …,199,200
SVM/SVR	Kernel function	*linear, rbf, poly, and sigmoid*
	C	$2^{-15}, 2^{-14}, \ldots, 2^{4}, 2^{5}$
	d	1,2, …,9,10
	γ	$2^{-10}, 2^{-9}, \ldots, 2^{2}, 2^{3}$
RFC/RFR	Quality of the split	*gini and entropy*
	Number of features	1,2, …,19,20
	Number of trees	10,20, …,990,1000

Abbreviations: *relu* - rectified linear unit; *tanh* - hyperbolic tangent activation function; *lbfgs* – Limited-memory Broyden–Fletcher–Goldfarb–Shanno; *sgd* - stochastic gradient descent; *adam* - adaptive moment estimation.

The steps for implementing machine learning for classification are presented as follows:

- *Step 1:* Split the dataset into training and validation subsets.

```
x_train_clas, x_test_clas, y_train_clas, y_test_clas = train_test_split(X,
y_clas, test_size=0.2, random_state=42, stratify=y_clas)
```

- *Step 2:* Scale data into the range [0,1] and encode categorical variables.

```
ordinalEncoder_clas = OrdinalEncoder()
scaler_clas = MinMaxScaler(feature_range=(0, 1))
x_train_clas[lb_column] =
ordinalEncoder_clas.fit_transform(x_train_clas[lb_column])
x_test_clas[lb_column] =
ordinalEncoder_clas.fit_transform(x_test_clas[lb_column])
scaler_clas.fit(x_train_clas)
X_train_clas = scaler_clas.transform(x_train_clas)
X_test_clas = scaler_clas.transform(x_test_clas)
```

- *Step 3:* Create a new dataframe to store these assessment criteria such as GM, ACC, F-Score, and MCC values.

```
index_clas = ['GM','ACC','F-Score','MCC']
df_clas_train = pd.DataFrame(index=index_clas)
df_clas_test = pd.DataFrame(index=index_clas)
```

a. *Turn hyperparameters of the MLP model, fit the model, and calculate the assessment criteria using the training and validation datasets.*

✓ Tune hyperparameters of the MLP model

```
MLP_model = MLPClassifier(early_stopping=True, random_state=42)
parameters = {'activation':['logistic', 'tanh', 'relu'],
          'solver':['lbfgs', 'sgd', 'adam'],
          'hidden_layer_sizes': tuple(np.arange(1,201,1))}
MLP_search = GridSearchCV(MLP_model, parameters, scoring='accuracy',
n_jobs=-1)
MLP_result = MLP_search.fit(X_train_clas, y_train_clas)
```

✓ Define the best hyperparameters

```
MLP_result.best_params_: {'activation': 'logistic', 'hidden_layer_sizes':
51, 'solver': 'adam'}
```

✓ Fit the model

```
MLP_model = MLPClassifier(early_stopping=True, random_state=42,
                          activation='logistic', solver='adam',
hidden_layer_sizes=51)
MLP_result = MLP_model.fit(X_train_clas, y_train_clas)
```

✓ Predict, calculate, and store the GM, ACC, F-Score, and MCC values in the dataframe

```
# On the training dataset
rs_MLP = []
Y_pred = MLP_result.predict(X_train_clas)
rs_MLP.append(geometric_mean_score(y_true=y_train_clas, y_pred=Y_pred,
average='weighted'))
rs_MLP.append(accuracy_score(y_true=y_train_clas, y_pred=Y_pred))
rs_MLP.append(f1_score(y_true=y_train_clas, y_pred=Y_pred,
average='weighted'))
rs_MLP.append(matthews_corrcoef(y_true=y_train_clas, y_pred=Y_pred))
df_clas_train['MLP'] = rs_MLP

# On the validation dataset
rs_MLP = []
Y_pred = MLP_result.predict(X_test_clas)
rs_MLP.append(geometric_mean_score(y_true=y_test_clas, y_pred=Y_pred,
average='weighted'))
rs_MLP.append(accuracy_score(y_true=y_test_clas, y_pred=Y_pred))
rs_MLP.append(f1_score(y_true=y_test_clas, y_pred=Y_pred,
average='weighted'))
rs_MLP.append(matthews_corrcoef(y_true=y_test_clas, y_pred=Y_pred))
df_clas_test['MLP'] = rs_MLP
```

b. *Turn hyperparameters of the SVM model, fit the model, and calculate the assessment criteria using the training and validation datasets.*

✓ Tune hyperparameters of the SVM model

```
SVM_model = SVC(random_state=42)
parameters = {'kernel':['linear','rbf', 'poly','sigmoid'],
              'degree': np.arange(1,11,1), 'gamma': 2.0**np.arange(-10, 3,
7), 'C': 2.0**np.arange(-15,4,1)}
```

```
SVM_search = GridSearchCV(SVM_model, parameters, scoring='accuracy',
n_jobs=-1)
SVM_result = SVM_search.fit(X_train_clas, y_train_clas)
```

