

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни «Ймовірнісні моделі та статистичне оцінювання
в інформаційних системах»

Тема: «Прогнозування проблемних об'єктів будівництва»

Виконала:

студент групи ЗПІ-зп01

Кононов М. А.

Перевірив:

к.т.н., ст. доцент каф. ТК

Ліхоузова Т.А.

1. Загальна характеристика предметної області та постановка задачі

Будівництво – це галузь матеріального виробництва, в якій створюються основні фонди виробничого і невиробничого призначення: готові до експлуатації будівлі, споруди, їх комплекси. Цей термін охоплює:

- будівельні роботи, серед них земляні роботи і спорудження, конструктивні зміни, реставраційні роботи, капітальний і поточний ремонт (куди входять чистка й пофарбування) та знесення усіх видів будинків чи будівель;
- цивільне будівництво, куди входять земляні роботи й спорудження, конструктивні зміни, капітальний і поточний ремонт та знесення, наприклад, аеропортів, доків, гаваней, внутрішніх водних шляхів, гребель, захисних споруд на берегах річок і морів поблизу зон обвалів, автомобільних доріг і шосе, залізниць, мостів, тунелів, віадуків та об'єктів, пов'язаних з наданням послуг, а також комунікації, дренаж, каналізація, водопостачання й енергопостачання;
- монтаж та демонтаж будов і конструкцій з елементів заводського виробництва, а також виробництво збірних елементів на будівельному майданчику.

У реальному житті інколи трапляються ситуації, коли на певному етапі проектування, будівництва, здачі в експлуатацію, розподілення житлової площі об'єкт будівництва стає незавершеним або незаселеним. Наша робота присвячена дослідженню таких об'єктів усіх основних галузей будівництва: промислового, транспортного, житлово-цивільного. Для цього потрібна велика кількість даних як про успішні, так і про затримувані, недобудовані та незаселені будівлі. У всіх країнах, їх окремих регіонах, містах та селищах ситуація в житловій сфері зовсім різна, так само як і наявність або відсутність даних про нерухомість, зокрема історію будівництва. Тому збір і класифікація даних про житлові об'єкти часто є дуже великою проблемою. Офіційні джерела, авторами яких часто є урядові заклади, не завжди надають точну, достовірну та зручну для використання інформацію. Такі дані потребують рутинну обробку, та часто такий процес є важким для автоматизації.

Головна ціль нашої роботи – розробка інструментів для збору, класифікації й аналізу даних про успішні та проблемні об’єкти житлового будівництва, використання зібраної інформації для реалізації декількох моделей машинного навчання з ціллю прогнозування стану майбутніх будинків. Реалізоване нами дослідження може бути корисним для державних органів влади та місцевого самоврядування, а також інвесторів, робітників будівничих компаній та покупців житла. При належній реалізації такий проект здатен запобігти фінансуванню потенційно невдалого будівництва та втримати клієнтів від ризикового придбання житлової площі до здачі їх житла в експлуатацію. Вхідними даними для рішення названих задач має стати інформація про вдалі та невдалі об’єкти житлового будівництва у певному місті або регіоні однієї з країн у вигляді таблиці, наприклад, CSV-формату. Результати надаватимуться у вигляді нового стовпця до існуючої таблиці, який містить прогноз стану лише майбутніх житлових об’єктів.

1.1. Огляд предметної області

Проблемний об’єкт житлового будівництва – це нерухомість, фінансування якої здійснювалось із залученням коштів фізичних та / або юридичних осіб, яка не прийнята в експлуатацію в установленому законодавством порядку та яка включена до переліку проблемних об’єктів житлового будівництва, затвердженого державними органами влади. Слід зазначити, що до проблемних об’єктів нерухомості належать не лише об’єкти самочинного чи незавершеного будівництва. Проблемною є також нерухомість, щодо якої відсутні документи про право власності, нерухомість, що має незаконні перепланування або перебуває у заставі чи іпотеці, нерухомість, що арештована банком чи виконавчою службою. Інколи такий об’єкт може стати проблемним через помилки державного реєстратора, здійснені під час первинної реєстрації такої нерухомості в державному реєстрі речових прав на нерухоме майно або під час перереєстрації.

