A prototype of a machine learning workflow to classify land use from housing market dynamics. Part of a Longitudinal Analysis of housing sales in the Greater Toronto-Hamilton Area.

by

Stepan Oskin

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Civil and Mineral Engineering
University of Toronto

Abstract

A prototype of a machine learning workflow to classify land use from housing market dynamics. Part of a Longitudinal Analysis of housing sales in the Greater Toronto-Hamilton Area.

Stepan Oskin
Master of Applied Science
Graduate Department of Civil and Mineral Engineering
University of Toronto
2019

There is ample evidence of the role of land use and transportation interactions in determining urban spatial structure. The new data sources introduced by increased digitization of human activity, such as Teranet's dataset of real estate sales, offer opportunities for development of integrated urban models or studies conducting longitudinal analysis of changes in land value distributions. To facilitate this, data from various sources (Census, TTS, etc.) needs to be merged at the land parcel level to enhance datasets with additional attributes, while maintaining the ease of data storage and retrieval. In addition, accurate land use information needs to be added to Teranet records to allow separating sales data by major property types. This thesis proposes a prototype of a workflow to augment Teranet's dataset with data from multiple sources and use machine learning to classify land use of each Teranet record based on the housing market dynamics.

Contents

Chapter 1

Introduction

The fundamental link between transportation and urban form creates a feedback relationship between land development, travel needs, viability of alternative modes, accessibility, and other important characteristics of the urban transportation system. Numerous "top-down" and "bottom-up" models have been designed to analyze and forecast the behaviour of urban regions and interaction of their transportation and land use systems. Since urban systems are complex in nature and require "re-solving" over and over, data science process models present a good fit for this task with their iterative structure.

Increased digitization of human activity, such as introduction of the POLARIS land registration system by the Government of Ontario in 1985, produces a wealth of new information that can be used to study interactions between land use and transportation at a fine spatial and temporal scale. Teranet's dataset of real estate transactions presents a wealth of information on the housing market of Ontario and can be used for empirical studies of transportation-land use interaction. However, along with the opportunities, the new data sources also present new challenges. Teranet's dataset has some data quality issues that need to be addressed and might require special skills to work with due to its size. But most importantly, it is very limited in the number of features available for each transaction.

One of the major challenges of working with Teranet's data is the lack of features describing each transaction, namely the type of property being sold. As Teranet's records have timestamps (dates) and coordinates of parcel centroids for each record, they can be joined with other data sources via temporal and spatial relationships, integrity of which needs to be maintained when combining data sources to ensure semantic interoperability. These relationships are implemented through a standardized data preparation workflow using Python via a series of jupyter notebook which was designed as a part of this master's thesis. In addition, since all the available sources of land use information have their limitations, a prototype of a machine learning workflow to classify land use from the housing market dynamics is proposed and tested within this thesis.

This thesis generally follows the structure proposed by CRISP-DM, a comprehensive data mining methodology and a process model[44] (shown on figure 1.1). CRISP-DM breaks down the life cycle of a data mining project into six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment.

The first phase of CRISP-DM is business understanding, which involves such key elements as definition of project goals and their acceptance criteria and definition of the target variable of analysis[36]. Chapter 2 focuses on definition of the goals of this thesis: it provides background information on land

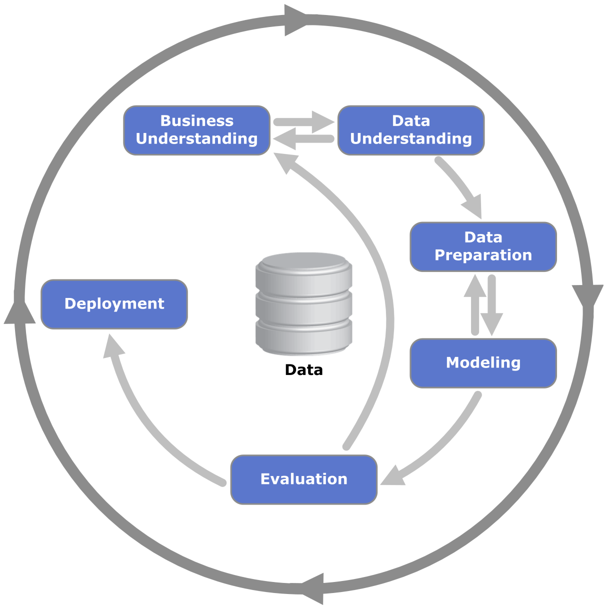


Figure 1.1: Phases of the CRISP-DM Reference Model, adapted from Shearer[44].

use and transportation models and some of the ongoing research efforts at UTTRI involving the housing market, outlines the role of Teranet's dataset in these efforts, challenges of its usage and the proposed solution. Definition of the target variable is discussed in chapter 5 and criteria for model evaluation are discussed in chapter 6. Chapter 3 covers the second phase of CRISP-DM process model, data understanding, and discusses the nature of spatial and temporal relationships of different data sources that are used in this master's thesis. Chapter 4 covers the third phase, data preparation, and presents the data preparation workflow designed to implement the relationships introduced in chapter 3. Chapter 5 covers the fourth phase, modeling, and describes a prototype of a machine learning workflow to classify land use from the housing market dynamics. Chapter 6 covers the fifth phase, evaluation, and presents the discussion of the results and chapter 7 presents the conclusion and outlines opportunities for future work.

Chapter 2

Background information

The first phase of the CRISP-DM process model for data mining projects is business understanding; this chapter discusses the complex interaction of land use and transportation, provides a brief overview of the history of development of land use and transportation (LUT) models, presents some legal and historical background for Teranet's dataset of land registration records, and finishes with a discussion of challenges of working with Teranet's data and the proposed solution.

2.1 Complexity of urban systems and "wicked" problems

In her famous 1961 book, Jane Jacobs[19] described a city as "a problem in organized complexity"; since then, many other researchers have remarked that urban systems exhibit complex behaviour[4, 7]. Complexity of a system can be defined as a state or quality of being intricate or complicated. For a system to be complex is not necessarily the same as to be complicated; complex systems can be simple, i.e. governed by a single equation. Complexity of a system has to do with the intrinsic ability of a system to surprise us with its behaviour; that the system is hard to understand, despite the mechanics of it being relatively simple.

In 1973, a little over a decade after Jacobs, Rittel and Webber[41] presented a path-breaking conceptualization; this conceptualization characterized urban planning problems as "wicked" problems: problems which cannot be definitively described and for which it makes no sense to talk of "optimal solutions". In their paper, Rittel and Webber stated that such "wicked" problems are never "solved", and that the focus instead becomes on iteratively "re-solving" the problems over and over. More than 40 years after their original publication, Rittel and Webber's ideas remain relevant to the policy sciences today: there is an intense interest in the nature of "wicked" problems and the complex tasks of identifying their scope, viable responses, and appropriate mechanisms and pathways to improvement[12]. Interaction between land use and transportation, which is discussed in the following section, presents a prime example of urban complexities and "wicked" problems.

2.2 Transportation-land use cycle

Among the reasons why transportation and land use interaction is "wicked" are such aspects as pluralism of expectations among stakeholders, institutional complexity in policy making, and scientific uncertainty[37]. More importantly, there is a fundamental link between transportation and urban form: urban form has an enormous impact on the type and cost of transportation systems needed to serve residents of a metropolitan area[21]. Transportation, in turn, influences land development and location choices of people and firms, and thus completes the formation of a feedback relationship that Stover and Koepke[47] referred to as a cycle. Interconnections between points (activities) in space can be perceived through the medium of the transportation system[34]

Figure 2.1 illustrates the complex interactions between land use and transportation system as summarized by Miller, Kriger and Hunt[34].

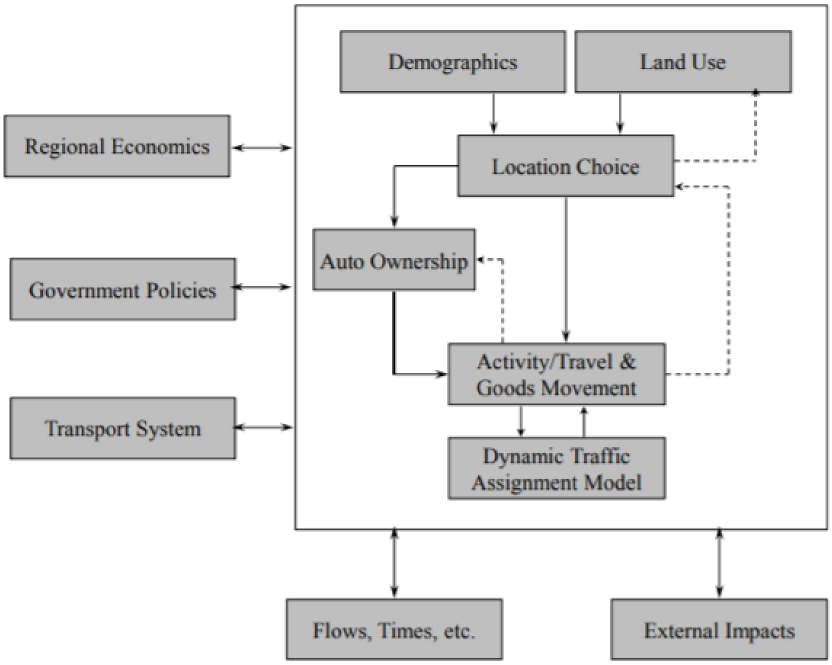


Figure 2.1: An Idealized Integrated Urban Model System, adapted from Miller, Kriger and Hunt[34].

Many different types of models are used in planning, such as demand forecasting models projecting traffic or ridership, or land use models projecting and distributing population and jobs within an area. At an earlier stage of model development, some analysts argued that there is no significant link between transportation and land use, given the near-ubiquity of the transportation (road) network[34]. However, the unprecedented urban growth of the 21st century introduced new challenges for urban systems such as extreme road congestion, equity of access to jobs and services among low-income households, energy scarcity, environmental and GHG impacts from transportation systems and public health impacts of land use patterns[30, 35].

It became apparent that these “transport problems” cannot be solved through transportation policies and investment alone, that the physical design of the city at the “macro” and “micro” scale critically interfaces with the demand for and performance of the transportation system. In addition, to accurately assess the costs and benefits of an expensive long-term transportation infrastructure investment, “feedback” effects of these investments on urban form, land values, property taxes, quality of life, etc. need to be quantified and included in evaluation and decision making. Thus, today there is a steadily growing recognition within the urban policy field that the interaction between transportation and land use does exist and does matter[30].

In the context of models, integrated urban models (IUMs) aim to capture the complex relationship between urban systems such as transportation and land use more accurately. Integrated land use-transportation models combine travel demand forecasting and land use forecasting functions and recognize that the distribution of population and jobs depends, in part, on transportation accessibility. The reverse is also true, and thus integrated models incorporate feedback relationship between transportation and land use, with economic decisions by households and firms acting as one of the links between the two systems[34].

2.3 Evolution of LUT models

The history of treating cities as systems via simulation models of transportation and land use dates back to the 1950s when General System Theory and Cybernetics came to be applied in the softer social sciences[4]. The first operational simulation model that truly integrated land use and transportation is considered to be A Model of Metropolis built in 1964 by Ira S. Lowry for the Pittsburgh region based on economic base theory[25]. It was a highly aggregate model based on theories of spatial interaction, such as the gravity model that was popular in quantitative geography and transportation planning at the time[8]. Models based on a spatial interaction framework continued to be developed through mid-1980s, until developments in random utility theory allowed researchers to describe choices among discrete alternatives, such as the choice of travel mode, and generate models based on the study of disaggregate behaviour[18].

Figure 2.2 provides the general overview of chronological development of LUT models as summarized by Iacono[18].

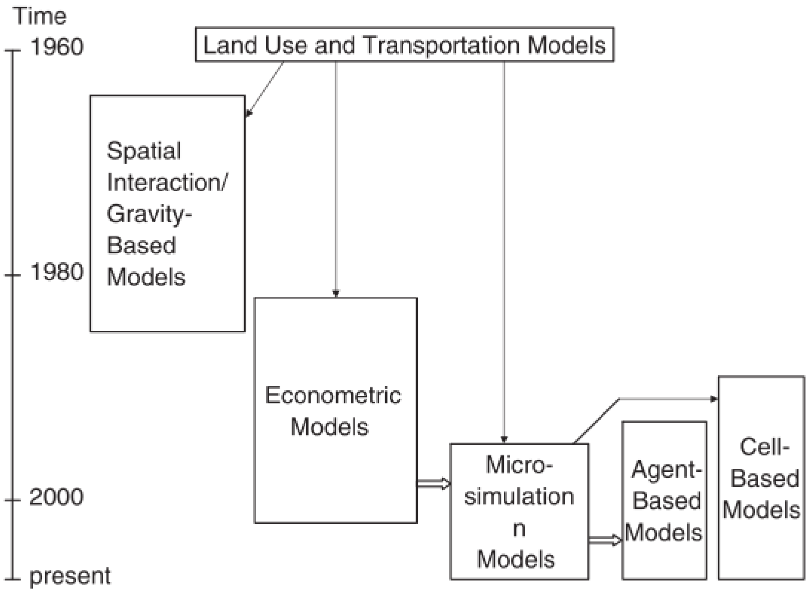

Time
1960
1980
2000
present
Land Use and Transportation Models
Spatial Interaction/ Gravity-Based Models
Econometric Models
Micro-simulation n Models
Agent-Based Models
Cell-Based Models

Figure 2.2: Chronological development of LUT models as summarized by Iacono[18].

The modeling paradigm changed fundamentally in the early 1990s along with the advances in computing power and efficiency of data storage. Urban systems used to be viewed as hierarchical and centrally organized equilibrium structures, or “top-down”. Instead, now they were considered to be structured from the “bottom-up”, dynamically retaining their integrity through interactions of numerous microelements[4]. A new broad class of LUT models that could fall under the title of “microsimulation” began to be developed: these included such classes of models as activity-based travel, cell-based models, multi-agent models, and more recently comprehensive urban microsimulation models that reflect the dynamics of changes in the population and the urban environment[18]. “Micro” in the microsimulation implies that the model must be highly disaggregated spatially, socio-economically and in its representation of processes. “Simulation” implies that the model must be numerical, stochastic, have an explicit time dimension, and “evolve” into the end state rather than “solve for it”[32]. An example of such a model has been developed by the University of Toronto ILUTE team; their product is an integrated urban model capable of microsimulating urban demographic evolution, housing markets and travel behaviour over extended periods of time[31].

The Integrated Land Use, Transportation, Environment (ILUTE) model system is a system of disaggregate agent-based microsimulation models for Greater Toronto–Hamilton Area (GTHA). It includes such components as land use, activity/travel, urban economics, auto ownership, demographics and emissions/energy use. The system uses disaggregate models of spatial socioeconomic processes to evolve from a known base case to a predicted end state of the GTHA in 1-year time steps[33]. In ILUTE, the system state of GTHA is defined in terms of the individual persons, households, dwelling units, firms, etc. that collectively define the urban region being modeled. Many markets are of interest within ILUTE, such as housing, labour, commercial, real estate, etc., and are modeled via microsimulation. For example in HoMES, the housing market module of ILUTE, houses are auctioned off one dwelling at a time to interested bidders in a disaggregate implementation of Martinez’ Bid Choice theory[43, 27].

Among the major barriers to implementation of integrated urban models since their introduction were such aspects as data hungriness and computational requirements[34]. However, as an increasing amount of aspects of human life becomes digitalized, a wealth of new data is produced and can be used to model and analyze dynamics of urban systems[1, 10]. Furthermore, continuing methodological advances in computing, such as cost-effective High Performance Computing (HPC), detailed GIS-based datasets and machine learning methods, mean that former barriers now represent opportunities for model system development[31]. An example of such digitalization of human activity is the introduction of the Province of Ontario Land Registration Information System (POLARIS) in 1985 by the Government of Ontario[48]. Introduction of POLARIS lead to a complete digitization of land registration records which in turn allowed the creation of an extensive dataset of real estate transactions by the Teranet Enterprises Inc. Due to its high spatial and temporal resolution, Teranet’s dataset holds a wealth of information about Ontario’s housing market from 1985 to 2017 and thus presents a valuable resource to inform and validate the design of ILUTE and HoMES. In addition, Teranet’s dataset plays an important part in the Longitudinal Housing Market Research conducted by UTTRI which is discussed in the following section.

2.4 Longitudinal Housing Market Research conducted by UTTRI

There is ample evidence of the role of land use and transportation interactions in determining urban spatial structure. Accessibility and mobility provided by transportation systems drive economic development and impact travel behaviour and location of households and firms. Similarly, urban sprawl and location of activities drive travel demand and the need for building transport networks. Land values in a metropolitan region are an outcome of such land use-transportation interactions.

Researchers at the University of Toronto Transportation Research Institute are conducting longitudinal analysis to identify trends and changes in land value distributions using the Teranet sales data over the 30-year period (1986-2016) in the Greater Toronto-Hamilton Area (GTHA). The aim is to understand the spatial-temporal effects of changes in socio-economic characteristics, transportation accessibility and built-environment on land values. Data from various sources (e.g., Census, Transportation Tomorrow Survey, etc.) is required to be merged at the land parcel level to enhance datasets with additional attributes, while maintaining the ease of data storage and retrieval for analysis as needed. In addition to that, accurate land use information needs to be added to Teranet records to allow separating sales data by major property types.

A brief legal and historical background of Teranet's dataset, main challenges of working with it and the proposed solution are discussed in the remainder of this chapter.

2.5 POLARIS: electronic system of land registration in Ontario

All land owned in Canada is registered in a public land registry in the applicable province. Each province and territory in Canada has its own land registry system, whether it is a land titles system, a registry system or a combination of both, with each system having its own rules. The registry system is a public record of documents evidencing transactions affecting land. In the land titles system, the applicable provincial government determines the quality of the title, and essentially guarantees (within certain statutory limits) the title to, and interests in, the property. As of 2015, most common law provinces and territories in Canada were using the land titles system or were in the process of converting from a registry system to a land titles system[29].

As of 2015, the Province of Ontario has largely converted from registry systems to a land titles system. In 1985, the Government of Ontario initiated the Province of Ontario Land Registration Information System (POLARIS) pilot project for the purposes of the conversion between systems and records automation. The Land Registration Reform Act (Ontario)[50] was introduced in 1990 to facilitate electronic search and registration of properties and the automation of paper-based records. POLARIS was built by the Province to house and process electronic land records, which in turn lead to the creation of an extensive dataset of land registration records managed by Teranet Enterprises Inc. Today, POLARIS is the search/registration and property maintenance system for all automated land records in Ontario.

2.6 Teranet's dataset, its challenges and the proposed solution

In 1991, the Government of Ontario established a partnership with Teranet, a Toronto-based organization, founded the same year, which provides e-services to legal, real estate, government, financial, and healthcare markets. The partnership was established to convert Ontario's land registration system to a more modernized electronic title system. The project involved taking a 200-year-old paper-based system and creating a database with electronic records for more than five million parcels of land. Teranet converted all qualified Registry properties in Ontario to the Land Titles system and automated existing paper Land Titles parcels. As a result, 99.9% of property in Ontario was parcelized and administered under the Land Titles system. Teranet fully automated the conversion of millions of paper-based documents and records into the Ontario Electronic Land Registration System (ELRS)[49]. Figure 2.3 presents an example of Teranet data: records from 2014 that fall within 1,000m of an arbitrarily selected target point, colored by land use spatially joined from DMTI datasets.