✓ Define the best hyperparameters

```
SVM_result.best_params_: {'C': 4.0, 'degree': 3, 'gamma': 0.125, 'kernel':
'poly'}
```

✓ Fit the model

```
SVM_model = SVC(random_state=42, kernel='poly', C=pow(2,2), degree=3,
gamma=pow(2,-3), probability=True)
SVM_result = SVM_model.fit(X_train_clas, y_train_clas)
```

✓ Predict, calculate, and store the GM, ACC, F-Score, and MCC values in the dataframe

```
# On the training dataset
rs_SVM = []
Y_pred = SVM_result.predict(X_train_clas)
rs_SVM.append(geometric_mean_score(y_true=y_train_clas, y_pred=Y_pred,
average='weighted'))
rs_SVM.append(accuracy_score(y_true=y_train_clas, y_pred=Y_pred))
rs_SVM.append(f1_score(y_true=y_train_clas, y_pred=Y_pred,
average='weighted'))
rs_SVM.append(matthews_corrcoef(y_true=y_train_clas, y_pred=Y_pred))
df_clas_train['SVM'] = rs_SVM

# On the validation dataset
rs_SVM = []
Y_pred = SVM_result.predict(X_test_clas)
rs_SVM.append(geometric_mean_score(y_true=y_test_clas, y_pred=Y_pred,
average='weighted'))
rs_SVM.append(accuracy_score(y_true=y_test_clas, y_pred=Y_pred))
rs_SVM.append(f1_score(y_true=y_test_clas, y_pred=Y_pred,
average='weighted'))
rs_SVM.append(matthews_corrcoef(y_true=y_test_clas, y_pred=Y_pred))
df_clas_test['SVM'] = rs_SVM
```

c. *Turn hyperparameters of the RFC model, fit the model, and calculate the assessment criteria using the training and validation datasets.*

✓ Tune hyperparameters of the RFC model

```
RF_model = RandomForestClassifier(random_state=42)
parameters = {'max_features':np.arange(1,X_train_clas.shape[1]+1,1),
              'criterion':['gini','entropy'],
              'n_estimators':np.arange(10,1010,10)}
RF_search = GridSearchCV(RF_model, parameters, scoring='accuracy',
n_jobs=-1)
RF_result = RF_search.fit(X_train_clas, y_train_clas)
```

✓ Define the best hyperparameters

```
RF_result.best_params_: {'criterion': 'gini', 'max_features': 1,
'n_estimators': 240}
```

✓ Fit the model

```
RF_model = RandomForestClassifier(random_state=42, criterion="gini",
                                  n_estimators=240, max_features=1)
RF_result = RF_model.fit(X_train_clas, y_train_clas)
```

✓ Predict, calculate, and store the GM, ACC, F-Score, and MCC values in the dataframe

```
# On the training dataset
rs_RF = []
Y_pred = RF_result.predict(X_train_clas)
rs_RF.append(geometric_mean_score(y_true=y_train_clas, y_pred=Y_pred,
average='weighted'))
rs_RF.append(accuracy_score(y_true=y_train_clas, y_pred=Y_pred))
rs_RF.append(f1_score(y_true=y_train_clas, y_pred=Y_pred,
average='weighted'))
rs_RF.append(matthews_corrcoef(y_true=y_train_clas, y_pred=Y_pred))
df_clas_train['RF'] = rs_RF

# On the validation dataset
rs_RF = []
Y_pred = RF_result.predict(X_test_clas)
rs_RF.append(geometric_mean_score(y_true=y_test_clas, y_pred=Y_pred,
average='weighted'))
rs_RF.append(accuracy_score(y_true=y_test_clas, y_pred=Y_pred))
rs_RF.append(f1_score(y_true=y_test_clas, y_pred=Y_pred,
average='weighted'))
rs_RF.append(matthews_corrcoef(y_true=y_test_clas, y_pred=Y_pred))
df_clas_test['RF'] = rs_RF
```

4.6. Machine Learning Implementation for Regression

The steps for implementing machine learning for regression are presented as follows:

➢ *Step 1:* Split the dataset into training and validation subsets.

```
x_train_reg, x_test_reg, y_train_reg, y_test_reg = train_test_split(X,
y_reg, test_size=0.2, random_state=42)
```