З метою врегулювання правовідносин, що виникли у зв’язку із залученням коштів на будівництво невдалих або збиткових споруд в деяких

країнах існують закони для визначення механізму зупинки або завершення їх будівництва, які передбачають:

- створення обласними міськими державними адміністраціями комісій для проведення технічного обстеження об'єктів житлового будівництва;
- проведення технічного обстеження проблемних об'єктів житлового будівництва з метою оцінки їх технічного стану та складення відповідного звіту, в якому обов'язково зазначається можливість або неможливість подальшого безпечного завершення будівництва або придатність до подальшої експлуатації;
- затвердження законодавчим органом влади вичерпного переліку проблемних об'єктів житлового будівництва;
- затвердження органами місцевого самоврядування цільових програм завершення будівництва, які включають сукупність взаємопов'язаних завдань і заходів, узгоджених за строками та ресурсним забезпеченням, спрямованих на організацію завершення будівництва проблемних об'єктів житлового будівництва;
- створення або визначення комунальних підприємств, які будуть виконувати функції замовників завершення будівництва проблемних об'єктів житлового будівництва.

У різних країнах світу ситуація з проблемними об'єктами будівництва має різний характер і ми спробуємо проаналізувати нашу державу. Вирішення питань добудови об'єктів незавершеного житлового будівництва є надзвичайно актуальним для України вже багато років, починаючи з однієї найгучніших та великомасштабних будівельних афер «Еліта-центр» на початку 2000-х років. Інвестори, потрапляючи на гачок реклами про європейський рівень комфорту, через брак повної інформації не можуть якісно та повною мірою оцінити ризики, пов'язані з інвестуванням у той чи інший об'єкт будівництва. У зв'язку з нечесною поведінкою деяких забудовників інвестори тривалий час не можуть отримати у власність об'єкти нерухомого майна, або повернути кошти, залучені у будівництво таких об'єктів.

Недобудовою може бути не лише котлован, а й готова будівля, іноді навіть заселена, але будинок не введено в експлуатацію та частково або повністю не підключено до мереж. Кількість ошуканих інвесторів поповнюється, що може перетворити проблему недобудов на соціальну.

За статистикою Міністерства розвитку громад та територій нині в Україні до категорії проблемних можна віднести понад 150 об'єктів житлового будівництва, третина з яких знаходиться у Києві. Для покращення їх статусу вищезазначений проект Закону працюватиме за наступним механізмом:

1. Обласні, Київська та Севастопольська міські держадміністрації утворюють комісії для проведення обстеження об'єктів будівництва;
2. Комісії проводять обстеження у два етапи: візуальний та інструментальний;
3. За результатами обстеження складається звіт із висновками щодо можливості/неможливості добудови;
4. Обласні, Київська та Севастопольська міські держадміністрації на підставі вказаних звітів формують пропозиції до переліку проблемних об'єктів житлового будівництва;
5. Міністерство розвитку громад та територій формує проект єдиного переліку проблемних об'єктів житлового будівництва і подає його на затвердження Кабміну;
6. Кабмін затверджує єдиний перелік проблемних об'єктів житлового будівництва і потім вже не може його коригувати;
7. Органи місцевого самоврядування публікують оголошення про початок процедури врегулювання зобов'язань відносно проблемного об'єкта;
8. Співвласники майнових прав (інвестори будівництва) подають заяви про включення до переліку співвласників;
9. Співвласники майнових прав або спеціально визначне комунальне підприємство подає до господарського суду заяву про затвердження плану врегулювання зобов'язань відносно проблемного об'єкта;
10. Господарський суд за розташуванням проблемного об'єкта розглядає справу впродовж 1-го місяця та виносить відповідну ухвалу;

11.Добудова проблемного об'єкта здійснюється за окремим кошторисом по кожному конкретному об'єкту коштами місцевого бюджету.

Проаналізувавши дані положення, ми дійшли висновку, що їх виконуваність залежить від добробуту держави: якщо в бюджеті або приватних осіб немає необхідних ресурсів, проблемний об'єкт не буде добудований. Такі пункти можуть функціонувати в будь-якій країні світу, де діє подібний закон про проблемні споруди.

Передбачення проблемності об'єктів, які знаходяться на певному етапі будівництва, здачі в експлуатацію чи розподілення й продажу житлової площі потребує поглибленого аналізу даних про існуючі успішні та неуспішні будівлі. Наявні дані здатні бути ефективними лише в певному регіоні, місті або селищі країни. Для отримання бажаних результатів ми маємо знайти набір певної інформації, завантажити його в спеціально розроблений програмний комплекс, виконати статистичний аналіз, побудувавши графіки та діаграми, після чого отримати прогноз стану майбутніх об'єктів. Отже, нами буде виконаний поглиблений аналіз даних про будинки одного міста.