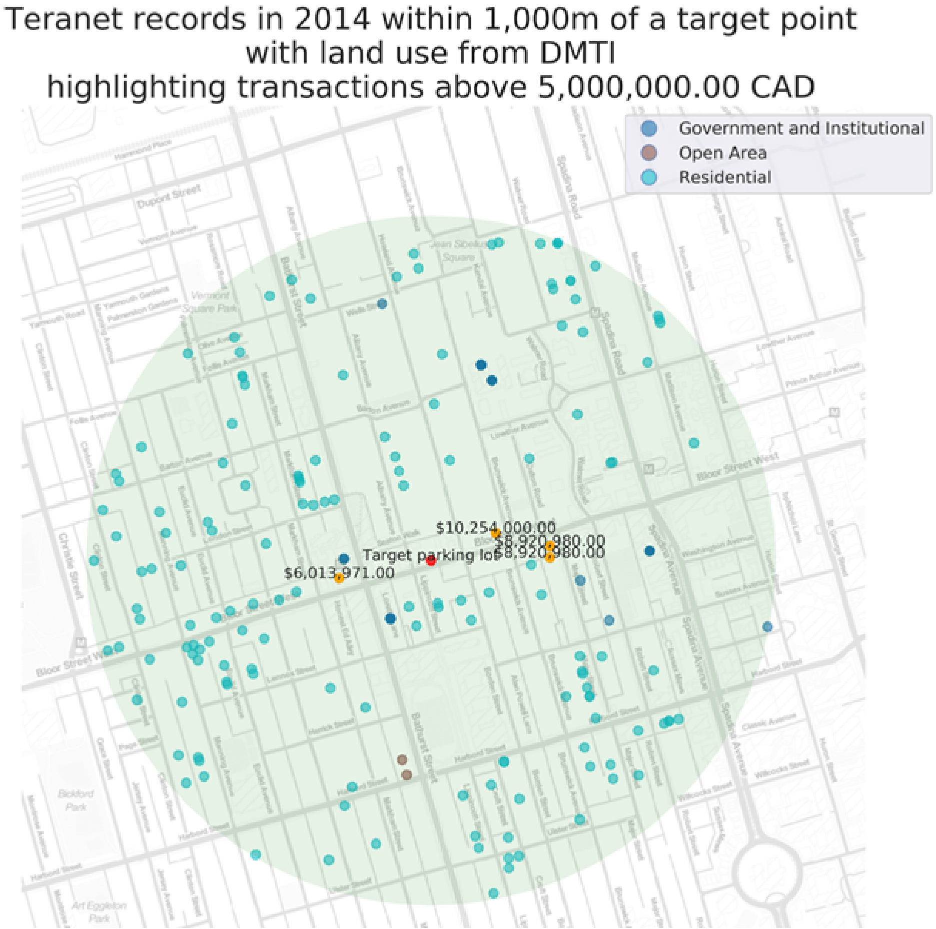


Figure 2.3: A sample map from Teranet dataset: records from 2014 within 1,000m of an arbitrarily selected target point, colored by land use spatially joined from DMTI datasets.

Teranet's dataset presents an extensive historical record of real estate transactions recorded in the Province of Ontario since the beginning of the nineteenth century. However, its available version also suffers from severe lack of features describing each transaction, which makes meaningful analysis or modeling difficult. At the same time, each Teranet transaction has a timestamp (date) and location information (x and y coordinates) and thus can be joined to variety of other geocoded urban data sources, such as Census demographics, Transportation Tomorrow Survey (TTS) and parcel-level land use information. However, joining these data sources together requires additional considerations, as they use different spatial units and are available at different temporal spans, as will be discussed in chapter 3.

One of the major attributes missing from the available version of Teranet's dataset is the information about the type of property being transacted, with records from various categories of residential, commercial and industrial properties all being mixed together in the same dataset. This introduces a major limitation on how Teranet's data can be used, limiting the ability to separate the transactions of different property types, identify submarkets of the housing market and see its fine-scale characteristics and dynamics. Detailed parcel-level land use information could offer some degree of filtering and can be joined from such sources as DMTI Spatial's land use and detailed land use manually collected by the Department of Geography at the University of Toronto.

However, these sources of land use information also have their limitations:

- DMTI's land use data does not offer any split between subcategories of residential properties and only covers the period of 2001–2014
- land use from the Department of Geography is much more detailed and accurate, but has been collected at a single point in time over the summer of 2012 and 2013
- neither of the available land use sources covers the full span of the Longitudinal Housing Market Research conducted by UTTRI (1986-2016)

At the same time, detailed land use from the Department of Geography offers an opportunity to train a machine learning model capable of recognizing certain property types that have characteristically different behavior on the housing market (e.g., high / low volume of transactions, median price ratio, etc.). Teranet records coming from a particular parcel (x and y coordinate pair) or pin (same parcel can include different pins, i.e. apartment complexes) can be combined to produce new features that characterize the behavior of this parcel or pin on the housing market. Combining these new features with spatially-joined variables from Census and TTS can yield a dataset that can be used to train and test a classification algorithm capable of determining the parcel land use based on the nature of its behavior on the housing market at Teranet transaction level (recognizing changes of land use with time).

The primary focus of this thesis consists of:

- augmenting Teranet's dataset with TTS, Census and land use variables, while maintaining the integrity of the spatial and temporal relationships between different data sources and ensuring the ease of storage and retrieval as needed for the Longitudinal Housing Market Research conducted by UTTRI
- investigating the opportunity for the implementation of a machine learning algorithm to classify land use based on the housing market dynamics

2.7 Chapter summary

The complex interaction of land use and transportation has been treated via simulation models since 1950s. Over the decades, the modeling paradigm has evolved from the highly aggregated and "top-down" representations of processes, such as gravity-based models, to highly disaggregated and "bottom-up", defining the macro state of the system through countless interactions of its microcomponents. These changes are represented in the family of microsimulation models which can define the state of an urban region in terms of the individual persons, households, dwelling units, firms, etc. and evolve from a know base case into a future state via simulation.

It also became more clear with time that land use and transportation systems are deeply interconnected, and thus must be modeled together to incorporate feedback relationship, as is attempted by integrated urban models. An example of the class of integrated microsimulation urban model systems, Integrated Land Use, Transportation, Environment, or ILUTE, model system has been developed at the University of Toronto for the Greater Toronto–Hamilton Area.

Since their introduction, among the major barriers to the implementation of integrated models were their data hungriness and computational requirements. However, continuing methodological advances in computing expand modeling capabilities and increased digitization of human activity creates new datasets with spatial and temporal resolution that was not available before. Together, these developments present opportunities for further improvement and validation of ILUTE and its modules.

An example of new emerging data sources is Teranet's dataset of real estate sales that was created after the introduction of POLARIS land registration system by the Province of Ontario in 1985. Teranet's dataset plays an important part in the Longitudinal Housing Market Research conducted by UTTRI . One of the major challenges of working with Teranet's data is the lack of features describing each transaction, namely the type of property being sold. As Teranet's records have timestamps (dates) and coordinates of parcel centroids for each record, they can be joined with other data sources via temporal and spatial relationships; nature of this relationships will be discussed further in chapter 3.

Chapter 3

Spatial and temporal relationships between data sources

The second phase of the CRISP-DM process model for data mining projects is data understanding; this chapter discusses the nature of spatial and temporal relationships of different data sources that are used in this master's thesis. As was discussed in section 2.6, one of the main challenges of working with Teranet's data is the lack of available features. At the same time, Teranet records have timestamps (dates) and location information (x and y coordinates) and thus can be joined to a variety of other urban data sources, such as Census demographics, Transportation Tomorrow Survey (TTS) and parcel-level land use information. However, as will be discussed in this chapter, these data sources use different spatial units and are available at different temporal spans; therefore, special consideration must be taken when joining data from these sources with respect to their temporal and spatial relationships to ensure semantic interoperability.

Different data sources joined to Teranet's dataset are described in this chapter, the implementation of the spatial and temporal relationships via the standardized data preparation workflow in Python and a PostgreSQL relational database are described in chapter 4. An Entity Relationship (ER) diagram for the database created as a part of this master's thesis can be found in Appendix A; its referential integrity constraints were implemented based on the spatial and temporal relationships between data sources that will be introduced in this chapter.

3.1 Description of data sources used

The following data sources were combined into the GTHA housing market database that was created as a part of this master's thesis:

1. Teranet's dataset

   Due to the introduction of POLARIS by the Province of Ontario in 1985 (discussed in section 2.5), Teranet's dataset includes a complete population of real estate transactions recorded in Ontario from 1985 up to October of 2017 (records prior to 1985 appear to be incomplete, see figure 3.1). Since Teranet's dataset has a high number of records, it can be used to investigate aspects relating to the housing market at a very fine spatial and temporal scale.

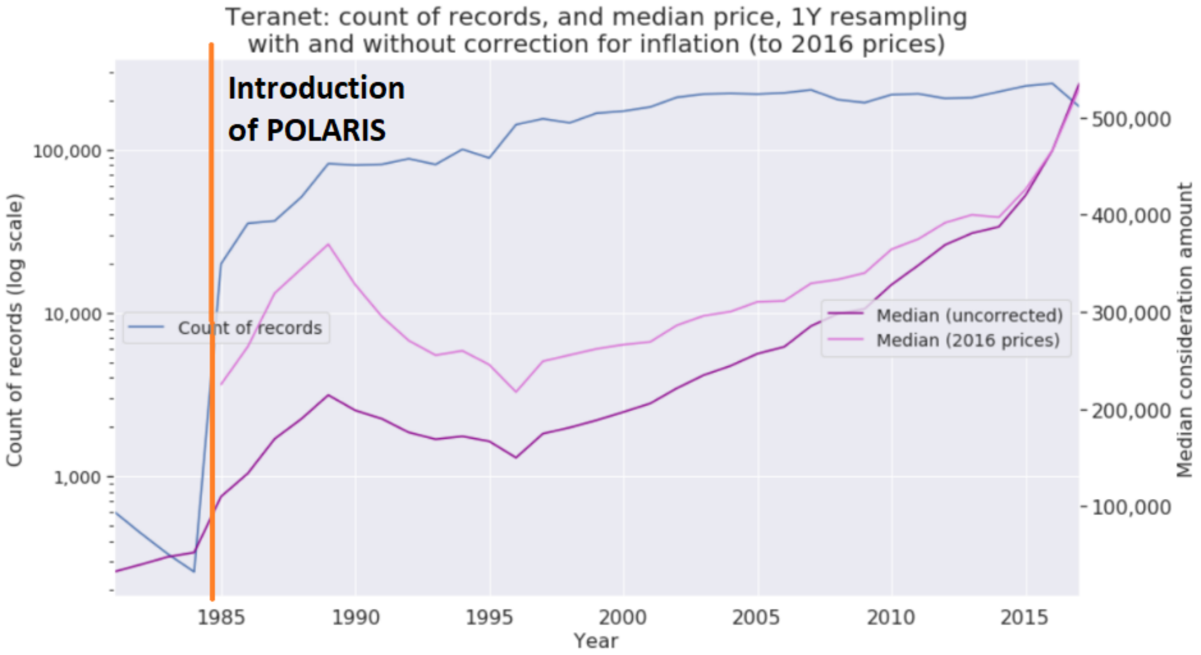


Figure 3.1: Time series of total count of Teranet records within GTHA boundary (log scale, left y-axis) and their median consideration amount (right y-axis), resampled by 1-year intervals. It can be seen that there is a dramatic increase in the total count of records between 1984 and 1985, which coincides with the introduction of POLARIS electronic land registration system by the Government of Ontario in 1985, which was discussed in section 2.5. Teranet records prior to 1985 appear to be incomplete.

2. Select variables from the Census of Canada

   One of the major sources of demographic and statistical data in Canada are the datasets collected under the national Census program. Census datasets provide valuable insights into the latest economic, social and demographic conditions and trends in Canada and are used to plan important public services. Statistics Canada collects every five years the national Census of Canada and disseminates the information by a range of geographic units, also referred to as ”Census geography”[26].

3. Select variables from the Transportation Tomorrow Survey (TTS)

   Another major source of information for most transportation planning studies concerned with Southern Ontario is the Transportation Tomorrow Survey (TTS), an origin-destination travel survey[13]. The Transportation Tomorrow Survey (TTS), undertaken every five years since 1986, is a cooperative effort by local and provincial government agencies to collect information about urban travel in southern Ontario. TTS represents a retrospective survey of travel taken by every member (age 11 or over) of the household during the day previous to the telephone or web contact. The information collected and the method of collection has remained relatively consistent over the seven surveys; TTS survey data includes characteristics of the household, characteristics of each person in the household, and details of the trips taken by each member of the household, including details on any trips taken by transit[2].

4. Land use from DMTI Spatial Inc. by year (2001-2014)

DMTI Spatial Inc., a Digital Map Products company, is a major provider of location based information in Canada. DMTI has been providing industry leading enterprise Location Intelligence solutions for more than a decade to Global 2000 companies and government agencies[16].

5. Detailed land use information from University of Toronto's Department of Geography collected in 2012 and 2013

   The detailed land-use data provided by University of Toronto's Department of Geography is a combination of parcel boundaries (from Teranet) and manually coded land-use data produced using Google maps and streetviews; it was collected by Prof. Andre Sorensen and Prof. Paul Hess's research project.

3.2 Spatial relationships between data sources

Most urban areas are divided into zones or planning areas on the basis of maintaining similar population sizes and following built or natural boundaries like roads or rivers. Census geography follows a certain hierarchy defined by Statistics Canada, with the largest top-level divisions being provinces and territories, and the lowest-tier divisions to which Census data is disseminated being Dissemination Areas (DAs)[46]. Statistics Canada defines a Dissemination Area as a small area composed of one or more neighbouring dissemination blocks, roughly uniform in population size targeted from 400 to 700 persons to avoid data suppression[45].

To simulate the changes in accessibility, metropolitan regions are usually broken down into a set of small geographic zones, similar (or in many cases identical) to the set of zones used for regional travel forecasting. For TTS variables, the finest level of spatial aggregation is that of the Traffic Zone, also referred to as the Traffic Analysis Zone (TAZ). A Traffic Zone is a polygon which typically falls along the centre line of roads or the natural geographic boundaries[14]. Not as a rule, but TAZ zones roughly follow Census tract boundaries, which are slightly bigger than DA boundaries. Figure 3.2 presents an example of TAZ polygons overlaid with Census DA boundaries.

TTS data has been collected for changing TAZ boundaries, or in other words, different zone systems due to growing population and expanding extents of the survey in the GTHA region over the years. To make the TTS data consistent for comparing over all years from 1986 to 2016, the Data Management Group (DMG) at the University of Toronto Transportation Research Institute (UTTRI), the custodian of the dataset derived from TTS, made all surveys available in the 2001 zone system, for convenience of researchers (any zone system could have been chosen for that matter). UTTRI used the 2001 TAZ system to model travel times for the GTHA on EMME for all TTS years based on the origin-destination trip data collected in the survey. The travel time data was used to create further transportation accessibility variables.

Land use data collected by DMTI and by the Department of Geography uses the spatial unit of a parcel polygon. Teranet records have attributes representing x and y coordinates matching parcel centroids.

Below is the summary of spatial units used by the data sources that were combined into the GTHA housing market database, designed and implemented as a part of this thesis:

- Point data

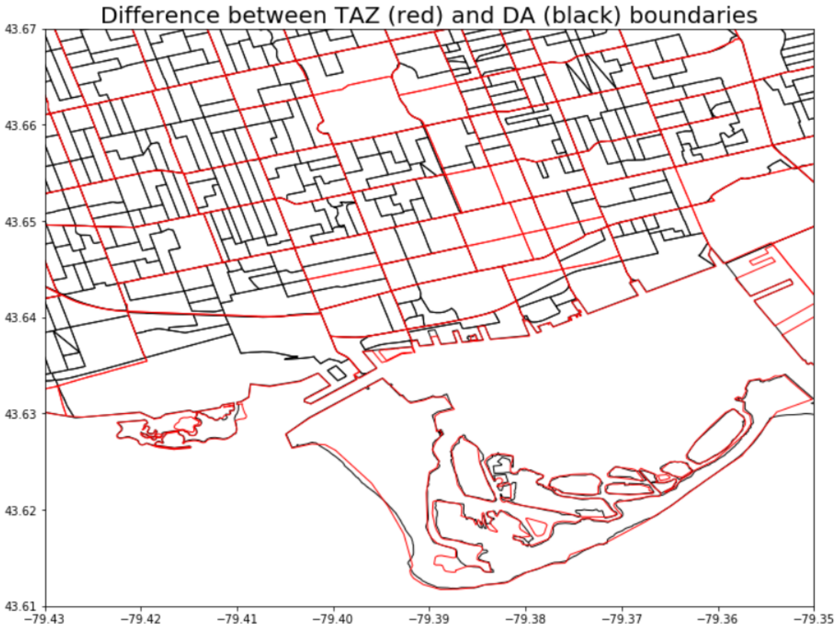


Figure 3.2: Spatial relationship between datasets: difference between Traffic Analysis Zones (TAZ, red) and Census Dissemination Area (DA, black).

  - Teranet

- Parcel-level data (polygons)
  - detailed land use from the Department of Geography
  - land use from DMTI
- DA-level data (polygons)
  - Census variables
- TAZ-level data
  - TTS variables

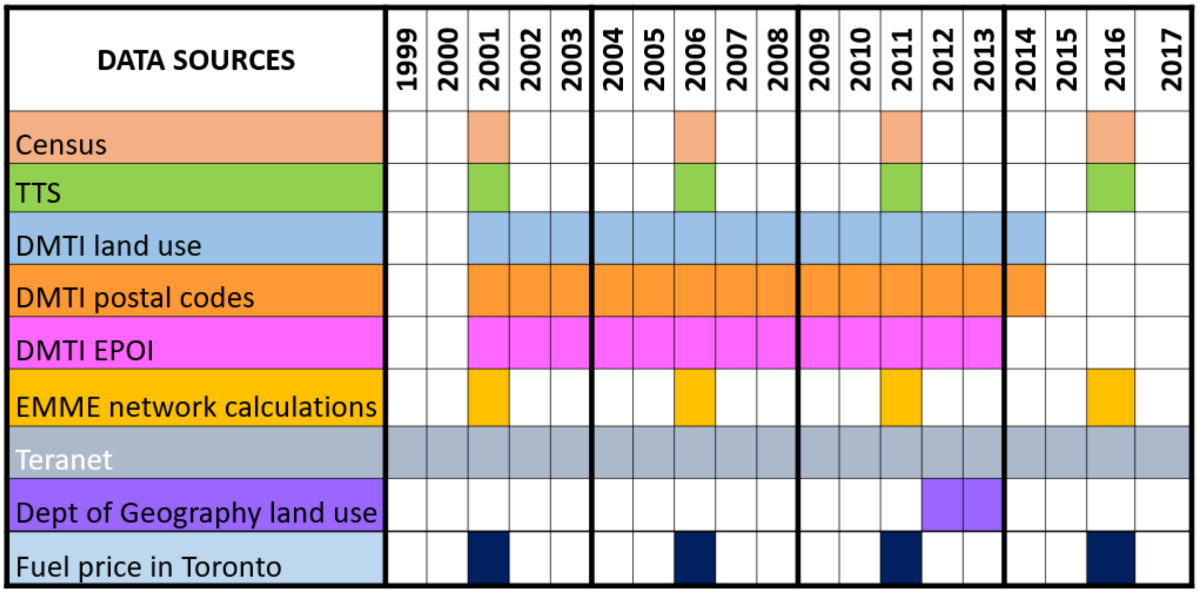
When joining these data sources, difference in spatial units needs to be respected, which can be more challenging when spatially joining polygons with polygons, since it might require area-weighted spatial interpolation of data to a common unit of analysis. In addition, polygon-based data can also vary with time, as is the case with DMTI's land use information, which is available by year. To simplify relating different polygon-based data sources with each other, all of them can be brought together to a single level of time-indexed points, such as Teranet transactions. This allows flexibility in combining data from polygon-based data sources to a common point level while maintaining the integrity of spatial and temporal relationships through polygon-to-point spatial joins. Implementation of these relationships is described in chapter 4, temporal relationships between different data sources are described in the following section.