➢ *Step 2:* Scale data into the range [0,1] and encode categorical variables.

```
ordinalEncoder_reg = OrdinalEncoder()
scaler_reg = MinMaxScaler(feature_range=(0, 1))
x_train_reg[lb_column] =
ordinalEncoder_reg.fit_transform(x_train_reg[lb_column])
x_test_reg[lb_column] =
ordinalEncoder_reg.fit_transform(x_test_reg[lb_column])
scaler_reg.fit(x_train_reg)
X_train_reg = scaler_reg.transform(x_train_reg)
X_test_reg = scaler_reg.transform(x_test_reg)
```

➢ *Step 3:* Create a new dataframe to store these assessment criteria such as R^2, MAE, and RMSE values.

```
index_reg = ['R2','MAE','RMSE']
df_reg_train = pd.DataFrame(index=index_reg)
df_reg_test = pd.DataFrame(index=index_reg)
```

a. *Turn hyperparameters of the MLP model, fit the model, and calculate the assessment criteria using the training and validation datasets.*

✓ Tune hyperparameters of the MLP model

```
MLP_model = MLPRegressor(early_stopping=True, random_state=42,
max_iter=500)
parameters = {'activation':['logistic', 'tanh', 'relu'],'solver':['lbfgs',
'sgd', 'adam']
           ,'hidden_layer_sizes': tuple(np.arange(1,201,1))}
MLP_search = GridSearchCV(MLP_model, parameters, scoring='r2', n_jobs=-1)
MLP_result = MLP_search.fit(X_train_reg, y_train_reg)
```

✓ Define the best hyperparameters

```
MLP_result.best_params_: {'activation': 'tanh', 'hidden_layer_sizes': 71,
'solver': 'adam'}
```

✓ Fit the model

```
MLP_model = MLPRegressor(early_stopping=True, random_state=42,
max_iter=500, activation='tanh', hidden_layer_sizes=71, solver='adam')
MLP_result = MLP_model.fit(X_train_reg, y_train_reg)
```

✓ Predict, calculate, and store the R^2, MAE, and RMSE values in the dataframe

```
# On the training dataset
rs_MLP = []
Y_pred = MLP_result.predict(X_train_reg)
rs_MLP.append(r2_score(y_true=y_train_reg, y_pred=Y_pred))
rs_MLP.append(mean_absolute_error(y_true=y_train_reg, y_pred=Y_pred))
rs_MLP.append(np.sqrt(mean_squared_error(y_true=y_train_reg,
y_pred=Y_pred)))
df_reg_train['MLP'] = rs_MLP

# On the validation dataset
rs_MLP = []
Y_pred = MLP_result.predict(X_test_reg)
rs_MLP.append(r2_score(y_true=y_test_reg, y_pred=Y_pred))
rs_MLP.append(mean_absolute_error(y_true=y_test_reg, y_pred=Y_pred))
rs_MLP.append(np.sqrt(mean_squared_error(y_true=y_test_reg,
y_pred=Y_pred)))
df_reg_test['MLP'] = rs_MLP
```

b. *Turn hyperparameters of the SVR model, fit the model, and calculate the assessment criteria using the training and validation datasets.*

✓ Tune hyperparameters of the SVR model

```
SVR_model = SVR()
parameters = {'kernel':['rbf'], 'gamma': 2.0**np.arange(-15, 5, 1),
              'C': 2.0**np.arange(-5,15,1)}
SVR_search = GridSearchCV(SVR_model, parameters, scoring='r2', n_jobs=-1,
return_train_score=True)
SVR_result = SVR_search.fit(X_train_reg, y_train_reg)
```

✓ Define the best hyperparameters

```
SVR_result.best_params_: {'C': 128.0, 'gamma': 4.0, 'kernel': 'rbf'}
```

✓ Fit the model

```
SVR_model = SVR(C=128, gamma=4, kernel='rbf')
SVR_result = SVR_model.fit(X_train_reg, y_train_reg)
```

✓ Predict, calculate, and store the GM, ACC, F-Score, and MCC values in the dataframe

```
# On the training dataset
rs_SVR = []
Y_pred = SVR_result.predict(X_train_reg)
rs_SVR.append(r2_score(y_true=y_train_reg, y_pred=Y_pred))
rs_SVR.append(mean_absolute_error(y_true=y_train_reg, y_pred=Y_pred))
rs_SVR.append(np.sqrt(mean_squared_error(y_true=y_train_reg,
y_pred=Y_pred)))
df_reg_train['SVR'] = rs_SVR

# On the validation dataset
rs_SVR = []
Y_pred = SVR_result.predict(X_test_reg)
rs_SVR.append(r2_score(y_true=y_test_reg, y_pred=Y_pred))
rs_SVR.append(mean_absolute_error(y_true=y_test_reg, y_pred=Y_pred))
rs_SVR.append(np.sqrt(mean_squared_error(y_true=y_test_reg,
y_pred=Y_pred)))
df_reg_test['SVR'] = rs_SVR
```

c. *Turn hyperparameters of the RFR model, fit the model, and calculate the assessment criteria using the training and validation datasets.*