1.2. Огляд доступних джерел даних

Для побудови системи машинного навчання, яка прогнозуватиме проблемність певного об'єкта будівництва за наявною про нього інформацією, ми маємо здобути історичні дані про вдалі та невдалі будинки. Чим більше існуючих об'єктів ми проаналізуємо, тим точніше працюватиме наша програма для певного регіону, селища або всієї країни. На основі зібраних даних ми маємо сформулювати та оцінити ознаки, які впливають на те, чи перетвориться проект на недобудову. Доступність інформації про проблемні об'єкти будівництва залежить від того, наскільки сильну увагу держава приділяє сфері інформаційних технологій, чи висвітлює вона дані про будівельні об'єкти в офіційних джерелах або чи проявляють таку ініціативу приватні компанії. Ми спробуємо надати та проаналізувати дані про житло у нашій країні, зокрема в місті Києві. Столиця – це одне з не багатьох міст, про об'єкти будівництва якої можна знайти багато інформації. Зібраних даних може бути достатньо для

функціонування певної моделі машинного навчання з метою передбачення стану майбутнього житла.

На офіційному сайті Київської міської ради kyivcity.gov.ua є актуальний перелік проблемних об'єктів будівництва (об'єктів самочинного будівництва та довгобудів). Він зроблений у вигляді таблиці, яка по кожному району надає дані про адресу будівлі, її замовника, відомість про наявність або відсутність вихідних даних на проектування та дозволу на виконання будівельних робіт, інформацію про підключення до інженерних мереж та про наявні причини зупинення будівельних процесів, у тому числі на підставі судових рішень або приписів ДАБІ, правовстановлюючі документи на земельну ділянку, поточний стан об'єктів тощо. Більш детальну інформацію про кожен будівлю, яка знаходиться у місті Києві та деяких прилеглих до нього селах, можна знайти на порталі Київської нерухомості my-realty.kiev.ua/doma. Даний ресурс надає детальні дані, які включають в себе номер проекту, рік завершення або початку будівництва, матеріали, кількість поверхів, висоту стелі, фото.

Потужним недоліком існуючих веб-сайтів з інформацією про об'єкти будівництва є відсутність можливості отримати дані в зручному для аналізу вигляді, тобто як файл таблиці або бази даних. Тому для побудови системи машинного навчання або просто збереження та аналізу існуючих даних потрібно виконувати ручний увід дуже великої кількості записів. У зв'язку з цим нами було прийнято рішення ввести вручну лише інформацію про майже всі проблемні об'єкти житлового будівництва, зазначені на вищевказаному офіційному порталі Київської міської ради, а дані про завершені будівлі згенерувати випадковим чином. Введені дані «накладаються» на реальні адреси, перелік яких можна завантажити у зручному табличному вигляді з офіційних сайтів деяких поштових сервісів. Для автоматизації цього процесу нами був розроблений невеликий скрипт на мові Python, який завантажує всю таблицю адрес із CSV-файлу та відомості про невдалі об'єкти в базу даних PostgreSQL. Для роботи з цією базою даних нами була зроблена веб-сторінка, яка має наступний вигляд:

Django web page

127.0.0.1:8000 95%

Область: Київ Населений пункт: м. Київ Вулиця: алея Вернадського Академіка

Додати будинок Зберегти зміни в БД Видалити з таблиці Видалити з бази даних Зберегти таблицю у CSV-файл Генерувати випадкові дані Спрогнозувати стан