3.3 Temporal relationships between data sources

In addition to using different spatial units, data sources joined with Teranet's dataset are available at different temporal spans:

- Teranet records have individual timestamps (date) on each record
- Census and TTS variables are sampled once in 5 years
- DMTI's land use data is available by year and covers a time span from 2001 to 2014
- Detailed land use from the Department of Geography was collected at a single point in time during the summers of 2012 and 2013

Temporal matching between Teranet records and DMTI data can be done directly: DMTI land use for each year from 2001 to 2014 can be spatially joined with a subset of Teranet records from the corresponding year; such approach would ignore changes of land use types that occur within a year, but would recognize land use changes between the years for which DMTI land use data is available. Since the detailed land use provided by the Department of Geography was collected at a single point in time, it can be joined to all Teranet records; however, it should be kept in mind that this land use data will be the most accurate around its time of collection in 2012 and 2013, and will become increasingly less accurate with an increase of the temporal span of Teranet records.



DATA SOURCES	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Census			■					■					■					■	
TTS			■					■					■					■	
DMTI land use			■	■	■	■	■	■	■	■	■	■	■	■	■	■			
DMTI postal codes			■	■	■	■	■	■	■	■	■	■	■	■	■	■			
DMTI EPOI			■	■	■	■	■	■	■	■	■	■	■	■	■				
EMME network calculations			■					■					■					■	
Teranet	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Dept of Geography land use														■	■				
Fuel price in Toronto			■					■					■					■	

Figure 3.3: Temporal spans of data sources used in the GTHA housing market database.

As for Teranet and Census / TTS variables, they can be matched in a number of ways:

1. Direct match with appropriate Teranet subsets
   - match Census / TTS variables only with Teranet records from the corresponding year (for example, Teranet records from 2016 matched with 2016 Census / TTS variables)

- benefits:
  - Census / TTS variables would be composed of the actual values recorded by the survey
- disadvantages:
  - limited use of Teranet data since only records from Census / TTS years can be matched

2. Interpolation of discrete Census / TTS variables
   - discrete Census / TTS variables can be turned into continuous via interpolation
   - benefits:
     - closest temporal match between Teranet and Census / TTS variables
   - disadvantages:
     - additional assumptions need to be made for each Census / TTS variable
3. Assign temporal spans to each Census / TTS survey as new features to Teranet records
   - each Census / TTS survey is assigned a temporal span of 5 years; this 5-year span represents a group of Teranet records to which variables from this survey can be matched (for example, Census variables of 2016 are matched with Teranet record from 2014 to 2018)
   - benefits:
     - avoid interpolation assumptions
   - disadvantages:
     - step-change in Census / TTS variables

To avoid additional interpolation assumptions and use the actual values recorded from Census and TTS surveys, the third option has been chosen for matching Teranet records with Census / TTS variables. Each Census / TTS survey is assigned a 5-year time span centered at the survey year (i.e., 2014–2018 for 2016 survey year) and new foreign keys are introduced to Teranet records to allow matching with 5-year time spans of Census / TTS variables. Implementation of temporal relationships for Teranet records will be described in chapter 4. Figure 3.3 presents the temporal spans assigned to each data source for joining with Teranet records.

3.4 Chapter summary

Variables that can be joined to augment Teranet's dataset, such as Census and TTS surveys and parcel-level land use data, are defined using different spatial units and are available at varying temporal spans. These relationships need to be respected when combining variables from these data sources into a single dataset to ensure semantic interoperability. The integrity of spatial relationships can be ensured by spatially joining all polygon-based data sources to Teranet points. Temporal relationships between DMTI land use and Teranet sales records are incorporated by performing separate spatial join operations for each annual DMTI land use dataset with a corresponding Teranet subset. For Census and TTS variables, additional foreign keys are introduced assigning 5-year spans to each Teranet record corresponding to a Census / TTS survey; these foreign keys indicate which Teranet records should be joined to a particular Census or TTS survey. Implementation of these relationships via a standardized data preparation workflow in Python and a PostgreSQL relational database are described in chapter 4.

Chapter 4

Data preparation

The third phase of the CRISP-DM process model for data mining projects is data preparation; this chapter discusses the steps taken to implement the spatial and temporal relationships between the different data sources combined into the GTHA housing market database; the nature of these relationships was discussed in chapter 3. In addition, this chapter introduces new features derived from native Teranet attributes for land use classification, which will be described in chapters 5 and 6.

The “wicked” nature of transportation and land use interaction introduced in chapter 2 dictates the need to iteratively “re-solve” transportation and land use planning problems instead of focusing on finding some single “optimal solution”. This approach resembles the methodologies typically employed for data science projects, where the sequence of steps is iterated over, producing a more meaningful solution on each new iteration of the cycle, as defined by such process models as CRISP-DM[44]. Similarly, data preparation can be followed in a linear manner, but is very likely to be iterative in nature[9].

Data preparation plays a critical role in research projects:

- it is a prerequisite for any meaningful analysis
- data quality and the amount of useful information that it contains can determine the success of application of machine learning algorithms[40]
- it is often required to allow the introduction of constraints necessary for implementation of RDBMS

To facilitate easy modification and replication of the data preparation process for data sources related to the GTHA housing market database, a streamlined data preparation workflow using Python via a series of jupyter notebooks has been established as a part of this master’s thesis. It accomplishes three main objectives:

- Clean Teranet dataset and correct its records for consistency.
- Introduce new keys that would allow efficient joining of other data sources, such as Census and TTS variables or parcel-level land use information, while maintaining the integrity of spatial and temporal relationships that were discussed in chapter 3.

- Engineer the new features that can be used by the machine learning algorithm along with the features from the joined datasets to classify land use, which will be discussed in chapters 5 and 6.

This chapter introduces the concepts of “Tidy Data” and database normalization and outlines the standardized data preparation workflow that was designed to implement the spatio-temporal relationships discussed in chapter 3 for all data sources that were combined into the GTHA housing market database that was implemented as a part of this master’s thesis.

4.1 Tidy data and database normalization

Hadley Wickham in his paper “Tidy Data”[51] formalized the way in which a shape of the data can be described and what goal should be pursued when formatting data. The “tidy data” standard is closely related to Edgar F. Codd’s relational algebra and has been designed to facilitate initial exploration and analysis of the data, and to simplify the development of data analysis tools that work well together. As an integral part of his relational model, Codd[11] proposed a process of database normalization, or restructuring of a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. Normalization entails organizing the columns (attributes) and tables (relations) of a database to ensure that their dependencies are properly enforced by database integrity constraints. The principles of “tidy data” formulated by Wickham essentially reformulate Codd’s ideas in statistical language.

According to Wickham[51], “tidy data” is a standard way of mapping the meaning of a dataset to its structure. A dataset is “messy” or “tidy” depending on how rows, columns and tables are matched up with observations, variables and types.

In “tidy data”:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

This definition of “tidy data” matches Codd’s 3rd normal form[11], but with the constraints framed in statistical language, and the focus put on a single dataset rather than on many connected datasets common in relational databases. “Messy data” is any other arrangement of data.

The structure of Teranet’s dataset conforms with the “tidy data” format. Contrary to Teranet, tables with select Census and TTS variables had variables for different Census and TTS years recorded as columns. This needed to be addressed by unpivoting these tables and introducing a new attribute ‘year’ to be used as a part of a composite foreign key when joining these variables with Teranet records; thus, both spatial and temporal relationships that were introduced in chapter 3 can be respected when combining these data sources together.

Tables with select Census and TTS variables were unpivoted to conform with the “tidy data” format:

- each Census / TTS variable now forms a single column

- each value of Census / TTS variables is indexed by a composite primary key constituting of its spatial identifier ('DAUID' or 'TAZ_O', introduced in chapter 3) and the year of the survey

Introduction of the new foreign keys to Teranet records is described in the following section.

4.2 Introduction of new keys and attributes via spatial and temporal relationships

To implement the spatial and temporal relationships between the data sources discussed in chapter 3, a number of new foreign keys needed to be introduced to Teranet records. The new foreign keys either represent spatial identifiers (such as 'dauid' or 'taz_o', corresponding to DA or TAZ within which a Teranet record is located), or an attribute identifying the year of Census or TTS survey, variables from which can be joined with this Teranet record. Foreign keys representing spatial identifiers were added via a series of spatial joins while foreign keys identifying temporal spans were generated based on temporal spans defined earlier.

New spatial identifiers were introduced to each Teranet record via a series of spatial joins:

1. 9,039,241 Teranet points were joined with 9,182 polygons of Dissemination Areas (DAs) for GTHA used by Census variables
    - Teranet records with coordinates falling outside of GTHA boundary were filtered out
    - From the original 9,039,241 records, 6,803,691 remained in the dataset
    - New foreign keys 'dauid', 'csduid' and attribute 'csdname' were added to each Teranet record
2. 6,803,691 Teranet points were joined with 1,716 polygons of Traffic Analysis Zones (TAZ) used by TTS variables
    - New foreign key 'taz_o' was added to each Teranet record
3. 6,803,691 Teranet points were joined with 525 polygons of Forward Sortation Areas (FSA) and 555,668 polygons of postal geography from DMTI's Platinum Postal Geography Suite
    - New foreign keys 'fsa' and 'pca_id' and attribute 'postal_code_dmti' were added to each Teranet record
    - These keys are not currently used for joining any variables, but were added to expand the potential for relating datasets, as might be needed for the Longitudinal Housing Market Research conducted by UTTRI, which was introduced in section 2.4
4. 6,803,691 Teranet points were joined with 1,664,862 polygons of parcel-level detailed land use provided by the Department of Geography
    - New foreign keys 'pin_lu', 'landuse' and 'prop_code' were added to each Teranet record
    - Foreign keys 'landuse' and 'prop_code' are codes that can be converted to land use categories that were used by the Department of Geography for GTA and Hamilton, respectively

- For records from Hamilton, 'prop_code' was converted to categories used by GTA land use and reassigned to 'landuse', bringing GTA and Hamilton records to a single system of land use categories

5. Subsets of Teranet points were joined with corresponding yearly polygons of parcel-level land use from DMTI

   - New attribute 'dmti_lu' was added to each Teranet record

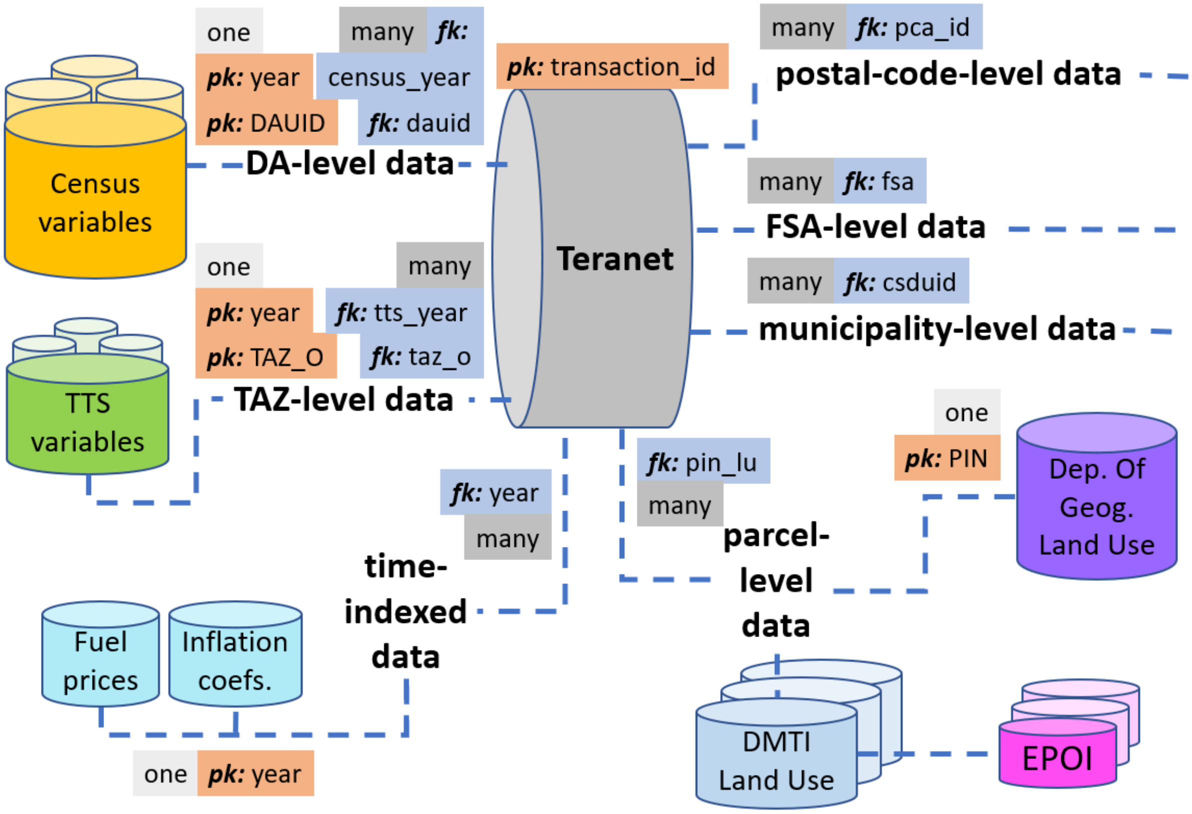


Figure 4.1: Relationships between datasets introduced during data preparation were then used to set up referential integrity constraints of the new PostgreSQL database for GTHA housing market data. The Entity Relationship (ER) diagram for the database created as a part of this thesis can be found in Appendix A; its referential integrity constraints were implemented based on the spatial and temporal relationships between data sources that were introduced in chapter 3.

Foreign keys representing temporal identifiers used for matching Teranet records with Census / TTS variables were generated from the registration date of each Teranet record, matching each year of Teranet records with a corresponding 5-year span covered by a Census or TTS survey, as was discussed in section 3.3. Diagram of relationships that were introduced between datasets for the implementation of RDBMS and their primary and foreign keys is presented on figure 4.1.

Following the steps described above ensures that the integrity of spatial and temporal relationships is maintained when combining attributes from different data sources at Teranet transaction level, as required by the Longitudinal Housing Market Research conducted by UTTRI, which was introduced in section 2.4. For example, Teranet records from 2007 would be spatially joined with DMTI land use data from 2007, and are matched by their attributes 'census_year' and 'tts_year' to Census and TTS variables from 2006 Census and TTS surveys. Census and TTS variables can be joined by appropriate 'dauid' and 'taz_o' (composite foreign keys are used when joining), and thus all data sources can be spatially and temporally aligned at the level of Teranet transactions.

Typically, a database table has an attribute, or a combination of multiple attributes, whose values are unique across the whole table; minimal combinations of attributes allowing unique identification of records are also referred to as candidate keys. Primary keys, chosen from candidate keys, are one of the most important concepts in database design. Almost every database table should have a primary key, chosen from a set of candidate keys. The main purpose of a primary key is to uniquely identify records in a table, and ideally, primary keys are determined using as few columns as possible.

In case of Teranet records, no combination of columns constitutes a candidate key, as it is possible to have two valid Teranet records that have duplicated values across all native features. For example, two same-price condo units sold in the same building on the same day can have duplicated values across all the native Teranet features. Such a scenario is not impossible since condo sales often come in large batches of sales over a short period of time; and in some cases, all records from a building are recorded under a single 'pin'.

To address this, a new surrogate key (artificial unique identifier for RDBMS) was added to each Teranet record via a new attribute 'transaction_id'. With this new attribute, Teranet's dataset can fit into a normalized database using the new attribute 'transaction_id' as its primary key. As was discussed in section 4.1, the unpivoted tables with select Census and TTS variables have composite primary keys consisting of a unique spatial identifier ('DAUID' or 'TAZ_O', respectively) and the year of the survey. Thus, semantic interoperability can be ensured when joining variables from these sources at the Teranet transaction level. Relationships introduced via the operations described in this section formulate referential integrity constraints that have been used to set up a PostgreSQL database of GTHA housing market data; its Entity Relationship (ER) diagram can be found in Appendix A.

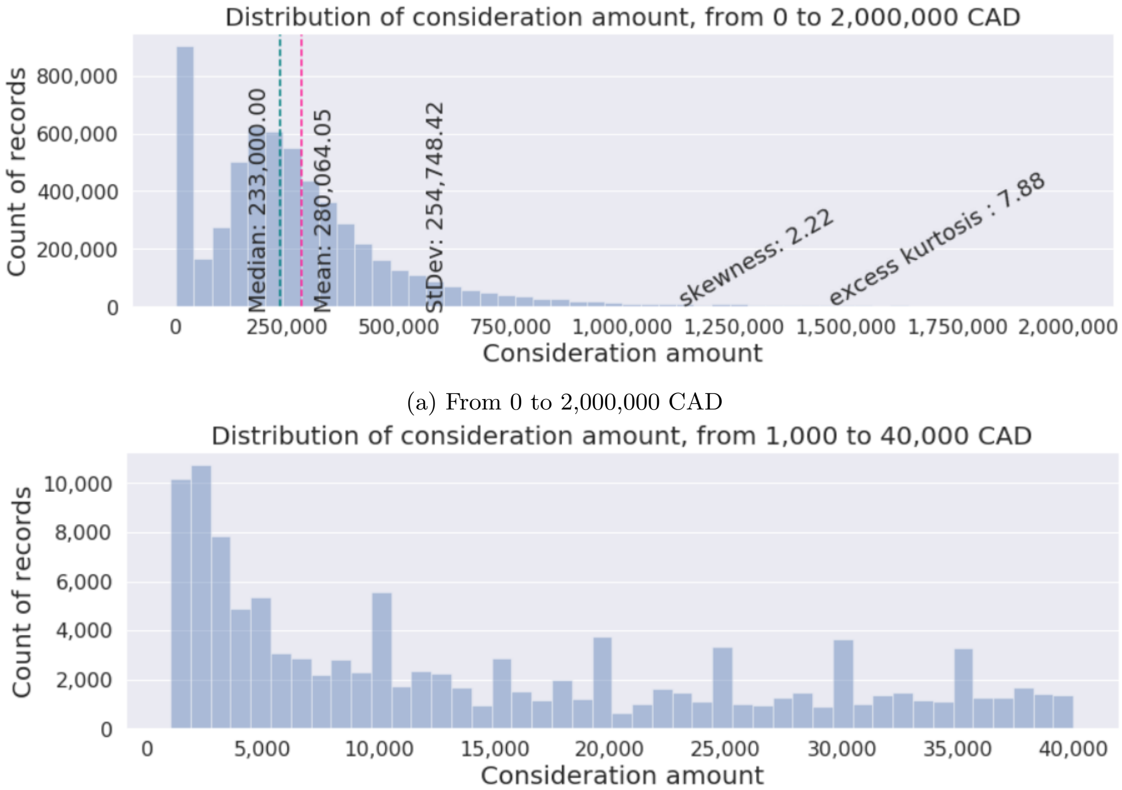
4.3 Outliers

Outliers present an issue for machine learning as they lead to difficulties in visualizing data and, more importantly, can degrade the predictive performance of an algorithm. In addition, outliers, if left unscaled, can significantly slow down or even prevent the convergence of many gradient-based algorithms, such as Logistic Regression[23]. However, an exact definition of what constitutes on outlier within a given dataset presents a challenge in itself, since it depends on hidden assumptions regarding the data structure and the applied detection method[6]. As defined by Hawkins, an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism[17].