✓ Tune hyperparameters of the RFR model

```
RFR_model = RandomForestRegressor(random_state=42)
parameters = {'max_features': np.arange(1,X_train_reg.shape[1]+1,1),
              'n_estimators':np.arange(1,101,1)}
RFR_search = GridSearchCV(RFR_model, parameters, scoring='r2', n_jobs=-1)
RFR_result = RFR_search.fit(X_train_reg, y_train_reg)
```

✓ Define the best hyperparameters

```
RFR_result.best_params_: {'max_features': 3, 'n_estimators': 30}
```

✓ Fit the model

```
RFR_model = RandomForestRegressor(random_state=42,
n_estimators=30,max_features=3)
RFR_result = RFR_model.fit(X_train_reg, y_train_reg)
```

✓ Predict, calculate, and store the GM, ACC, F-Score, and MCC values in the dataframe

```
# On the training dataset
rs_RFR = []
Y_pred = RFR_result.predict(X_train_reg)
rs_RFR.append(r2_score(y_true=y_train_reg, y_pred=Y_pred))
```

```
rs_RFR.append(mean_absolute_error(y_true=y_train_reg, y_pred=Y_pred))
rs_RFR.append(np.sqrt(mean_squared_error(y_true=y_train_reg,
y_pred=Y_pred)))
df_reg_train['RFR'] = rs_RFR

# On the validation dataset
rs_RFR = []
Y_pred = RFR_result.predict(X_test_reg)
rs_RFR.append(r2_score(y_true=y_test_reg, y_pred=Y_pred))
rs_RFR.append(mean_absolute_error(y_true=y_test_reg, y_pred=Y_pred))
rs_RFR.append(np.sqrt(mean_squared_error(y_true=y_test_reg,
y_pred=Y_pred)))
df_reg_test['RFR'] = rs_RFR
```

5. Result Comparision

5.1. Classification Models

The performance of machine learning models for classification problems using the training and validation datasets is shown in **Table 5-3**.

Table 5-1. Performance comparison of classification-based machine learning models

Model	Assessment Criteria							
	Training dataset				Validation dataset			
	GM	ACC	F-SCORE	MCC	GM	ACC	F-SCORE	MCC
MLP	0.65	0.61	0.53	0.34	0.56	0.41	0.34	0.26
SVM	0.72	0.66	0.59	0.45	0.49	0.32	0.21	0.14
RFC	1.00	1.00	1.00	1.00	0.70	0.63	0.57	0.41

5.2. Regression Models

The performance of machine learning models for classification problems using the training and validation datasets is shown in **Table 5-2**.

Table 5-2. Performance comparison of regression-based machine learning models

Model	Assessment Criteria					
	Training dataset			Validation dataset		
	R^2	MAE	RMSE	R^2	MAE	RMSE
MLP	-0.02	76.03	278.31	-0.09	101.04	235.20
SVM	0.07	42.01	265.57	0.02	97.06	223.88
RFR	0.85	35.64	108.38	-0.11	123.17	237.73

5.3. TOPSIS Implementation

The structure of the assessment matrix of the classification problem for running the package

"TOPSIS" in R is shown in **Figure 5-1**.

	A	B	C	D	E
1	Model	GM	ACC	F-Score	MCC
2	MLP	0.56	0.41	0.34	0.26
3	SVM	0.49	0.32	0.21	0.14
4	RF	0.7	0.63	0.57	0.41

Figure 5-1. Assessment matrix of the classification problem

The codes for running the "TOPSIS" in R are shown below:

```
library(topsis)
path = 'C:/Lam_Data/matrix.csv'
dat = read.csv(path,T)
attach(dat)
# Remove the column 'Model'
drop = c('Model')
dat_new = dat[,!(names(dat) %in% drop)]
mt = as.matrix(dat_new)
cot = dat['Model']
w = rep(1,length(dat_new))
# Maximize GM, ACC, F-Score, and MCC
i = c('+','+','+','+')
result = topsis(mt, w, i)
# Create a new dataframe to store the result
cot_new = cbind(cot, result$score, result$rank)
```

The rank of classification-based machine learning models using the validation dataset is shown in **Table 5-3**.

Table 5-3. The rank of classification-based machine learning models

Model	Score	Rank
RFC	1.0	1
MLP	0.3796732	2
SVM	0.0	3

The rank of regression-based machine learning models using the validation dataset is shown in **Table 5-4**.

Table 5-4. The rank of regression-based machine learning models

Model	Score	Rank
SVM	1.0	1
MLP	0.1926815	2
RFR	0.0	3

5.4. Discussion

- The negative R^2 values indicate that the chosen models (MLP and RFR) do not follow the trend of the data, so fit worse than a horizontal line.