<input type="checkbox"/>	№	Вулиця	№ буд.	Рік	Матеріали	Кількість поверхів	Висота стелі	Стан	Прогноз
<input type="checkbox"/>	1	вул. Ракетна	10	1999	цегла	24	2.62	Зданий	
<input type="checkbox"/>	2	вул. Ракетна	11	1965	монолітн	17	2.42	Зданий	
<input type="checkbox"/>	3	вул. Ракетна	12	1984	з/б пане	2	3.19	Зданий	
<input type="checkbox"/>	4	вул. Ракетна	13	1970	вентильс	22	2.75	Зданий	
<input type="checkbox"/>	5	вул. Ракетна	14	2023	керамбл	22	3.1	Будується	Успіх
<input type="checkbox"/>	6	вул. Ракетна	15	1978	газоблн	12	3.09	Зданий	
<input type="checkbox"/>	7	вул. Ракетна	16	1970	цегла	9	2.77	Зданий	
<input type="checkbox"/>	8	вул. Ракетна	17	1971	монолітн	12	3.15	Зданий	
<input type="checkbox"/>	9	вул. Ракетна	18	1975	утеплюв	12	3.12	Зданий	
<input type="checkbox"/>	10	вул. Ракетна	19	1974	утеплюв	12	2.72	Зданий	
<input type="checkbox"/>	11	вул. Ракетна	2	2009	з/б пане	13	2.92	Зданий	
<input type="checkbox"/>	12	вул. Ракетна	20	1984	цегла	4	2.83	Зданий	
<input type="checkbox"/>	13	вул. Ракетна	21	1969	цегла сіл	1	2.99	Зданий	
<input type="checkbox"/>	14	вул. Ракетна	22	1976	вентильс	17	2.83	Зданий	
<input type="checkbox"/>	15	вул. Ракетна	23	2011	монолітн	17	2.61	Зданий	
<input type="checkbox"/>	16	вул. Ракетна	24	2014	монолітн	25	2.7	Невдала будівля	
<input type="checkbox"/>	17	вул. Ракетна	24А	1992	к/б пане	4	2.77	Зданий	
<input type="checkbox"/>	18	вул. Ракетна	24Б	1976	цегла сіл	17	3.03	Зданий	
<input type="checkbox"/>	19	вул. Ракетна	25	1966	газоблн	17	2.63	Зданий	
<input type="checkbox"/>	20	вул. Ракетна	26	1993	керамбл	16	3.08	Зданий	
<input type="checkbox"/>	21	вул. Ракетна	27	1989	вентильс	10	2.63	Зданий	
<input type="checkbox"/>	22	вул. Ракетна	3	2003	вентильс	17	2.69	Зданий	
<input type="checkbox"/>	23	вул. Ракетна	4	1990	вентильс	19	2.76	Зданий	
<input type="checkbox"/>	24	вул. Ракетна	5	2001	шлакобл	13	2.74	Зданий	
<input type="checkbox"/>	25	вул. Ракетна	6	1996	к/б пане	24	2.92	Зданий	
<input type="checkbox"/>	26	вул. Ракетна	7	2018	з/б пане	8	2.8	Зданий	

Додаток дозволяє відображати, додавати, коригувати, доповнювати, видаляти, генерувати випадкові дані, а також прогнозувати стан майбутніх об'єктів. Ми зберігаємо наступні дані: назва вулиці, номер будинку, рік будівництва (зазвичай вказуємо рік здачі в експлуатацію, але у випадку невдалого об'єкту це поле містити рік початку роботи чи останнього оновлення інформації про стан), головні будівельні матеріали, кількість поверхів, висота стелі, один з п'яти станів («проектується», «будується», «призупинений», «невдала будівля», «зданий»), прогноз (це поле отримує значення «успіх» або «недобудова» лише у рядках, в яких рік будівництва більше або дорівнює 2022. Уведення назви міста або вулиці для прискорення введення даних супроводжується відображенням випадного списку, який виглядає так:

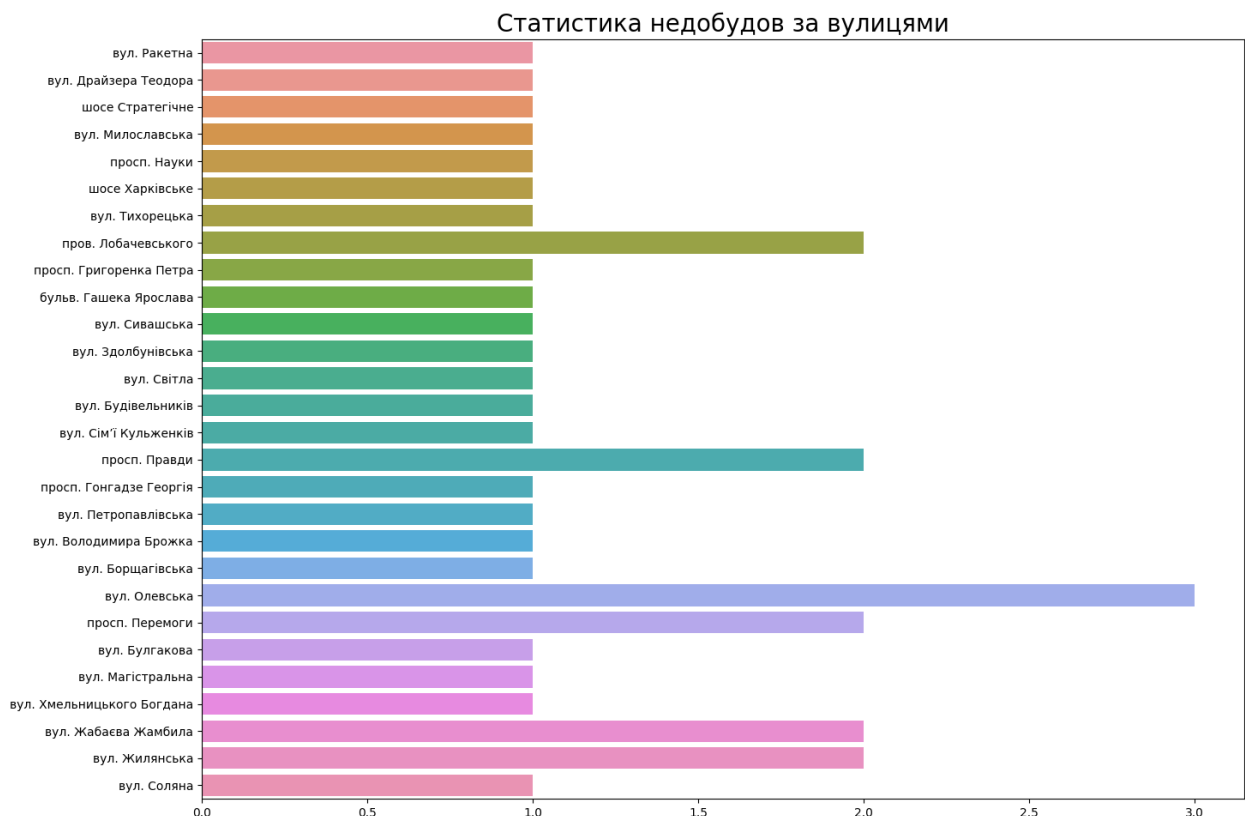
Вулиця