In terms of transaction price distribution, outliers at both the high and the low end are present in Teranet's dataset:

- There is a high number of transactions with very low consideration amounts (starting from a few dollars, shown on figure 4.2) that most likely represent transactions recording gifts of property, where some symbolic consideration amount has been used.

- At the same time, since Teranet's dataset includes sales of all categories of properties, some records



(a) From 0 to 2,000,000 CAD

(b) From 1,000 to 40,000 CAD

Figure 4.2: Outliers at the bottom end of the price distribution most likely represent gift transactions. Since no clear break can be identified, 10,000 CAD was used as the bottom cut-off threshold for consideration amount to filter Teranet records.

have significantly higher consideration amounts that can range in the hundreds of millions and billions of dollars; these records most likely correspond to sales of large commercial and industrial properties or whole residential buildings.

Since low outliers represent transactions that are not useful for analysis, they were removed from the dataset. However, there seems to be no way to establish what constitutes a reasonable bottom cut-off threshold, as there was no criteria available on how to define a gift transaction; furthermore, no clear break could be identified on the low end of the price distribution. Since there seems to be an exceptionally large spike of transactions with consideration amount under 10,000 CAD, they were considered to be low outliers and were removed from Teranet's dataset, further reducing the number of records from 6,803,691 to 5,188,513.

In case of outliers at the high end of the price distribution, they most likely correspond to transactions of expensive commercial and industrial property or whole residential buildings. Since these transactions are useful for research questions concerning commercial and industrial property, they were left in the dataset and instead have been marked with special new attributes "outlier" using different criteria. Since, again, there was no clear criteria available as to what would constitute a top outlier, instead of using a single criterion, seven different Boolean variables were added describing whether a record belongs to outliers according to a particular condition or not. For example, feature 'outlier_y_10' is a Boolean variable capturing if the price of a record, corrected for inflation, is over 10 times greater than the median price of all records for the corresponding year. Table 4.1 presents the outlier categories that were used and the count of Teranet records that each of them marked to belong to outliers.

Category	Count of records
outlier_y_3	251,537
outlier_y_5	137,814
outlier_y_10	84,260
outlier_y_20	53,866
outlier_xy_2	160,662
outlier_xy_4	57,766
outlier_xy_10:	35,778

Table 4.1: Categories of outliers and the count of Teranet records that were marked by them. For categories coded 'y', price of each record was compared to median price of all Teranet records for that year; for categories coded 'xy', prices were compared to median price of all records for that coordinate pair; a number represents the threshold that the ratio needed to exceed for a record to be considered an outlier.

These new "outlier" attributes were used when filtering records during the assignment of reduced land use classes, which will be discussed in section 5.1; in addition, they were tested as a part of the original full feature set for classification of land use along with the other new features engineered from Teranet's dataset, which will be discussed in section 4.4. However, in the case of classification, all of the new "outlier" features were filtered out in the process of feature selection; feature selection will be discussed in section 5.3. Instead, algorithms performed better with features that numerically represent price, such as 'med_price_xy', computed as the median consideration amount (in 2016 dollars) for a coordinate pair.

Some of the new features that have been engineered from original Teranet attributes, which will be discussed in section 4.4, also contain outliers on the high end of their distributions. For example, in the case of the feature 'xy_prev_sales' which captures the rolling count of Teranet records from a coordinate pair, counts of records from a coordinate pair that corresponds to a house would be in the order of 10^1, while counts of records from a coordinate pair that corresponds to an apartment building would be on the order of $10^2 - 10^4$. In these cases, outliers contain valuable information that can be used to classify Teranet records by property type, but these outliers still present a challenge for the convergence of gradient-based algorithms. To keep these records in the dataset while facilitating faster convergence, feature scaling was performed and will be discussed in section 5.5.

4.4 Engineering new features for the classification algorithm

In addition to producing new keys for joining datasets, a number of the new features was engineered from original Teranet attributes to be tested with classification algorithms (discussed in chapters 5 and 6). These new features were intended to give each Teranet record spatial and temporal "context" of the housing market dynamics by grouping records using different criteria. For example, feature 'xy_prev_sales' was computed as the rolling count of Teranet records coming from a particular coordinate pair; feature 'price_to_med_year' captures the ratio of consideration amount of a record to the median consideration amount of all Teranet records for the corresponding year, etc.

The following features have been added to each Teranet record from 1985 to 2017 ('xy' represents 'x' and ‘y’ coordinates concatenated together as strings, used to group together all records coming from a particular coordinate pair):

- ‘price_2016’: consideration amount (in 2016 dollars) using the coefficients from the Inflation Calculator provided by the Bank of Canada[3]
- ‘pin_total_sales’: count of Teranet records grouped by ‘pin’
- ‘xy_total_sales’: count of Teranet records grouped by ‘xy’
- ‘pin_prev_sales’: rolling count of Teranet records grouped by ‘pin’
- ‘xy_prev_sales’: rolling count of Teranet records grouped by ‘xy’
- ‘xy_first_sale’: a Boolean variable indicating whether it is the first record coming from this ‘xy’
- ‘pin_years_since_last_sale’: difference in years since the last record coming from this ‘pin’
- ‘xy_years_since_last_sale’: difference in years since the last record coming from this ‘xy’
- ‘xy_years_to_next_sale’: difference in years to the next record coming from this ‘xy’
- ‘da_days/years_since_last_sale’: difference in days/years since the last sale that occurred on this Dissemination Area
- ‘xy_sale_next_6m’: a Boolean variable indicating whether there will be another sale on this ‘xy’ in the upcoming 6 month
- ‘pin_price_cum_sum’: cumulative sum of price (in 2016 dollars) for all Teranet records coming from this ‘pin’
- ‘xy_price_cum_sum’: cumulative sum of price (in 2016 dollars) for all Teranet records coming from this ‘xy’
- ‘pin_price_pct_change’: percentage change of price (in 2016 dollars) from the last Teranet record from this ‘pin’
- ‘xy_price_pct_change’: percentage change of price (in 2016 dollars) from the last Teranet record from this ‘xy’
- ‘price_da_pct_change’: percentage change of price (in 2016 dollars) from the last Teranet record from this Dissemination Area
- ‘med_price_xy’: median price (in 2016 dollars) for all Teranet records from this ‘xy’
- ‘med_price_year’: median price (in 2016 dollars) for all Teranet records for this year
- ‘price_to_med_xy’: ratio of price (in 2016 dollars) to the median price of all records for this ‘xy’
- ‘price_to_med_year’: ratio of price (in 2016 dollars) to the median price of all records for this year
- ‘outlier_y_3’: a Boolean variable marking as outliers all records with the price more than 3 times greater than median for that year

- 'outlier_y_5': a Boolean variable marking as outliers all records with the price more than 5 times greater than median for that year
- 'outlier_y_10': a Boolean variable marking as outliers all records with the price more than 10 times greater than median for that year
- 'outlier_y_20': a Boolean variable marking as outliers all records with the price more than 20 times greater than median for that year
- 'outlier_xy_2': a Boolean variable marking as outliers all records with the price more than 2 times greater than median for that 'xy'
- 'outlier_xy_4': a Boolean variable marking as outliers all records with the price more than 4 times greater than median for that 'xy'
- 'outlier_xy_10': a Boolean variable marking as outliers all records with the price more than 10 times greater than median for that 'xy'

These new features were combined with TTS and Census variables via spatial and temporal relationships that were introduced in chapter 3 and were used to train and test a classification algorithm to classify land use at Teranet transaction level, which is discussed in chapter 5.

Some of the new engineered features, such as 'xy_years_since_last_sale' and 'xy_years_to_next_sale' contain missing values in each case of the first or the last Teranet record coming from a coordinate pair. Since classification algorithms cannot handle inputs with missing values, during classification these values have been replaced with the corresponding median, which will be discussed in section 5.2. The recorded version of Teranet's dataset with the new features and keys would still have them as missing.

4.5 Chapter summary

The principles of "Tidy data", closely related to Codd's principles of database normalization, formalize the way how a shape of the data can be described and what goal should be pursued when formatting data. The "tidy data" standard facilitates initial exploration and analysis of the data and simplifies the development of data analysis tools that work well together. The structure of Teranet's dataset conforms with the "Tidy data" format. Contrary to Teranet, tables with select Census and TTS variables had variables for different Census and TTS years recorded as columns and thus needed to be unpivoted. A new attribute 'year' was introduced to these tables to be used as a part of a composite foreign key when joining with Teranet records.

New foreign keys representing spatial identifiers, such as 'dauid' and 'taz_o', were added to Teranet records via a series of spatial joins. During the first join with Dissemination Area geometry used by Census, Teranet records with coordinates falling outside of GTHA have been filtered out. Furthermore, Teranet records with consideration amount under 10,000 CAD were considered to be outliers at the low end of price distribution and were removed from the dataset. Outliers at the high end of the price distribution were not removed, but instead were marked by seven new Boolean attributes using various criteria to define a top outlier. Finally, new features have been engineered to be tested with classification algorithms, which will be described in chapters 5 and 6. In addition, datasets with new keys and features have been loaded into a PostgreSQL database established using referential integrity constraints based on the spatial and temporal relationships that were introduced in chapter 3.

Chapter 5

A prototype of a machine learning workflow to classify land use

The fourth phase of the CRISP-DM process model for data mining projects is modeling; this chapter introduces a prototype of a machine learning workflow to classify land use from the housing market dynamics using the augmented Teranet dataset, production of which was described in chapter 4. As was previously mentioned in section 2.6, one of the major features missing from the available version of Teranet's dataset is the information about the type of property being transacted, which introduces a major limitation on how Teranet's data can be used. As was described in chapter 4, parcel-level land use information from DMTI and the Department of Geography has been spatially joined to Teranet records, but these sources of land use information also have their limitations:

- DMTI's land use data does not offer any split between subcategories of residential properties and only covers the period from 2001 to 2014
- land use data from the Department of Geography is a lot more detailed and accurate, but has been collected at a single point in time over the summer of 2012 and 2013
- neither of the available land use sources covers the full span of the Longitudinal Housing Market Research conducted by UTTRI (1986–2016)

To address this issue, detailed land use from the Department of Geography can be used as labelled data to train a machine learning model capable of recognizing certain property types that have characteristically different behavior on the housing market. For example, the proposed model would be able to differentiate a detached house from a condo through such features as high / low volume of transactions, ratio of price to median price for that year, etc. Chapter 4 described the production of a dataset that combines the new features engineered from Teranet data with Census and TTS variables based on spatial and temporal relationships that were discussed in chapter 3. In this chapter, this dataset is used to investigate the opportunity to implement a classification algorithm to determine the parcel land use at Teranet transaction level based on the housing market dynamics (land use is determined for each Teranet record, recognizing the changes of land use on the same parcel with time). This way, a machine learning algorithm could provide a scalable solution to automate a labour-intensive task of collecting parcel-level detailed land use and expand the temporal span for which the land use data collected by the Department of Geography can be used with accuracy.

5.1 Selecting and encoding the target variable

According to the results of an Exploratory Data Analysis (EDA), different property types have characteristically different behaviour on the housing market. Figure 5.1 shows the distributions of total count of Teranet records per coordinate pair by the 10 most frequently encountered land use categories used in the Department of Geography's dataset.

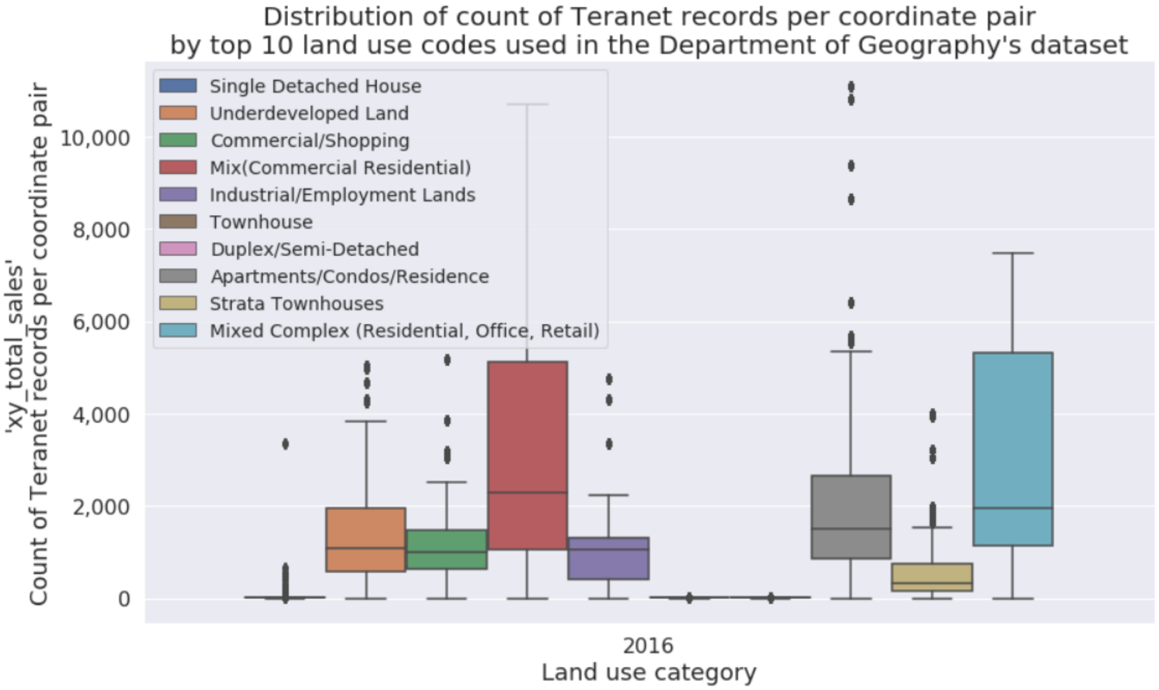
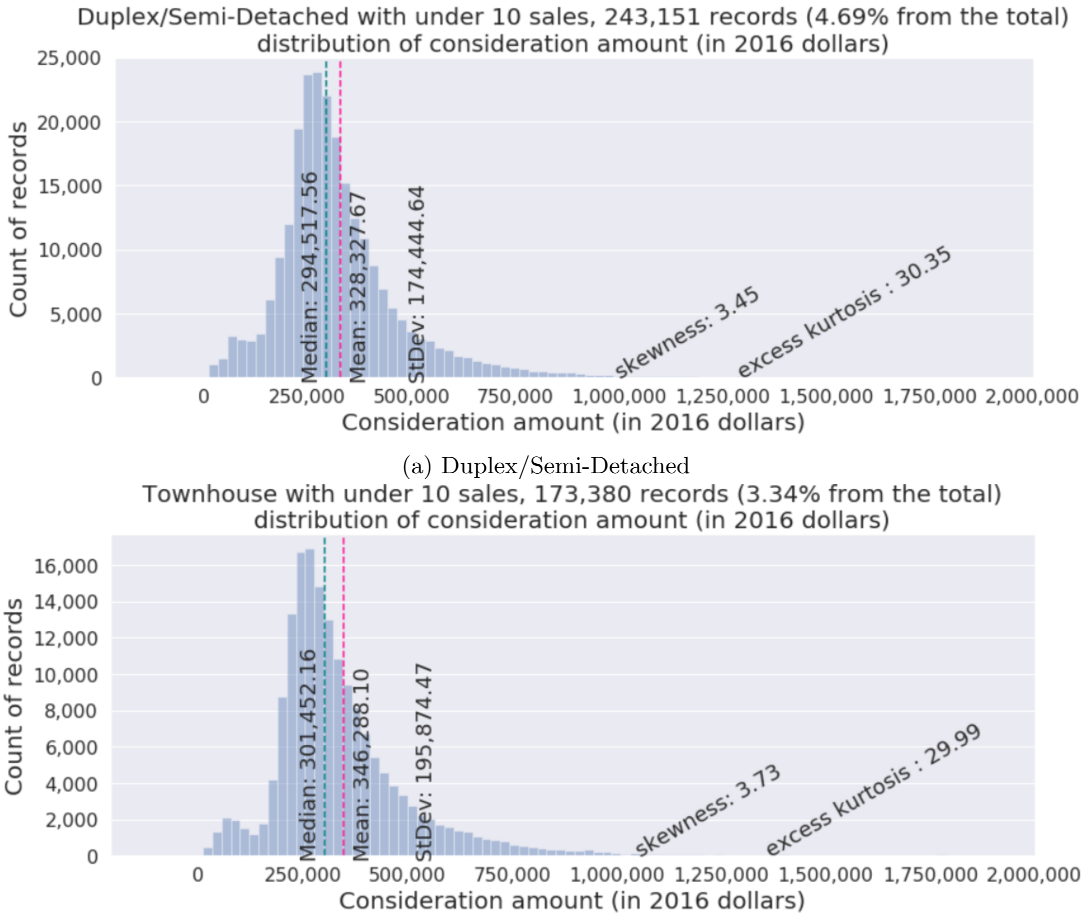


Figure 5.1: Distributions of total count of Teranet records per coordinate pair by the top 10 land use categories. It can be seen that detached houses, duplexes, and townhouses ("collapsed" boxplots) have much lower numbers of records per coordinate pair. It can also be seen that there are some outliers present on the category of detached houses (first boxplot from the left). These outliers most likely represent mislabelled records, as it is very unlikely that a detached house can have almost 4,000 sales.

The target variable used for the purposes of investigating the possibility of land use classification using the housing market dynamics was constructed by reducing the land use codes found in the Department of Geography's land use dataset. Department of Geography's land use categories that have the highest counts of Teranet records have been used to reduce all different property types to the three major land use classes. Many machine learning algorithms are subject to a frequency bias in which they place more emphasis on learning from data observations which occur more frequently; to address this, the three classes were selected to have a comparable number of Teranet records between themselves and thus produce a more balanced dataset. In addition, the chosen groupings combine categories that have a similar distribution of price and count of sales per coordinate pair between land use categories that were grouped together to form a single class. For example, detached and semi-detached houses and townhouses would have a much smaller frequency of transaction and a higher median price per coordinate pair when compared to condos and strata townhouses. Figure 5.2 shows an example of such grouping: for target class 1, Duplex/Semi-Detached properties are grouped together with Townhouses; together with Single Detached Houses the three categories form a single target class “house”.