- The predictive performance significantly fluctuates depending on the characteristics of output (regression or classification problems). For the classification problem, the RFC is better than the others (**Table 5-3**). Otherwise, the SVM is more suitable than the MLP and RFR (**Table 5-4**).
- Variables (including physical, environmental, and operational factors) can be added to the dataset and the models can be retrained.
- With the same dataset, developed machine learning models for classification produced better results compared to regression models.
- Training Random Forest (including RFC and RFR) model takes time longer than the MLP and Support Vector Machine models.

6. Sewer Condition Map

The condition of all sewer pipes in the study area can be predicted using the input data in **Figure 2-93** and constructed machine learning model. In this tutorial, we use the RFC model to predict the condition of all sewer pipes. The steps for predicting and visualizing the results are shown as follows:

- *Step 1:* Run the RFR model using input data in **Figure 2-93** based on the steps presented in subsection 4.6.
- *Step 2:* Check the status of all sewer pipes (**Figure 6-1**).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	PipeID	Age	Diameter	Depth	Slope	Length	PipeType	NetworkT	Connectic	PipeForm	Material	Rainfall	Geology	GroundW	SoilType	Building	LandCove	RoadClass	TrafficVol	Population	Prediction
2	976	22	160	-1.26	8.73	42.75	Stormwat	Main	Others	S	PVC	148.04	Geology_	9.96	Soiltype_	No	LandUse_	RoadClass	0	23.57	Very good
3	981	23	225	-1.57	-2.8	46.93	Combinec	Main	MUF	S	BET	150.09	Geology_	11.28	Soiltype_	No	LandUse_	RoadClass	0	68.66	Very good
4	1187	46	160	-1.71	4.91	25.06	Stormwat	Main	MUF	S	PVC	147.05	Geology_	15.13	Soiltype_	No	LandUse_	RoadClass	0	71.74	Very good
5	1221	60	225	-2.1	0.16	11.52	Wastewat	Main	MUF	S	BET	151.22	Geology_	19.02	Soiltype_	No	LandUse_	RoadClass	0	63.54	Very good
6	1222	62	300	-2.01	-0.14	66.62	Combinec	Main	MUF	S	BET	151.53	Geology_	22.36	Soiltype_	No	LandUse_	RoadClass	0	63.54	Very good
7	1229	70	125	-2.04	2.27	69.32	Wastewat	Main	MUF	S	BET	151.54	Geology_	26.18	Soiltype_	Yes	LandUse_	RoadClass	0	197.79	Very bad
8	1230	60	225	-1.5	2.1	55.34	Wastewat	Main	MUF	S	BET	152.64	Geology_	26.51	Soiltype_	Yes	LandUse_	RoadClass	0	89.16	Very good
9	1231	60	150	-1.99	1.91	2.85	Stormwat	Main	MUF	S	BET	151.46	Geology_	20.04	Soiltype_	No	LandUse_	RoadClass	0	69.69	Very good
10	1272	70	150	-2.25	1.9	18.63	Wastewat	Main	MUF	S	BET	150.79	Geology_	23.54	Soiltype_	No	LandUse_	RoadClass	0	248	Very good
11	1280	70	150	-2.17	3.7	16.15	Wastewat	Main	MUF	S	BET	151.27	Geology_	24	Soiltype_	No	LandUse_	RoadClass	0	236.73	Very good
12	1294	48	225	-1.71	2.26	56.39	Wastewat	Main	MUF	S	BET	152.85	Geology_	25.75	Soiltype_	No	LandUse_	RoadClass	0	171.14	Very bad
13	1328	57	200	-1.93	0.26	34.56	Stormwat	Main	MUF	S	BET	151.64	Geology_	24.36	Soiltype_	No	LandUse_	RoadClass	0	236.73	Very good
14	1330	57	200	-1.98	2.21	2.1	Stormwat	Main	MUF	S	BET	151.78	Geology_	24.82	Soiltype_	No	LandUse_	RoadClass	0	236.73	Very good
15	1331	60	225	-2.23	1.69	10.91	Combinec	Main	MUF	S	BET	152.54	Geology_	27.47	Soiltype_	No	LandUse_	RoadClass	0	89.16	Very good
16	1341	37	300	-1.91	0.64	26.25	Combinec	Main	MUF	S	BET	147.38	Geology_	13.99	Soiltype_	No	LandUse_	RoadClass	6800	71.74	Very bad
17	1345	27	160	-1.45	11.33	15.98	Wastewat	Main	Others	S	PVC	147.33	Geology_	14.09	Soiltype_	No	LandUse_	RoadClass	0	71.74	Very good
18	1346	27	160	-2.86	8.96	29.82	Wastewat	Main	Others	S	PVC	147.41	Geology_	14.26	Soiltype_	No	LandUse_	RoadClass	0	71.74	Very good
19	1354	26	200	-2.72	1.8	62.02	Wastewat	Main	Others	S	PVC	153.29	Geology_	19.73	Soiltype_	No	LandUse_	RoadClass	0	0	Very good

Figure 6-1. Condition of sewer pipes in the study area

- *Step 3:* Join the results in the above step with the pipe layer using the unique index in QGIS (**Figure 6-2**).