The three target classes that were introduced are:

- Class 0: “condo”, including Apartments/Condos/Residence and Strata Townhouses
- Class 1: “house”, including Single Detached Houses, Duplex/Semi-Detached and Townhouses
- Class 2: “other”, including Commercial/Shopping, Mix (Commercial Residential), Industrial/Employment Lands, and everything else



(a) Duplex/Semi-Detached

(b) Townhouse

Figure 5.2: Distribution of price (in 2016 dollars) for two out of the three property types that are grouped together under Class 1: “house”. Both categories have similar distributions of price and records per coordinate pair between themselves and with Single Detached Houses, and thus all three categories are grouped together to form a single target class “house”.

Classification algorithms, a subcategory of algorithms for supervised learning, predict the discrete unordered categorical class labels of new instances based on past observations. A trained algorithm is capable of using a set of rules that were learned from past observations to distinguish the new instances between the possible classes. Once the classes have been determined and assigned, the target variable must be encoded, as it is considered good practice to provide class labels as integer arrays to algorithms to avoid technical glitches and improve computational efficiency. Class labels are not ordinal and classification estimators in the scikit-learn[38] machine learning library in Python treat them as categorical data that does not imply any order.

5.2 Missing values

As was mentioned in section 4.4, some of the new features engineered from Teranet attributes, such as 'xy_years_since_last_sale' and 'xy_years_to_next_sale' contain missing values in each case of the first or the last Teranet record coming from a coordinate pair. Since classification algorithms cannot handle inputs with missing values, during classification these values were replaced using the following logic:

- For all the records with missing 'xy_years_since_last_sale' (first record from a coordinate pair), values were replaced with the median 'xy_years_to_next_sale' for this subset (median time interval in years between all future sales from this Teranet subset).
- For all the records with missing 'xy_years_to_next_sale' (last record from a coordinate pair), values were replaced with the median 'xy_years_since_last_sale' for this subset (median time interval in years between all past sales from this Teranet subset).

This replacement affects a large number of Teranet records, as there are many coordinate pairs with a low count of transactions in the dataset. At the same time, it allows the classification algorithm to work on the majority of Teranet data, which improves the generalization of the algorithm and allows classifying land use of most Teranet records. As will be discussed in chapter 6, features that have missing values also have strong predictive power for land use classes and thus are best kept in the input set with the missing values replaced. This replacement is done during the fitting of classification algorithm and is not recorded in the final version of Teranet's dataset with new features and keys, where the missing values are left as missing.

5.3 Dimensionality reduction

Quality of the data plays a critical part in the success of application of a machine learning algorithm, and one of important aspects of data quality is the dimensionality of input space. The input space may contain features that are either redundant or irrelevant; highly correlated features also introduce multicollinearity and can make the model unstable, or too sensitive to small changes in the input data. In machine learning, statistics, and information theory, dimensionality reduction is the process of reducing the number of random variables under consideration; there is a number of reasons for reducing dimensionality of a dataset:

1. reducing the number of features improves computational efficiency and reduces training times
2. in case of a low signal-to-noise ratio in the dataset, dimensionality reduction can improve the predictive performance of an algorithm[40]
3. simpler models are easier to interpret[20]

4. excessive complexity of the model could cause overfitting[40]
5. the curse of dimensionality[5] (in the context of machine learning, the curse of dimensionality describes the phenomena where the feature space becomes too sparse for the size of the training dataset)[40]

There is a number of approaches that could be utilized to reduce the dimensionality of the feature space. There are two main categories of dimensionality reduction techniques:

- feature selection, also referred to as Feature Subset Selection, or (FSS), where a subset of the original features is selected
- feature extraction, where a new feature subspace is constructed from information derived from the original feature set

Feature selection can either be performed manually or by utilizing a feature selection algorithm. Exhaustive evaluation of all possible feature subsets is computationally unfeasible even for a moderate number of features. For example, if we are given a feature set with $m = 64$ features and want to reduce it to $n = 11$, an exhaustive evaluation would involve over 10^{11} possible feature subsets:

$$_mC_n =_{64} C_{11} = \binom{64}{11} = \frac{64!}{11!(64-11)!} = 743,595,781,824 \tag{5.1}$$

Feature selection algorithms present a practical approach to feature selection at scale; such algorithms combine a search strategy for proposing new feature subsets with an objective function to evaluate these subsets; objective function plays the role of a feedback signal used by the search strategy to choose between candidate subset. Objective functions are divided into three major groups:

- filters
  - evaluate candidate feature subsets by their information content (e.g., inter/intra class distance, mutual information, etc.)
  - advantages: have faster execution time since there generally is no iterative computation; have good generality since they evaluate the intrinsic properties of the data.
  - disadvantages: tend to select large feature subsets due to monotonic objective functions
- wrappers
  - use a classifier to evaluate subsets by their predictive accuracy
  - advantages: have better accuracy since they select subsets based on specific interactions between the classifier and the dataset
  - disadvantages: slower to execute since a classifier needs to be re-trained multiple times; selected feature subset will be specific to the classifier that was used to evaluate the candidate subsets.
- embedded methods
  - feature selection is performed as a part of model construction process (i.e., LASSO method for constructing a linear model with L1 regularization)[24]

A combination of different FSS techniques was applied to reduce the dimensionality of the dataset with Teranet, TTS, and Census variables for classification from 64 to 11 input features. The first method utilized was SelectFromModel in scikit-learn, a wrapper FSS algorithm that selects an optimal size and composition of a feature subset by fitting the provided classifier to training data and getting the importance weights from the fit model. SelectFromModel with Random Forest classifier has determined the optimal number of features to be 18; this number of features was provided to the other FSS algorithms to select the 18 best features from the dataset according to each method.

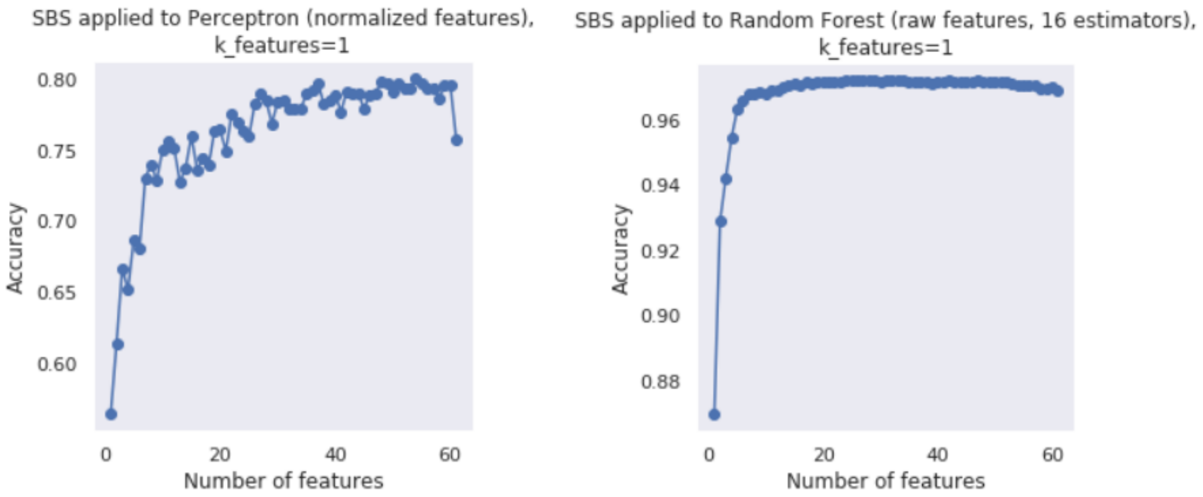
The second FSS method that was applied to the augmented Teranet dataset comes from the filter family of objective functions: a univariate feature selection algorithm SelectKBest in scikit-learn; this method selects k highest scoring features based on their scores on the specified univariate statistical test. Univariate FSS techniques can be useful for better understanding of the data, as they examine each feature individually to determine the strength of its relationship with the target variable. However, due to this fact, these methods will not necessarily result in improvement in the generalization of a learning algorithm, and thus their results have been used for feature selection in combination with other methods discussed in this section. The following statistical tests from scikit-learn have been selected for scoring features using SelectKBest FSS algorithm:

- Chi-squared stats of non-negative (pre-normalized) features for classification tasks
- ANOVA F-value between label/feature for classification tasks
- Mutual information for a discrete target

The third family of FSS algorithms that were applied were the recursive feature elimination algorithms, wrapper algorithms that evaluate candidate feature subsets and recursively eliminate features until the desired number of dimensions is reached. RFE class in scikit-learn facilitates feature ranking with recursive feature elimination using the provided external classifier that assigns weights to features. A more exhaustive approach to recursive feature elimination is the use of a sequential feature selection algorithm. Sequential feature selection algorithms are a family of greedy search algorithms that are used to reduce an initial d-dimensional feature space to a k-dimensional feature subspace where $k < d$.

Sequential Backward Selection (SBS) is a classic sequential FSS algorithm which aims to reduce the dimensionality of the initial feature subspace with a minimum decay in performance of the classifier; in some cases of model overfitting, SBS can even improve the predictive power of a classifier. On each iteration, a feature is removed using the defined criterion function J that we want to minimize; J can simply be defined as the difference in performance of the classifier before and after the elimination of a particular feature. Figure 5.3 presents plots produced by running SBS algorithm on augmented Teranet dataset with Perceptron and Random Forest; Sequential Backward Selection algorithm implemented as a Python class by Raschka and Mirjalili[40] has been used for this thesis.

As can be seen from the plots produced by SBS, most features in the augmented Teranet dataset can be eliminated without a significant drop in performance of both linear and tree-based models. As such, the final set of features to be used for land use classification was selected by combining the feature subsets selected by each of the FSS methods described above. Features that have been selected by at least four different methods were selected as the final feature subset that was tested with the classification algorithms and used for final classification of land use; figure 5.4 presents the 11 features that were selected using this logic: these are the 11 features that were selected by at least four different FSS techniques.



(a) Perceptron

(b) Random Forest

Figure 5.3: Sequential Backward Selection (SBS) is a classic sequential FSS algorithm; on each iteration, it eliminates the feature that results in the minimum drop in performance of the provided classifier. Figure 5.3a shows SBS run on augmented Teranet dataset with Perceptron and figure 5.3b shows SBS with Random Forest. It can be seen that many features can be eliminated without a significant drop in performance of both classifiers, especially in the case of Random Forest.

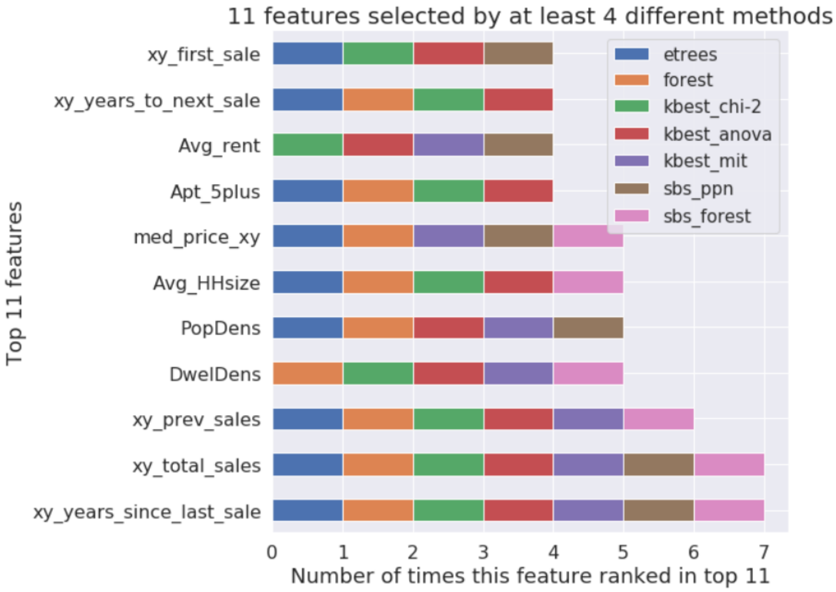


Figure 5.4: FSS results: 11 features that were selected by at least four different feature selection methods. These features were used to select the best-performing model and make the final classification of land use of Teranet records.

From the perspective of embedded feature selection methods, another possible approach to reduce the complexity of a model is to use L1 regularization to penalize large individual weights. L1 regularization introduces a penalty term to the cost function which is taken as the norm of the weight vector defined as:

$$L1: \ \|\boldsymbol{w}\|_1 = \sum_{j=1}^{m} |w_j| \tag{5.2}$$

L1 regularization is similar to L2 regularization, which is defined as:

$$L2: \ \|\boldsymbol{w}\|_2^2 = \sum_{j=1}^{m} w_j^2 \tag{5.3}$$

where m is the number of features.

However, since in the case of L1 regularization, the sum of squares of weights is replaced with the sum of the absolute values of weights, L1 regularization usually yields sparse feature vectors, since most feature weights will be zero[40, 24]. Sparsity of feature vectors can help us get rid of irrelevant features in a high-dimensional dataset; in this context, L1 regularization can be understood as a technique for feature selection[40]. L1 regularization was attempted with Logistic Regression and Linear Support Vector Classifier models in scikit-learn, but did not result in a significant improvement in model performance.

5.4 Tuning model hyperparameters

One of the critical aspects of evaluating the performance of a machine learning model is the assessment of the algorithm's ability to not only perform well on the data that was used to train it, but also to generalize well to unseen, or test, data. Evaluating the model on the same dataset that was used to train it will result in the optimistically biased estimate of the predictive power of a classifier; hence the need to split the dataset into separate training and test subsets. Comparing predictions made by the algorithm to true labels in the test subset can be understood as the unbiased performance evaluation of the model[40]; in this context, bias refers to the difference between the expected prediction accuracy of a model and its true prediction accuracy[39].

To facilitate this, all GTHA Teranet records from 2011 to 2014 have been split into two subsets using random subsampling: 70% of the data was used to train models and tune their hyperparameters, while 30% of the data has been used as a test subset for the unbiased performance evaluation of the classifier. Train and test subsets have been stratified across the target classes: in this context, stratification means that training and test subsets will have the same proportions of class labels as the input dataset.

There are two types of parameters in machine learning: those that are learned by parametric models from training data (i.e., weights in logistic regression), and the parameters that tune the performance of a learning algorithm, or its hyperparameters (i.e., regularization parameter in logistic regression or maximum depth of a decision tree). If a model is too simple, it can suffer from underfitting and show poor performance on both the training and the test data (have high bias; the bias of a learning algorithm is the persistent error that it is expected to make when trained on a given sample size[15]). On the other hand, if a model is too complex, it can overfit the training data and have poor generalization on test data (have high variance; the variance captures the random variation in the algorithm from one training set to another[15]). An acceptable bias-variance trade-off can be found by tuning the hyperparameters of a learning model, but care must be taken to ensure unbiased assessment of its generalization performance.

There are no hard-and-fast rules that guarantee the best performance of a classifier on a given dataset[39]; thus, it becomes important to evaluate the performance of a classifier with different hyperparameters to find the best model settings prior to using the model to make predictions. However,

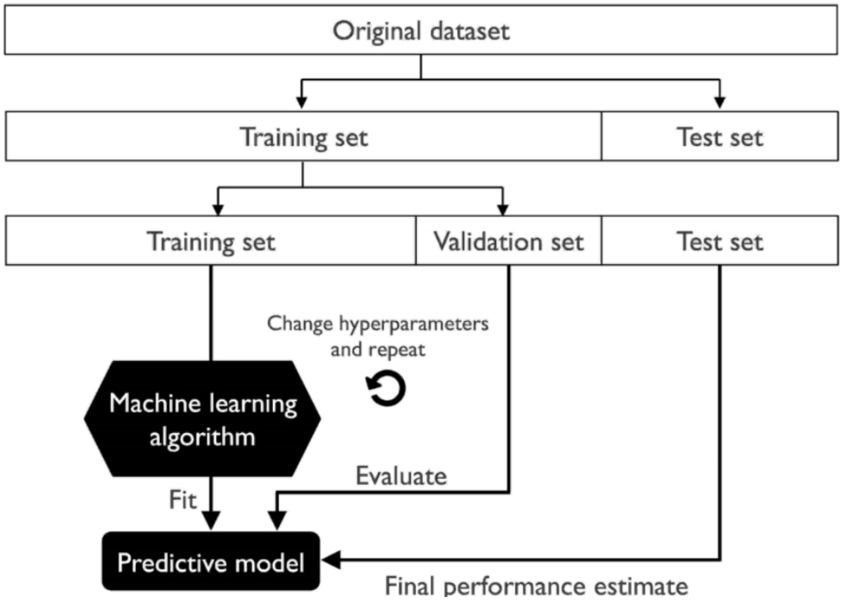


Figure 5.5: Holdout validation of a machine learning model as summarized by Raschka and Mirjalili[40].

when evaluating different hyperparameters for estimators, there is a risk of overfitting on the test set because knowledge about the test set can "leak" into the model; evaluation metrics will no longer report on generalization performance. To address this issue, yet another part of the dataset can be held out as a so-called "validation set": training proceeds on the training set, with model hyperparameters being evaluated by its performance on the validation set; when the experiment seems to be successful, final evaluation can be done on the test set. Figure 5.5 presents an example of a workflow for tuning and evaluating a machine learning model using hold-out validation, as summarized by Raschka and Mirjalili[40].

There is a shortcoming in this approach: by partitioning the available data into three sets, the number of samples which can be used for learning the model is drastically reduced; results can depend on a particular random choice of the train and validation sets. A solution to this problem is a procedure called cross-validation (CV for short): validation set is produced from the training set using such techniques as k-fold CV, while the test subset of data unseen by the model is held out for its final evaluation. In a basic approach called k-fold CV, the training set is split into k smaller sets and on each of k iterations a model is trained using $k-1$ of the folds as training data; the resulting model is validated on the held-out test fold of the data through such performance metrics as model accuracy. The performance measure reported by k-fold cross-validation is then the average of the values computed in the loop. Figure 5.6 presents k-fold cross-validation workflow to evaluate model performance as summarized by Raschka and Mirjalili[40].

This approach can be computationally expensive, but does not waste too much data, as is the case when fixing an arbitrary validation set. Empirical evidence shows that a good standard value for k in k-fold cross-validation is 10, as seen in experiments conducted by Kohavi on various real-world datasets[22]. A slight further improvement in bias and variance estimates over the standard k-fold cross-validation can be achieved by using stratified k-fold cross-validation, as shown in the study conducted by Kohavi. In stratified cross-validation, the class proportions are preserved in each fold to ensure that each

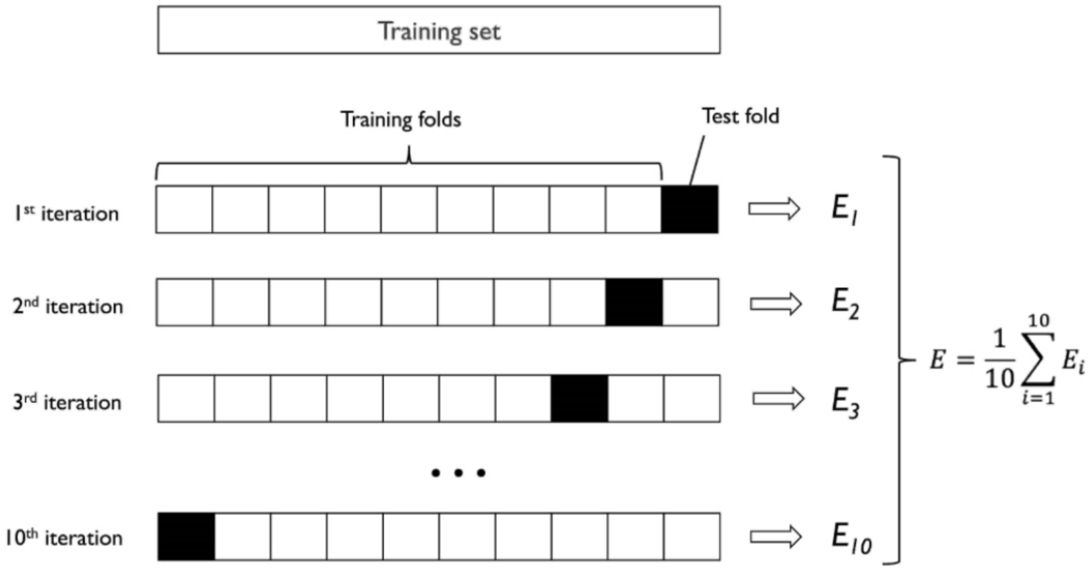


Figure 5.6: k-fold cross-validation as summarized by Raschka and Mirjalili[40]. In the case of $k = 10$, the training dataset is divided into 10 folds, and during the 10 iterations, nine folds are used for training, and one fold is used as the test set for the model evaluation. Estimated performances (for example, classification accuracy or error) for each fold are then used to calculate the estimated average performance E of the model.

fold is representative of the class proportions in the training dataset. For this thesis, stratified k-fold cross-validation workflow has been used to tune model hyperparameters via grid search before the final evaluation of model performance was done using the held-out test set; model evaluation results will be discussed in chapter 6, examples of validation curves for decision tree and random forest are presented on figure 6.4.