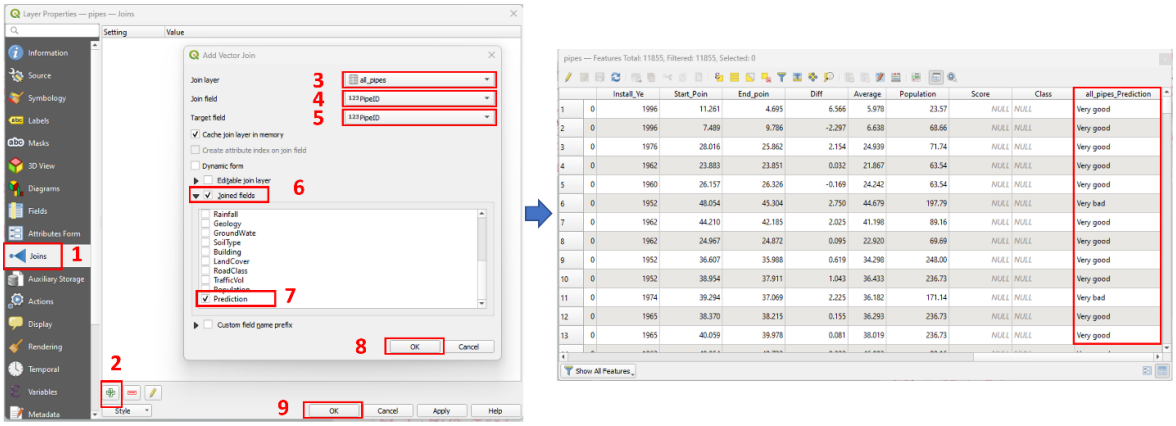


Figure 6-2. Joining the predictive condition with the corresponding pipe

➢ *Step 4:* Create a new column, called *"Condition"* to save the condition of sewer pipes (**Figure 6-3**).

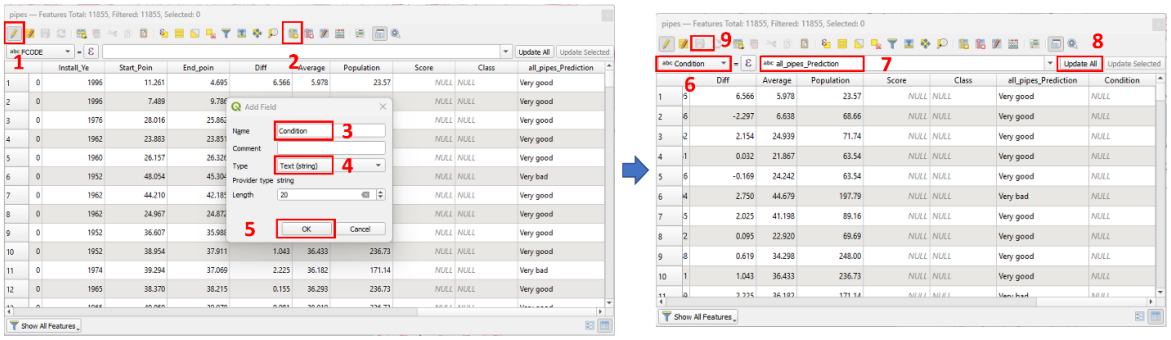


Figure 6-3. Assigning condition for sewer pipes

➢ *Step 5:* Visualize the condition of sewer pipes on the map (**Figure 6-4**).