5.5 Feature scaling

Many machine learning algorithms are designed with the assumption that each feature takes values close to zero or, more importantly, that all features vary on comparable scales[23]. Raw data rarely comes in the form and shape that is necessary for the optimal performance of a learning algorithm, and thus data preprocessing via scaling is one of the most crucial steps in machine learning workflows[40]. Bringing different variables to the same scale can be accomplished by such techniques as standardization or normalization; in addition, outliers that are present in data can be addressed by using non-linear transformations, such as quantile or power transforms.

The following preprocessing methods have been tested on the training dataset:

- standardization (mean removal and variance scaling)
- normalization (min-max scaling to a range of [0, 1])
- max-abs (scale to a range of [-1, 1], maximum absolute value of each feature is scaled to unit size)
- robust scaling (more robust estimates for data with outliers)

- power transform (Yeo-Johnson, mapping to a Gaussian distribution)
- quantile transform (mapping to a Gaussian distribution)
- quantile transform (mapping to a uniform distribution)
- sample-wise L2 transform (normalize samples individually to unit norm)

All these preprocessing techniques were tested with classification algorithms for prediction accuracy and fit times; results will be presented in chapter 6. It is important to note that the parameters for the previously mentioned procedures, such as feature scaling and dimensionality reduction, are solely obtained from the training dataset, and the same parameters are later reapplied to transform the test dataset and the full Teranet dataset for the final classification of land use; this way, an unbiased estimate of model generalization can be obtained.

5.6 Model selection

An important point to be summarized from the famous No Free Lunch Theorems (NFL)[52, 53] by David H. Wolpert is that no single classifier works best across all possible scenarios, as there is a lack of a priori distinctions between learning algorithms. In practice, it is essential to compare the performance of at least a handful of different classification algorithms, since each of them has its inherent biases, in order to select the best performing model. No single classification model enjoys superiority if we don't make any assumptions about the form of the function that maps input features to target variables and how this function can be learned[40].

There are several possible ways of categorizing machine learning algorithms; one of them is to separate algorithms into parametric and non-parametric models. In parametric models, such as logistic regression, perceptron or linear Support Vector Machine (SVM), model parameters are estimated from the training set to learn a function that allows classifying new data without the need for the original training set, once the model has been fit. In addition, the number of model parameters is fixed and does not change with the size of the training data. In contrast, non-parametric models, such as instance-based learning models like k-Nearest Neighbors, can't be characterized by a fixed set of parameters, and the number of parameters grows with the amount of training data.

Another way of categorizing the algorithms that were used in this thesis is into linear, tree-based, and nearest neighbors model classes. Models from the following three main classes of classification algorithms have been tested with the augmented Teranet dataset:

- Linear models

  Linear models are used by a popular and broad class of procedures for solving classification tasks. These models aim at dividing the feature space into a collection of regions labelled according to the values that the target can take, where the decision boundaries between those regions are linear hyperplanes. Examples include:

  - Perceptron learning algorithm

    Perceptron is a simple classification algorithm suitable for large-scale learning that was introduced by Frank Rosenblatt[42] in 1957 based on the McCulloch-Pitts (MCP) artificial neuron model[28] that was introduced in 1943. It is a parametric non-regularized model that only updates its weights on wrong predictions that it makes.

- Logistic Regression (L2, L1 regularization)

  Another simple yet more powerful parametric algorithm for linear and binary classification problems is logistic regression; it can also be expanded to multi-class problems through such techniques as One-versus-All (OvA). Logistic regression models the probabilities of an observation belonging to each of the K classes via linear functions and is typically estimated by maximum likelihood. It is often used as an inference tool to understand the role of input variables in explaining the target, since it produces easily interpretable coefficients; in addition, it can also have significant predictive power in the cases when target classes are linearly separable[40].

- Linear Discriminant Analysis

  Linear Discriminant Analysis (LDA) is most commonly used as a dimensionality reduction technique, but it can also have some practical uses as a classifier by itself. LDA assumes that the joint densities of all features given target's classes are multivariate Gaussians with the same covariance for each class.

- Quadratic Discriminant Analysis

  Quadratic Discriminant Analysis (QDA) is a classifier with a quadratic decision boundary; it relaxes the common covariance assumption of LDA through estimating a separate covariance matrix for each class.

- Linear Support Vector Classification (L2, L1 regularization)

  Support Vector Machines (SVM) represent another family of powerful and widely used algorithms used for maximum margin classification: optimization objectives for SVMs is to maximize the margin instead of minimizing the classification error. The margin is defined as the distance between the separating hyperplane (decision boundary) and the training samples that are closest to this hyperplane, which are the so-called support vectors[40].

- Naive Bayes Classifier

  A Naive Bayes Classifier is a probabilistic classification technique based on Bayes Theorem with an assumption of independence among input features. It determines the probability that an example belongs to some class, calculating the probability that an event will occur given that some input event has occurred. These algorithms are computationally efficient and suitable for large datasets, but have the disadvantage of the assumption about the independence of input features.

- Tree-based models

  Decision tree classifiers are attractive models due to their predictive power, computational efficiency and interpretability. In addition, these models are scale-invariant and require less data preprocessing. These models construct a flow-chart-like decision tree by learning simple decision rules similar to asking a series of questions, or in other words, introduce a set of boolean conjunctions. Examples include:

  - Decision Tree Classifier

  Decision trees can build complex decision boundaries by dividing the feature space into rectangles by learning a series of questions to infer the class labels of the samples; questions are selected to maximize Information Gain (IG) on each split of the tree, defined as the difference in selected impurity measure (Gini, entropy, or classification error) between parent and child nodes. Splitting procedure is repeated until all the leaves are pure (consist of samples belonging to the same target class), or until selected maximum depth of the tree is reached.

  - Random Forest Classifier

    High variance plays a more important role in poor performance of decision trees rather than high bias; thus, ensemble methods based on voting can improve their performance since voting methods are capable of reducing variance of a learning algorithm[15]. Decision trees can be unstable due to small variations in the data, and a more robust approach is to combine multiple trees into an ensemble model. Random forest is a hugely popular classification algorithm due to its good classification performance, scalability, and ease of use[40]. Random forest averages multiple (deep) decision trees each of which individually suffers from high variance, to build a model with better generalization performance and less susceptible to overfitting. Prediction of the class label is aggregated by a majority vote from the selected number of trees, each of which is trained on a bootstrapped sample of records using a randomly selected subset of features.

- Nearest Neighbors

  Neighbors-based classification presents a type of instance-based non-parametric learning: it does not attempt to construct a discriminative function to classify the target variable; instead, it simply stores instances of the training data. The principle behind nearest neighbors is to find a predefined number of training samples closest in distance to the new point, and predict the target label of a new sample using the majority vote of its neighbors. The number of samples can either be a user-defined constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning). Commonly used distance metrics include Manhattan and Euclidean distance.

  - k-Nearest Neighbors Classifier

    k-Nearest Neighbors classifier is a distance-based classification algorithm that uses a predefined number of samples considered as neighbors to each new point according to the selected distance metrics (Manhattan, Euclidean distances are a common choice). Class label of each new point is assigned by the majority vote of its neighbors. Number of neighbors k and distance metric are the hyperparameters of a k-NN model.

In order to compare different models, metrics to measure performance needed to be established, which will be discussed in section 6.1. Results of model evaluation are presented in chapter 6.

5.7 Chapter summary

One of the major features missing from Teranet's dataset is land use information covering a period of time starting from 1985. This chapter introduced a prototype of machine learning workflow to classify land use at Teranet transaction level from housing market dynamics as reflected by augmented Teranet variables. Such topics as methodology for feature selection, filling of the missing values, feature scaling, model hyperparameter tuning, unbiased assessment of model performance via such techniques as train-test split and k-fold cross-validation and selection of learning algorithms to be tested have been discussed in this chapter. Chapter 6 presents the evaluation of model performance.

Chapter 6

Model evaluation

The fifth phase of the CRISP-DM process model for data mining projects is evaluation; this chapter presents the evaluation of results of the machine learning workflow to classify land use that was introduced in chapter 5; classification algorithms described in section 5.6 were tested on the augmented Teranet dataset, production of which was described in chapter 4. As was discussed in section 5.6, no single classifier works better than others across all possible scenarios, and thus it is important to compare the performance of several classification algorithms on a given dataset; this chapter presents the evaluation of predictive and computational performance of machine learning algorithms used for classifying land use from the housing market dynamics.

6.1 Metrics for evaluating model performance

Before different models can be compared, metrics to measure model performance and model acceptance criteria must be discussed first.

- 0–1 loss and prediction accuracy

  Both the prediction error (ERR) and accuracy (ACC) provide general information about how many samples are misclassified by a given model. Classification accuracy (ACC) is a common metric used to compare the performance of different classifiers; it is defined as the proportion of correctly classified instances:

  $$ACC = 1 - ERR = \frac{\text{Correct classifications}}{\text{Total classifications}} \tag{6.1}$$

  Another way to define this characteristic is the prediction error, ERR, which can be computed directly from classification accuracy:

  $$ERR = 1 - ACC = \frac{\text{Incorrect classifications}}{\text{Total classifications}} \tag{6.2}$$

  In addition, prediction error can be computed as the expected value of the 0–1 loss over n samples in a given dataset S:

$$ERR_S = \frac{1}{n}\sum_{i=1}^{n} L(\hat{y}_i, y_i) \tag{6.3}$$

In mathematical optimization and decision theory, a loss function is a function that maps event values onto a real number representing some ”cost” associated with an undesired event. 0–1 loss $L(\cdot)$ is a loss function frequently used in statistics and decision theory, and can be defined as:

$$L(i,j) = \begin{cases} 0 & \text{if } \hat{y}_i = y_i \\ 1 & \text{if } \hat{y}_i \neq y_i \end{cases} \tag{6.4}$$

The 0–1 loss function is equivalent to classification accuracy, since the only aspect being considered is if model predictions match the true target labels or not.

- Precision and recall

  When performing binary classification predictions (multi-class methods can be extended from binary classification using such techniques as One-versus-All, or OvA), there’s four types of outcomes that could occur:

  - True positives (TP): observation belongs to a class and was correctly predicted to belong to it.
  - True negatives (TN): observation does not belong to a class and was correctly predicted to not belong to it.
  - False positives (FP): observation does not belong to a class, but was incorrectly predicted to belong to it.
  - False negatives (FN): observation belongs to a class, but was incorrectly predicted to not belong to it.

  These four outcomes can be combined into a confusion matrix, which lays out the performance of a learning algorithm. Two important binary classification metrics can be derived from a confusion matrix: precision and recall.

  Precision is defined as a fraction of relevant examples (true positives) among all of the examples that were predicted to belong in a certain target class:

$$PRE = \frac{TP}{TP + FP} \tag{6.5}$$

  Recall is defined as a fraction of examples which were predicted to belong to a class (true positives) with respect to all of the examples that truly belong to that class.

$$REC = \frac{TP}{FN + TP} \tag{6.6}$$

  In case of a multi-class classification problem, these metrics can be produced using individual confusion matrices constructed separately for each class (OvA).

- F1 score, also known as balanced F-score or F-measure

  In practice, precision and recall are often combined into a single metric known as the F1-score:

  $$F1 = 2\,\frac{PRE \times REC}{PRE + REC} \tag{6.7}$$

- Scoring metrics for multi-class classification

  Precision, recall, and F1 score are metrics specific to binary classification systems. However, their application can be extended to multi-class problems via One-versus-All (OvA) classification and averaging techniques.

  Micro-average is calculated from the individual TPs, TNs, FPs, and FNs of classification system with k classes:

  $$PRE_{micro} = \frac{TP_1 + \cdots + TP_k}{TP1 + \cdots + TP_k + FP_1 + \cdots + FP_k} \tag{6.8}$$

  Macro-average is calculated as the mean of scores for each class:

  $$PRE_{macro} = \frac{PRE_1 + \cdots + PRE_k}{k} \tag{6.9}$$

  Micro-averaging can be useful to weigh each instance or prediction equally, while macro-averaging evaluates the overall performance of a classifier with regard to the most frequent class labels by weighting all classes equally[40].

  The weighted macro-average is calculated by weighting the score of each class label by the number of true instances when calculating the average. The weighted macro-average is useful when dealing with class imbalances, that is, different numbers of instances for each label.

As Raschka states in his article[39], model evaluation is a complex topic by itself; when asking the question of which evaluation metric to select for assessing model performance, a more meaningful question to ask would be why does the assessment of these models matter at all. Under the current understanding of the problem of land use classification, the ideal model would be able to provide accurate classification of land use of the maximum possible number of properties across all classes, or in our case, land use for each Teranet record coming from a location; an ideal model would correctly classify every Teranet transaction by its land use. Under the current formulation of the problem, no special emphasis was put on the correct classification of any particular property type (as reflected by the three target classes "house", "condo", or "other") or on any particular kind of classification error. Therefore, the model that can classify the most records correctly would be considered the best performing model for the purposes of this thesis.

Since target classes introduced in section 5.1 are fairly balanced, classification accuracy (ACC) presents an acceptable metric to evaluate model performance; other classification metrics discussed in this section could be used for additional reference regarding the details of model performance. Classification errors made on particular target classes can be assessed by class-wise precision and accuracy metrics, as found on the classification report shown on figure 6.6; in addition, confusion matrices can provide additional information on class-wise errors made by the model (confusion matrices are presented on figure 6.8). In addition, it would be beneficial if the model can provide indication of the relative importance of features in making accurate land use predictions through such outputs as feature importance coefficients.

6.2 Evaluating model performance

Model hyperparameters were tuned for each classifier via grid search and k-fold cross validation, as was described in section 5.4; tuning was validated via validation curves produced for each of the hyperparameters. After the tuning, performance of the following models have been compared:

- Perceptron learning algorithm, learning rate $\eta = 0.5$, maximum iterations=5, features transformed with quantile transformation (uniform PDF)

  model code: ppn_qu_eta0.5_maxiter5

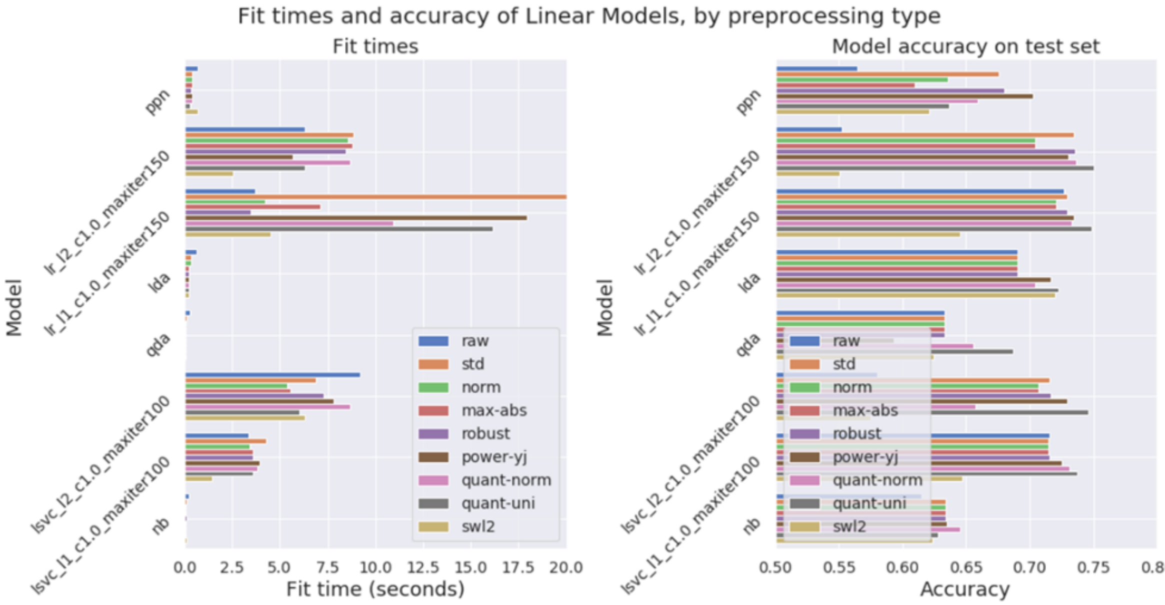


Figure 6.1: Fit times and accuracy of linear models, by feature scaling technique. Different scaling techniques have a strong effect on the performance of linear models.