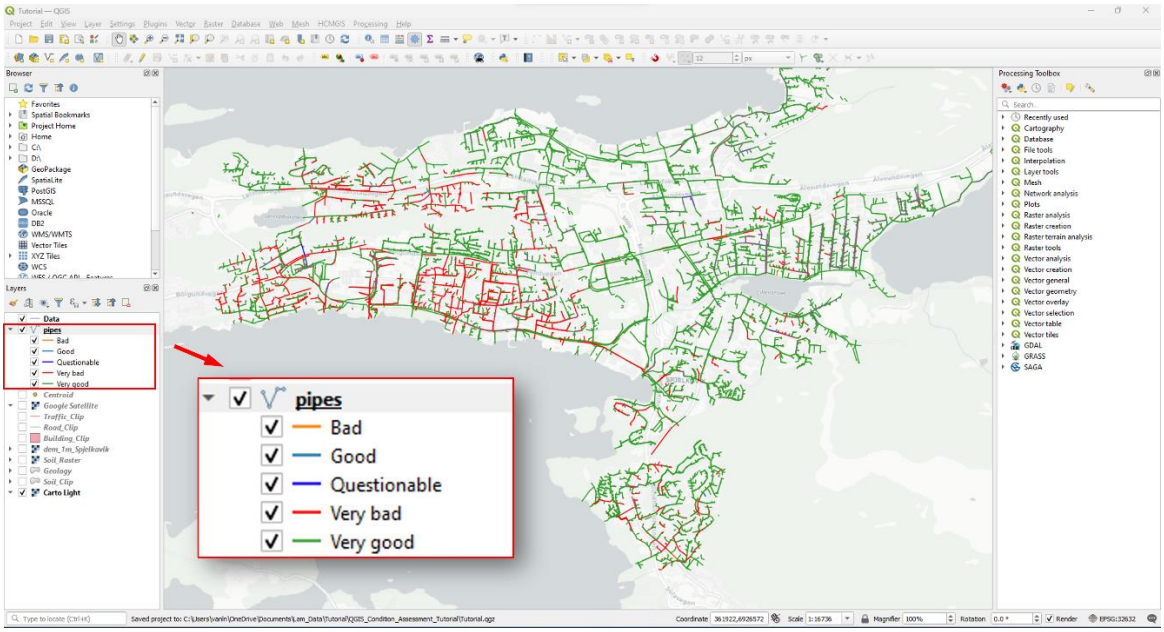


Figure 6-4. Condition of sewer pipes in 2022

References

Ahmadi, M., Cherqui, F., De Massiac, J.-C., & Le Gauffre, P. (2014). Influence of available data on sewer inspection program efficiency. *Urban Water Journal*, *11*(8), 641-656. https://doi.org/10.1080/1573062X.2013.831910

Beheshti, M., Sægrov, S., & Ugarelli, R. (2015). Infiltration/inflow assessment and detection in urban sewer system.

Behzadian, M., Khanmohammadi Otaghsara, S., Yazdani, M., & Ignatius, J. (2012). A state-of the-art survey of TOPSIS applications. *Expert Systems with Applications*, *39*(17), 13051-13069. https://doi.org/10.1016/j.eswa.2012.05.056

Breiman, L. (2001). Random Forests. *Machine learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Routledge.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189-215. https://doi.org/10.1016/j.neucom.2019.10.118

Chakraborty, S. (2022). TOPSIS and Modified TOPSIS: A comparative analysis. *Decision Analytics Journal*, *2*, 100021. https://doi.org/10.1016/j.dajour.2021.100021

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

de Oliveira, L. M., Maillard, P., & de Andrade Pinto, E. J. (2017). Application of a land cover pollution index to model non-point pollution sources in a Brazilian watershed. *CATENA*, *150*, 124-132. https://doi.org/10.1016/j.catena.2016.11.015

Ebrahimy, H., Feizizadeh, B., Salmani, S., & Azadi, H. (2020). A comparative study of land subsidence susceptibility mapping of Tasuj plane, Iran, using boosted regression tree, random forest and classification and regression tree methods. *Environmental Earth Sciences*, *79*(10), 223. https://doi.org/10.1007/s12665-020-08953-0

Haugen, H. J., & Viak, A. (2018). *Datafl yt – Klassifi sering av avløpsledninger*. https://docplayer.me/211256711-Norsk-vann-rapport-dataflyt-klassifisering-av-avlopsledninger.html

Hawari, A., Alkadour, F., Elmasry, M., & Zayed, T. (2020). A state of the art review on condition assessment models developed for sewer pipelines. *Engineering Applications of Artificial Intelligence*, *93*, 103721. https://doi.org/10.1016/j.engappai.2020.103721

Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299-314. https://doi.org/10.1080/10618600.1996.10474713

Kumar, S. S., & Shaikh, T. (2017, 6-7 Sept. 2017). Empirical Evaluation of the Performance of Feature Selection Approaches on Random Forest. 2017 International Conference on Computer and Applications (ICCA),

Kwak, T. Y., Woo, S. I., Chung, C. K., & Kim, J. (2020). Experimental assessment of the relationship between rainfall intensity and sinkholes caused by damaged sewer pipes. *Nat. Hazards Earth Syst. Sci.*, *20*(12), 3343-3359. https://doi.org/10.5194/nhess-20-3343-2020

Laakso, T., Kokkonen, T., Mellin, I., & Vahala, R. (2018). Sewer Condition Prediction and Analysis of Explanatory Factors. *Water 2018, Vol. 10, Page 1239*, *10*(9), 1239-1239. https://doi.org/10.3390/W10091239

Larose, D. T., & Larose, C. D. (2014). *Discovering knowledge in data: an introduction to data mining* (Vol. 4). John Wiley & Sons.

Li, Y., Zou, C., Berecibar, M., Nanini-Maury, E., Chan, J. C. W., van den Bossche, P., Van Mierlo, J., & Omar, N. (2018). Random forest regression for online capacity estimation of lithium-ion batteries. *Applied Energy*, *232*, 197-210. https://doi.org/10.1016/j.apenergy.2018.09.182

Liu, T., Su, X., & Prigiobbe, V. (2018). Groundwater-Sewer Interaction in Urban Coastal Areas. *Water*, *10*(12). https://doi.org/10.3390/w10121774

Liu, X., & Yang, C. (2013, 16-18 Dec. 2013). A Kernel Spectral Angle Mapper algorithm for remote sensing image classification. 2013 6th International Congress on Image and Signal Processing (CISP),

Nguyen, L. V., Bui, D. T., & Seidu, R. (2022). Comparison of Machine Learning Techniques for Condition Assessment of Sewer Network. *IEEE Access*, *10*, 124238-124258. https://doi.org/10.1109/ACCESS.2022.3222823

Orhan, U., Hekim, M., & Ozer, M. (2011). EEG signals classification using the K-means clustering and a multilayer perceptron neural network model. *Expert Systems with Applications*, *38*(10), 13475-13481. https://doi.org/10.1016/j.eswa.2011.04.149

Salman, B., & Salem, O. (2012). Modeling Failure of Wastewater Collection Lines Using Various Section-Level Regression Models. *Journal of Infrastructure Systems*, *18*(2), 146-154. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000075

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*(3), 199-222.

Su, X., Liu, T., Beheshti, M., & Prigiobbe, V. (2020). Relationship between infiltration, sewer rehabilitation, and groundwater flooding in coastal urban areas. *Environmental Science and Pollution Research*, *27*(13), 14288-14298. https://doi.org/10.1007/s11356-019-06513-z

Suykens, J. A. K., & Vandewalle, J. (1999). Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, *9*(3), 293-300. 10.1023/A:1018628609742

Syachrani, S., Seok, H., David, Jeong, & Chung, C. S. (2013). Decision Tree–Based Deterioration Model for Buried Wastewater Pipelines. *Journal of Performance of Constructed Facilities*, *27*(5), 633-645. https://doi.org/10.1061/(ASCE)CF.1943-5509.0000349

Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics*, *17*(1), 168-192. https://doi.org/10.1016/j.aci.2018.08.003

Tien Bui, D., Pradhan, B., Lofman, O., & Revhaug, I. (2012). Landslide Susceptibility Assessment in Vietnam Using Support Vector Machines, Decision Tree, and Naïve Bayes Models. *Mathematical Problems in Engineering*, *2012*, 974638. 10.1155/2012/974638

Trafalis, T. B., & Ince, H. (2000, 27-27 July 2000). Support vector machine for regression and applications to financial forecasting. Proceedings of the IEEE-INNS-ENNS International

Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium,

Vazquezl, M. Y. L., Peñafiel, L. A. B., Muñoz, S. X. S., & Martinez, M. A. Q. (2021). A Framework for Selecting Machine Learning Models Using TOPSIS. In T. Ahram, *Advances in Artificial Intelligence, Software and Systems Engineering* Cham.

Wauters, M., & Vanhoucke, M. (2014). Support Vector Machine Regression for project control forecasting. *Automation in Construction*, *47*, 92-106. https://doi.org/10.1016/j.autcon.2014.07.014

Worldometer. (2022). *Norway Population (LIVE)*. Available online: https://www.worldometers.info/world-population/norway-population/ (accessed on March 08th, 2022)

Yin, X., Chen, Y., Bouferguene, A., & Al-Hussein, M. (2020). Data-driven bi-level sewer pipe deterioration model: Design and analysis. *Automation in Construction*, *116*, 103181. https://doi.org/10.1016/j.autcon.2020.103181

Zamanian, S., Hur, J., & Shafieezadeh, A. (2020). A high-fidelity computational investigation of buried concrete sewer pipes exposed to truckloads and corrosion deterioration. *Engineering Structures*, *221*, 111043. https://doi.org/10.1016/j.engstruct.2020.111043

Zendehboudi, A., Baseer, M. A., & Saidur, R. (2018). Application of support vector machine models for forecasting solar and wind energy resources: A review. *Journal of Cleaner Production*, *199*, 272-285. https://doi.org/10.1016/j.jclepro.2018.07.164

QGIS Development Team (2022). QGIS Geographic Information System. Open-Source Geospatial Foundation Project. http://qgis.osgeo.org.

Anon, (2020). Anaconda Software Distribution, Anaconda Inc. Available at: https://docs.anaconda.com/.