- Logistic regression with L2 regularization, regularization parameter $C = 0.1$, maximum iterations=100, features transformed with quantile transformation (uniform PDF)

  model code: lr_l2_c0.1_maxiter_100

- Logistic regression with L1 regularization, regularization parameter $C = 0.1$, maximum iterations=100, features transformed with quantile transformation (uniform PDF)

  model code: lr_l1_c0.1_maxiter_100

- Linear Discriminant Analysis Classifier, features transformed with quantile transformation (uniform PDF)

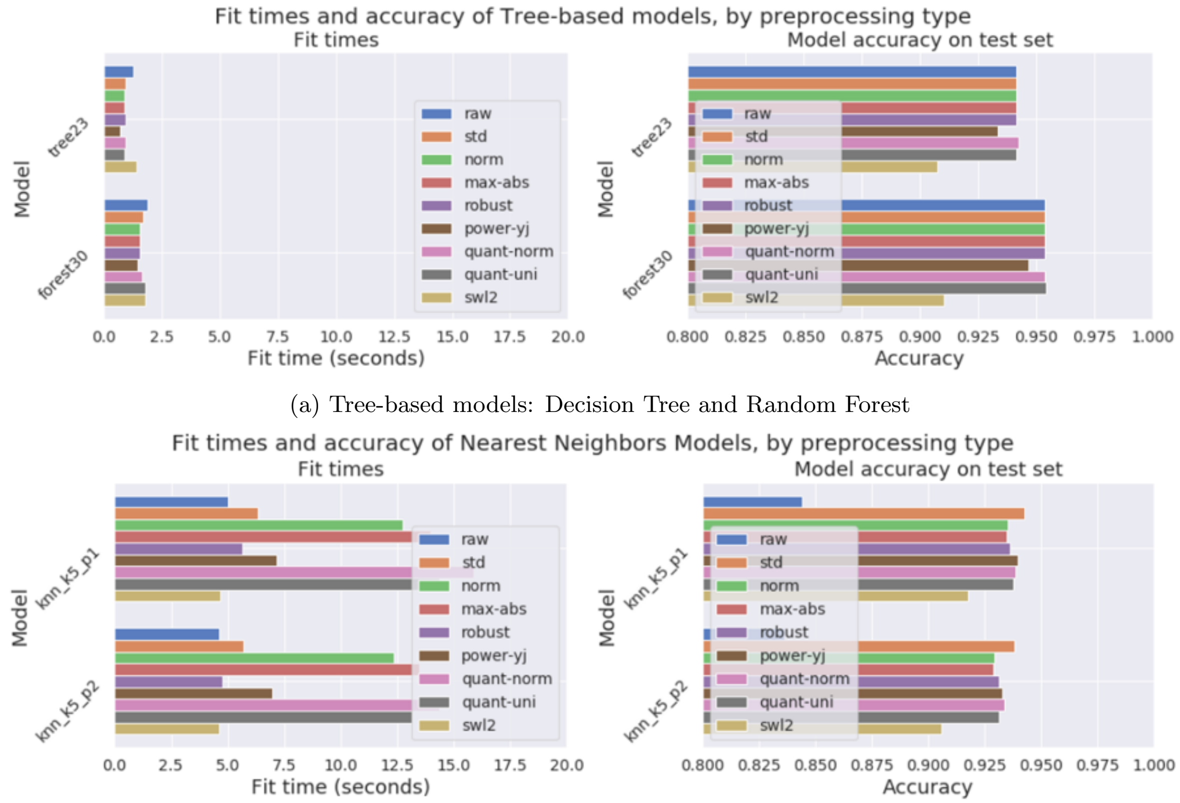
  model code: lda

- Quadratic Discriminant Analysis Classifier, features transformed with quantile transformation (uniform PDF)

  model code: qda

- Linear Support Vector Classification with L2 regularization, regularization parameter $C = 0.1$, maximum iterations=100, features transformed with quantile transformation (uniform PDF)

  model code: lsvc_l2_c0.1_maxiter_100



(a) Tree-based models: Decision Tree and Random Forest

(b) K-Nearest Neighbors: Manhattan and Euclidean distance

Figure 6.2: Fit times and accuracy for tree-based models and K-Nearest Neighbors, by feature scaling technique. Distance-based algorithms, such as K-Nearest Neighbors are affected by feature scaling, while tree-based models are scale-invariant.

- Linear Support Vector Classification with L1 regularization, regularization parameter $C = 0.1$, maximum iterations=100, features transformed with quantile transformation (uniform PDF)

  model code: lsvc_l1_c0.1_maxiter_100

- Gaussian Naive Bayes

  model code: nb

- Decision Tree, Gini impurity criterion, maximum depth of the tree=25, unscaled features

  model code: tree25

- Random Forest, Gini impurity criterion, number of estimators=50, unscaled features model code: forest50

- K-Nearest Neighbors, Manhattan distance metric, number of neighbors=4, standardized features

  model code: knn_p1_k4

Different feature scaling techniques have a strong effect on model fit times and predictive performance of linear models, as can be seen on figure 6.1. Similar to linear models, distance-based algorithms, such as K-Nearest Neighbors, are also strongly affected by feature scaling. In contrast, tree-based models, such as Decision Tree and Random Forest, are scale-invariant and thus require less preprocessing. Figure 6.2 presents fit times for Decision Tree, Random Forest, and K-Nearest Neighbors with Manhattan and Euclidean distance metrics.

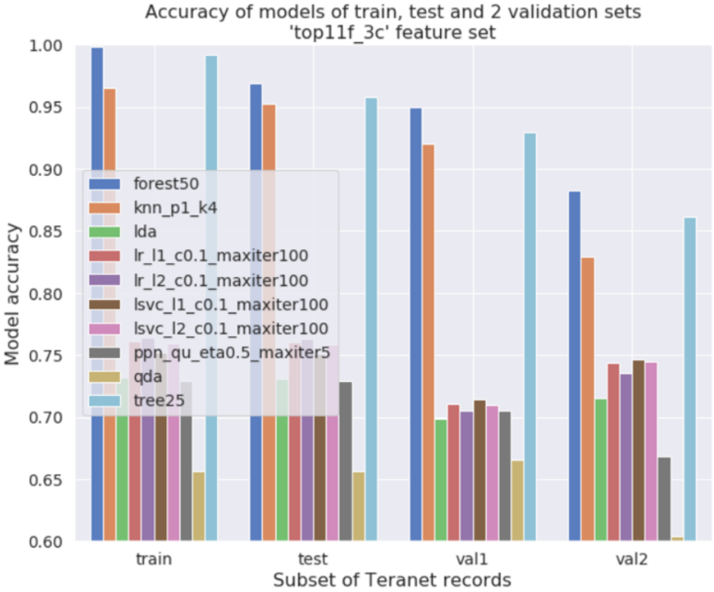


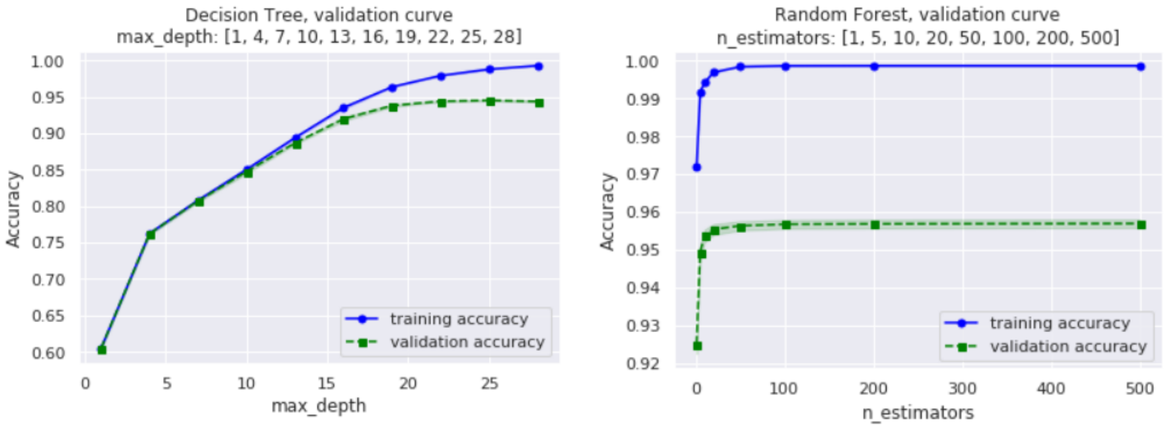
Figure 6.3: Model performance (accuracy) on train, test, and two additional validation subsets. Additional validation subsets are composed of Teranet records from 2010 and 2015; since land use information (target variable) can be less accurate for records in these subsets, they have been used as an additional reference for assessing the generalization of a learning algorithm.

Four different subsets of data have been used to test the best performing models: train, test, and two additional validation subsets. The train and test subsets represent Teranet records from 2011 to 2014 randomly sampled into 70% train and 30% test subsets; these are the primary subsets that were used for training and tuning the hyperparameters and then evaluating the performance of classifiers on unseen test data, as was described in section 5.4. The two additional validation subsets were composed of Teranet records from 2010 and 2015. Since the Department of Geography land use information (target variable) was collected in 2012 and 2013, it can be less accurate for these subsets; thus, they have not been used for model selection, training or primary testing, but were utilized to test fitted models as an additional reference for the generalization of performance of classifiers. Figure 6.3 presents model performance on train, test, and two additional validation subsets.

As can be seen on figure 6.3, in terms of prediction accuracy, tree-based and nearest neighbors models dramatically outperform linear models, such as logistic regression, linear SVC, perceptron and LDA classifier. This indicates that in the current feature space target classes are not linearly separable.

Best-performing linear models were able to reach classification accuracy around 75% on the test set, with all linear models performing close to each other. This performance is consistent across the train, test, and both extra validation subsets. Linear models do not seem to overfit the data, but instead suffer from high bias.

In comparison, tree-based models capable of drawing complex non-linear decision boundaries were able to achieve much higher classification accuracy, with both decision tree and random forest scoring above 95% on the test set. These models have a much lower bias on this dataset compared to linear models, but do overfit the training data to some degree under the current size of the training subset. Validation curves produced during hyperparameter tuning described in section 5.4 for decision tree by variating its maximum depth (shown on figure 6.4a) indicate that the model does need to be fairly complex to achieve better classification accuracy; it suffers from high bias until maximum depth of the tree reaches at least 15. Then, the model starts to pick up higher variance; all increases of maximum depth past 23 result mostly in overfitting the training data through unnecessary model complexity and do not yield better generalization. At a maximum depth of 23, the decision tree still has moderate variance, with predictions on the test set being 3% less accurate then those made on the training data. Overfitting of the tree also seems to be present on both extra validation subsets, with tree accuracy decreasing further on these subsets; however, it is challenging to understand whether this is due to the model overfitting training data or due to inaccuracies of target labelling in extra validation subsets.



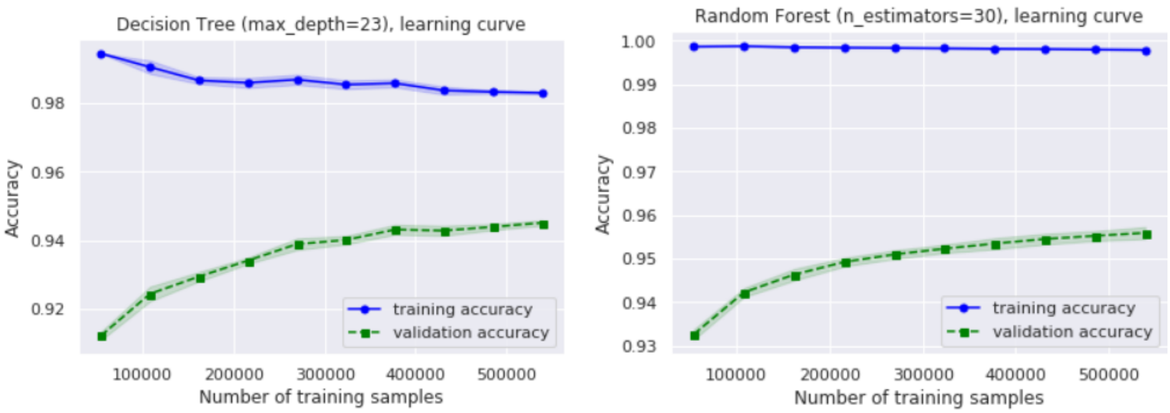
(a) Decision Tree (max_depth)

(b) Random Forest (n_estimators)

Figure 6.4: Validation curves for decision tree (max_depth) and Random Forest (n_estimators) that were produced during model hyperparameter tuning described in section 5.4.

Validation curves produced for random forest by varying the number of estimators (presented on figure 6.4b) show that the model also suffers from some variance and overfits training data. However, random forest still shows slightly higher prediction accuracy when compared to a single decision tree, and performs better on extra validation subsets. Since the inaccurate target labelling in extra validation subsets would affect both decision tree and random forest equally, even if it provides a pessimistically biased estimate of algorithm generalization, it can still be used to compare the models between themselves[39]. This indicates that random forest should have better generalization than the decision tree.

Learning curves produced for decision tree and random forest by variating the size of the training set (shown on figure 6.5) also confirm overfitting of both models. Their learning curves indicate that variance of both models could be reduced further if more data would be used for training.



(a) Decision Tree

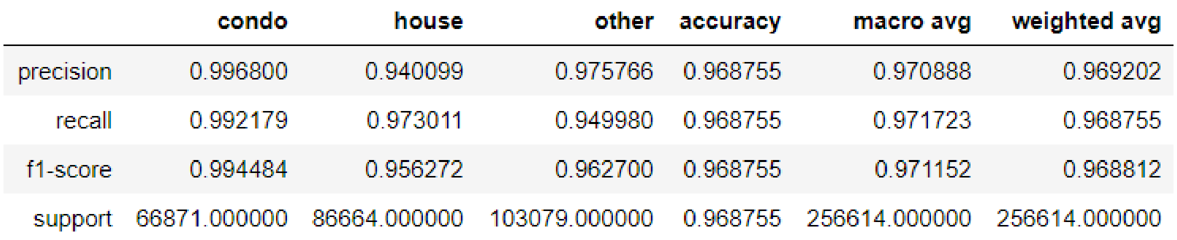
(b) Random Forest

Figure 6.5: Leaning curves for decision tree and random forest that were produced by variating the size of the training dataset. It can be seen that both models overfit training data less with the increase of the training set; variance of both models could be reduced further if more data would be used for training.

k-Nearest Neighbors also performs well on test data, with an accuracy slightly lower than that of tree-based models. It seems to have a low variance between training and test set, but does show stronger decay of performance on both extra validation subsets. In addition, k-NN requires additional preprocessing steps and has much longer fit times on larger datasets when compared with tree-based models; the computational cost of using a tree to predict data is logarithmic in the number of points used to train the tree.

6.3 Best performing model: Random Forest

As can be seen from the plots presented in section 6.2, random forest with 50 estimators and Gini impurity criterion showed the best results in terms of accuracy and fit times on all subsets. Figure 6.6 presents the classification report showing all main model performance metrics for the best performing model: random forest with 50 estimators using Gini impurity criterion.



	condo	house	other	accuracy	macro avg	weighted avg
precision	0.996800	0.940099	0.975766	0.968755	0.970888	0.969202
recall	0.992179	0.973011	0.949980	0.968755	0.971723	0.968755
f1-score	0.994484	0.956272	0.962700	0.968755	0.971152	0.968812
support	66871.000000	86664.000000	103079.000000	0.968755	256614.000000	256614.000000

Figure 6.6: Best model performance on test set: classification report for Random Forest with 50 estimators using Gini impurity criterion.

Figure 6.7 presents feature importance for the best performing model. It can be seen that the final random forest model had a fairly balanced use of selected new Teranet and Census variables; new attributes engineered from native Teranet features that were introduced in section 4.4 have strong predictive power over land use classes. It can be seen that the new attributes that relate to the frequency of Teranet records coming from a coordinate pair and new attributes related to time interval between

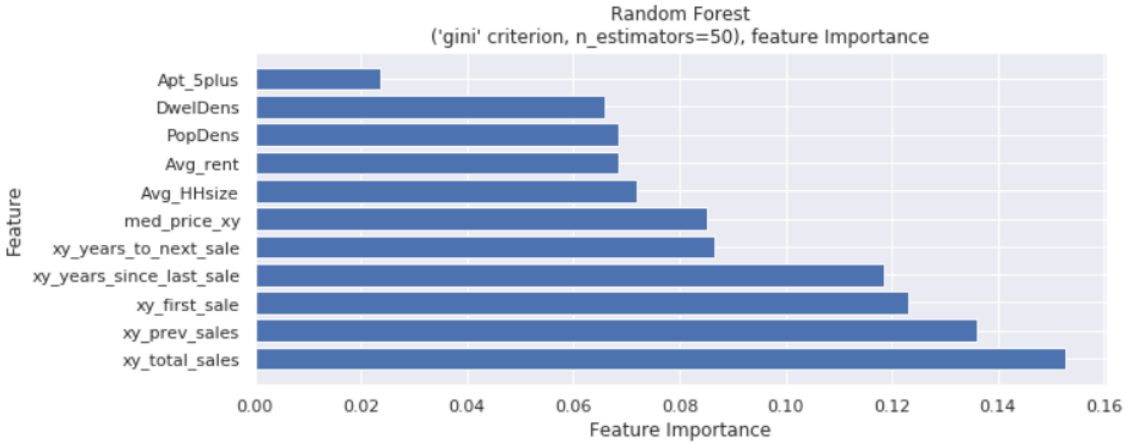


Figure 6.7: Random Forest with 50 estimators using Gini impurity criterion: feature importance.

the records from a coordinate pair had the highest importance coefficients for the random forest model. Census variables that were used to augment the Teranet dataset based on relationships that were discussed in chapter 3 are also related to land use classes and improve the classification accuracy of the model.

Confusion matrices that were introduced in the context of a binary classification problem in section 6.1 can be extended to plot multi-class classification predictions. Figure 6.8 presents confusion matrices with and without normalization for Random Forest on the test set.

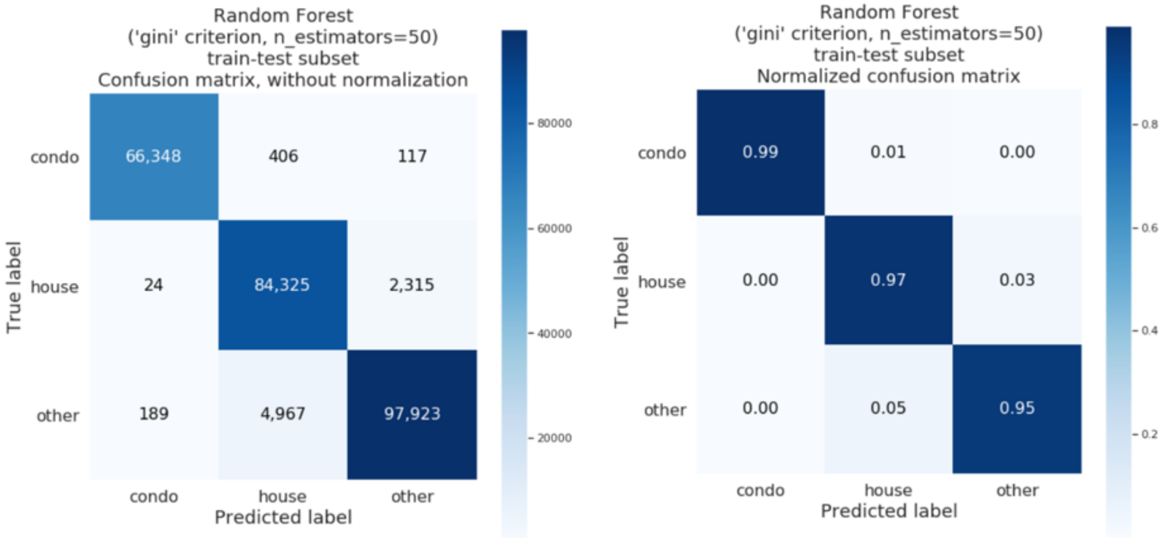


Figure 6.8: Best model performance on test set: confusion matrices with and without normalization for Random Forest with 50 estimators using Gini impurity criterion. It can be seen that despite some imbalance in classes, the model seems to perform consistently over all three land use categories.

It can thus be concluded from the classification report (shown in figure 6.6) and confusion matrices (shown in figure 6.8) that the model is accurate in its predictions for all three target classes; moderate class imbalance does not seem to affect accuracy of random forest in classifying the three reduced land use categories. Thus, this machine learning workflow is capable of producing classification of land use into three major categories ("house", "condo", and "other") for each Teranet record starting from 1985; unbiased performance estimate done on 30% of held-out test data show 97% of accuracy for random forest with 50 estimators and Gini impurity criterion. With further refinement of this workflow, land use can be classified for each Teranet record using newly engineered features and joined Census and TTS variables with a high degree of accuracy.

To facilitate further evaluation of the results produced by this thesis and allow ease of access to spatially and temporally related dataset introduced in chapter 3, the augmented dataset that was produced has been saved into a PostgreSQL relational database; land use produced for each Teranet record by the classification algorithm has been added to augmented Teranet dataset in the database as new feature 'lucr_predict'. This way, ease of access and retrieval of this information can be facilitated, as required by the needs of the Longitudinal Analysis of Housing Sales in the GTHA which was introduced in section 2.4; in addition, results of this machine learning workflow could be further investigated by a broader and mixed team of specialists. An Entity Relationship (ER) diagram for the database created as a part of this master's thesis is pressented in Appendix A.

To provide initial EDA of classification errors, counts of Teranet records misclassified by the algorithm by DA have been produced and mapped for the second extra validation subset from 2015, which was showing higher error rates compared to other subsets. Choropleth (equal interval split of the distribution) map of counts of classification error can be seen on figure 6.9. It can be seen that classification errors are highly localized and likely represent high-frequency transactions corresponding to condo units and mixed use properties.

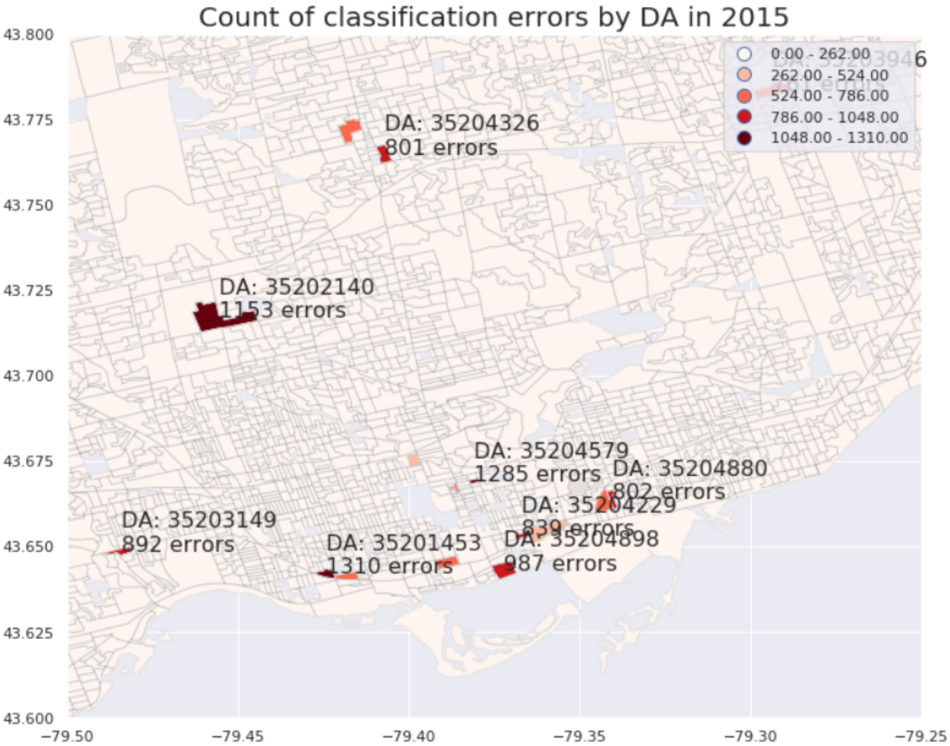


Figure 6.9: Choropleth (equal interval) map of count of misclassified Teranet records by DA. It can be seen that errors are highly localized.

6.4 Chapter summary

This chapter discussed metrics commonly used to assess performance of classification algorithms, selected criteria for model assessment and acceptance and presented results of model evaluation. Tree-based models and k-Nearest Neighbors significantly outperformed linear models in accuracy; tree-based models outperformed k-NN in fit times. Learning and validation curves were presented and discussed for tree-based models. Random forest with 50 estimators was the best performing model and achieved 97% accuracy on the test set. Tree-based models have low bias, but do suffer from some overfitting and could benefit from further increase in the size of the training set. Overall, the machine learning workflow introduced by this thesis seems to be able to classify land use of Teranet records based on selected features with a high degree of accuracy. The map of count of classification errors by DA produced from Teranet records from 2015 shows that these errors are highly localized and likely correspond to high-frequency transactions, such as condo units and mixed use properties. Further investigation of the results of land use classification is required and has been facilitated by transforming the augmented Teranet dataset and other related datasets produced for this master's thesis into a PostgreSQL relational database; land use predicted by the classification algorithm has been added as a new feature 'lucr_predict' to each Teranet record within the time span of Census / TTS variables. An Entity Relationship (ER) diagram for the database created as a part of this master's thesis can be found in Appendix A; its referential integrity constraints were implemented based on the spatial and temporal relationships between data sources that were introduced in chapter 3.

Chapter 7

Conculsion

Microsimulation models present the latest generation of integrated land use and transportation models and are well suited to analyze the complex interaction of transportation and land use. New data sources that appear with the increased digitization of human activity present opportunities to look at urban processes at unprecedented spatial and temporal scale, and thus present a lot of value for design and validation of integrated urban models and for longitudinal studies concerned with evolution of urban form. Introduction of POLARIS electronic land registration system by the Province of Ontario in 1985 lead to the creation of an extensive dataset of real estate transactions by Teranet Enterprises Inc. However, despite having very high spatial and temporal resolution, available version of Teranet's dataset suffered from severe lack of features describing each individual transaction.

One of the major attributes missing from Teranet data is the type of property being transacted, or land use information for the parcel where a transaction is recorded. Along with selected Census and TTS variables, detailed parcel-level land use from the Department of Geography and DMTI land use data have been spatially joined to each Teranet record. However, since both of these data sources have their limitations, detailed land use data from Department of Geography has been used to train an algorithm capable of classifying land use based on the housing market dynamics; this way, land use information can be made available for each Teranet record for the full timespan covered by the Longitudinal Housing Market Research conducted by UTTRI .

To augment Teranet's dataset, new variables were engineered from its native attributes to capture the housing market dynamics at the parcel level: for example, 'xy_total_sales' was computed as the total count of Teranet records coming from a particular coordinate pair; 'med_price_xy' represents the median price of all records coming from a coordinate pair, etc. To augment Teranet data with demographic and transport information, the new Teranet features were spatially and temporally joined with Census and TTS variables recorded at the level of a Dissemination Area and TAZ zone, respectively. Finally, the augmented Teranet dataset has been tested with machine learning algorithms, attempting to classify land use for each Teranet record within the span of Census / TTS variables, thus recognizing land use changes with time.

A range of preprocessing techniques has been tested with several linear, tree-based and nearest neighbors classification algorithms; tree-based models and k-Nearest Neighbors significantly outperformed linear models. The new features engineered from native Teranet attributes have shown to have strong predictive power when classifying land use. When joined with Census variables at the level of Dissemination Areas, new features engineered from Teranet's dataset allowed the classification of land use with a high level of accuracy. A random forest model was trained using a random 70% sample of all Teranet records with new features from 2011 to 2014 stratified by target classes ("condo", "house", or "other"); the model achieved 97% of accuracy on the test subset composed of the remaining 30% of records from 2011 and 2014. Tree-based models did show some degree of overfitting and could benefit from further increase in the size of training data, as indicated by their learning and validation curves.

Features engineered from native Teranet attributes that capture price ratios to median and frequency of transactions from a coordinate pair have strong predictive power for land use classes, as indicated by feature selection techniques and model coefficients. This workflow could be further improved by joining more Census / TTS variables to engineer new features; target classes also could be redefined to allow more meaningful classification. In addition, results of the classification preformed by this workflow need to be investigated. A map produced with counts of misclassified Teranet records per DA shows that errors seem to be highly concentrated and correspond to high-frequency transactions, such as condos and mixed use properties. To facilitate further investigation of classification results, augmented Teranet dataset with new feature 'lucr_predict' along with related Census and TTS tables has been transformed into PostgreSQL relational database to facilitate ease of access by a broader group of specialists. An Entity Relationship (ER) diagram for the database created as a part of this master's thesis can be found in Appendix A; its referential integrity constraints were implemented based on the spatial and temporal relationships between data sources that were introduced in chapter 3.

Appendices

Appendix A

Entity Relationship (ER) diagram for RDBMS

The Entity Relationship (ER) diagram for the PostgreSQL database that was implemented as a part of this thesis is presented on the following page. In this database, Teranet dataset augmented with new features and land use produced by the best-performing classification algorithm is combined with related Census and TTS tables. Referential integrity constraints of this database were set up to reflect the nature of the spatial and temporal relationships introduced in chapter 3.

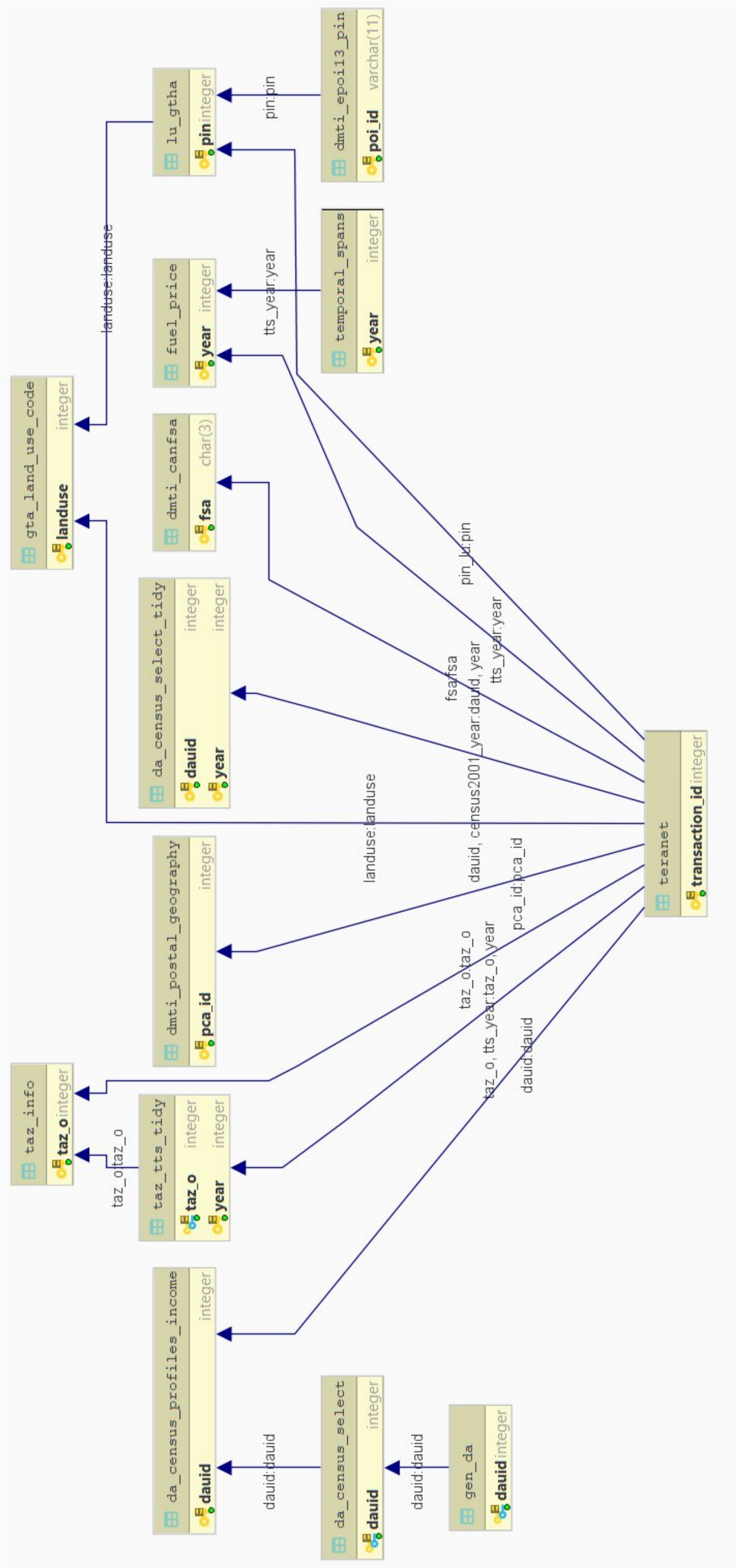


Figure A.1: Entity Relationship (ER) diagram for the PostgreSQL database that was implemented as a part of this thesis. In this database, Teranet dataset augmented with new features and land use produced by the best-performing classification algorithm is combined with related Census and TTS tables. Referential integrity constraints were set up to reflect the nature of the spatial and temporal relationships introduced in chapter 3

.

Bibliography

[1] Daniel Arribas-Bel. Accidental, open and everywhere: Emerging data sources for the understanding of cities. *Applied Geography*, 49:45–53, 2014.

[2] Bess Ashby. TTS 2016 City of Toronto Summary by Ward. Technical report, malatest, Toronto, 2018.

[3] Bank of Canada. Inflation Calculator, 2019.

[4] Michael Batty. Cities as Complex Systems: Scaling, Interactions, Networks, Dynamics and Urban Morphologies. 2008.

[5] Richard Bellman. The Theory of Dynamic Programming. Technical report, The RAND Corporation, 1954.

[6] Irad Ben-Gal. Outlier Detection. In *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, chapter Chapter 1, pages 1–16. Kluwer Academic Publishers, Tel-Aviv, 2005.

[7] Luís M.A. Bettencourt. The origins of scaling in cities. *Science*, 340(6139):1438–1441, 2013.

[8] Richard J Bouchard and Clyde E Pyers. Use of gravity model for describing urban travel: An analysis and critique. *Highway Research Record*, (88):1–43, 1965.

[9] Jason Brownlee. How to Prepare Data For Machine Learning, 2013.

[10] Cynthia Chen, Jingtao Ma, Yusak Susilo, Yu Liu, and Menglin Wang. The promises of big data and small data for travel behavior (aka human mobility) analysis. *Transportation Research Part C: Emerging Technologies*, 68:285–299, 2016.

[11] Edgar F Codd. *The relational model for database management : version 2*. Addison-Wesley Longman Publishing Co., Boston, MA, 1990.

[12] Kate Crowley and Brian W. Head. The enduring challenge of 'wicked problems': revisiting Rittel and Webber. *Policy Sciences*, 50(4):539–547, 2017.

[13] Data Management Group. Data Management Group at the University of Toronto Transportation Research Institute, 2014.

[14] Data Management Group. Survey Boundary Files, 2019.

[15] Thomas G Dietterich and Eun Bae Kong. Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms. pages 0–13, 1995.

[16] DMTI Spatial Inc. CanMap ® RouteLogistics User Manual v2014.2, 2014.

[17] Douglas M Hawkins. *Identification of outliers*. Chapman and Hall, London, UK, 1980.

[18] Michael Iacono, David Levinson, and Ahmed El-Geneidy. Models of transportation and land use change: A guide to the territory. *Journal of Planning Literature*, 2008.

[19] Jane Jacobs. The Death and Life of Great American Cities. In *New York*, volume 71, pages Alexander, C., Ishikawa, S., & Silverstein, M. (19. 1961.

[20] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer Texts in Statistics. Springer New York, New York, NY, 2013.

[21] Eric Damian Kelly. The Transportation Land Use Link. *Journal of Planning Literature*, 9(2):p.128–145, 1994.

[22] Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[23] Scikit learn developers. Compare the effect of different scalers on data with outliers, 2019.

[24] Scikit learn developers. L1 Penalty and Sparsity in Logistic Regression, 2019.

[25] Ira S Lowry. A Model of Metropolis. Technical report, Rand Corporation, Santa Monica CA, 1964.

[26] Map and Data Library. Canadian census geography (unit) definitions, 2019.

[27] F J Martinez. The bid-choice land-use model: an integrated economic framework. *Environment and Planning*, 24(January 1991):871–885, 1992.

[28] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity (reprinted from bulletin of mathematical biophysics, vol 5, pg 115-133, 1943). *Bulletin of Mathematical Biology*, 52(1–2):99–115, 1990.

[29] Heather McKean and Stella Di Cresce. The International Comparative Legal Guide to: Real Estate 2015, Chapter 6: Canada. Technical report, Global Legal Group, London, UK, 2015.

[30] Eric J Miller. Integrated urban modeling: Past, present, and future. *Journal of Transport and Land Use*, 11(1):387–399, 2018.

[31] Eric J Miller. The case for microsimulation frameworks for integrated urban models. *Journal of Transport and Land Use*, 11(1):1025–1037, 2018.

[32] Eric J Miller. Travel Demand Models, The Next Generation: Boldly Going Where No-One Has Gone Before. In K.G. Goulias and A.W. Davis, editors, *Mapping the Travel Behavior Genome, The Role of Disruptive Technologies, Automation and Experimentation*, chapter 2. 2019.

[33] Eric J Miller, Bilal Farooq, Franco Chingcuanco, and David Wang. Historical validation of integrated transport-land use model system. *Transportation Research Record*, (2255):91–99, 2011.

[34] Eric J Miller, David S Kriger, and John Douglas Hunt. *Integrated Urban Models for Simulation of Transit and Land Use Policies Guidelines for Implementation and Use.* 1998.

[35] Rolf Moeckel. Constraints in household relocation: Modeling land-use/transport interactions that respect time and monetary budgets. *Journal of Transport and Land Use*, 10(1):211–228, 2017.

[36] Robert Nisbet, Gary Miner, and Ken Yale. A Data Preparation Cookbook. In *Handbook of Statistical Analysis and Data Mining Applications*, chapter Chapter 18, pages 727–740. Elsevier, 2018.

[37] Guido Noto, Federico Cosenz, and Carmine Bianchi. *Urban Transportation Governance And Wicked Problems: A Systemic And Performance Oriented Approach.* Phd thesis, University of Palermo, 2015.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12:2825—-2830, 2011.

[39] Sebastian Raschka. Model Evaluation , Model Selection , and Algorithm Selection in Machine Learning. *arXiv e-prints*, (arXiv:1811.12808):arXiv:1811.12808, 2018.

[40] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning, 2nd Ed.* Packt Publishing, Birmingham, UK, 2 edition, 2017.

[41] Horst W.J. Rittel and Melvin M. Webber. Dilemmas in a General Theory of Planning. *Policy Sciences*, 4(2):155–169, 1973.

[42] Frank Rosenblatt. The Perceptron: a Percieving and Recognizing Automation. Technical report, Cornell Aeronautical Laboratory, Inc., Buffalo, NY, 1957.

[43] Adam Rosenfield, Franco Chingcuanco, and Eric J Miller. Agent-based housing market microsimulation for integrated land use, transportation, environment model system. *Procedia Computer Science*, 19:841–846, 2013.

[44] Colin Shearer. The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 5(4):13–22, 2000.

[45] Statistics Canada. Dissemination area (DA), 2015.

[46] Statistics Canada. Hierarchy of standard geographic units, 2018.

[47] Vergil G Stover and Frank J Koepke. *Transportation and Land Development.* Pearson College Div, 1988.

[48] Teranet Enterprises Inc. About POLARIS, Teranet, 2019.

[49] Teranet Enterprises Inc. https://www.teranet.ca, 2019.

[50] The Government of Ontario. Land Registration Reform Act, R.S.O. 1990, c. L.4, 1990.

[51] Hadley Wickham. Tidy Data. *Journal of Statistical Software*, 59(10), 2014.

[52] David H Wolpert. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1391–1420, 1996.

[53] David H Wolpert and William G Macready. No Free Lunch Theorems for Optimization. Technical Report 1, 1997.