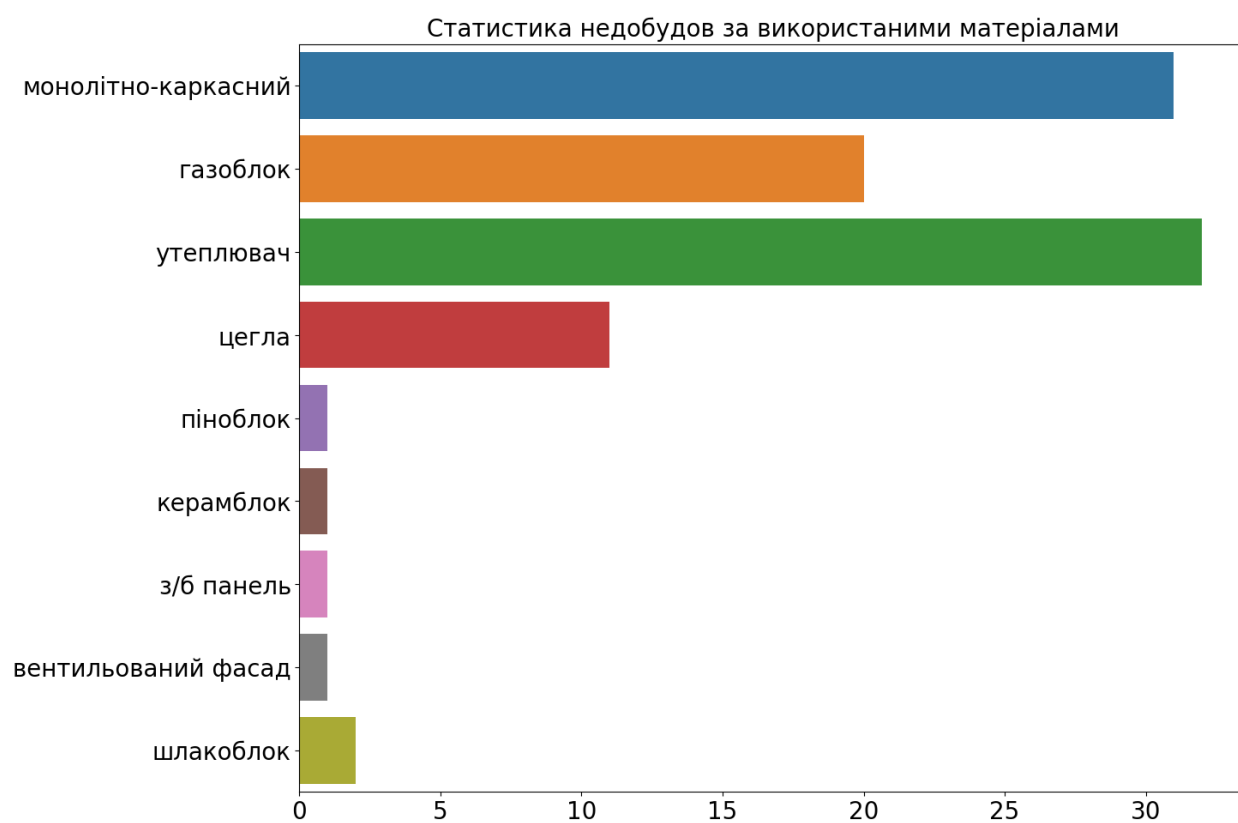
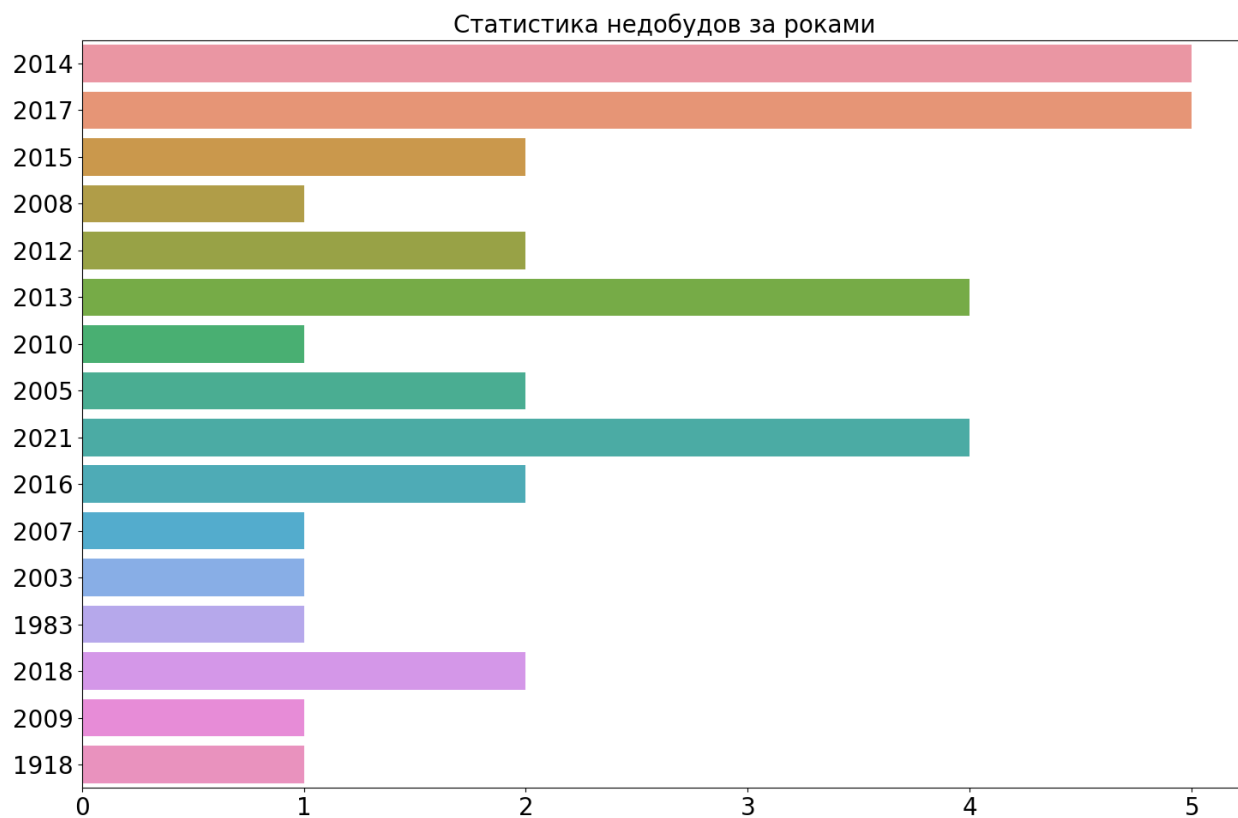
шевч

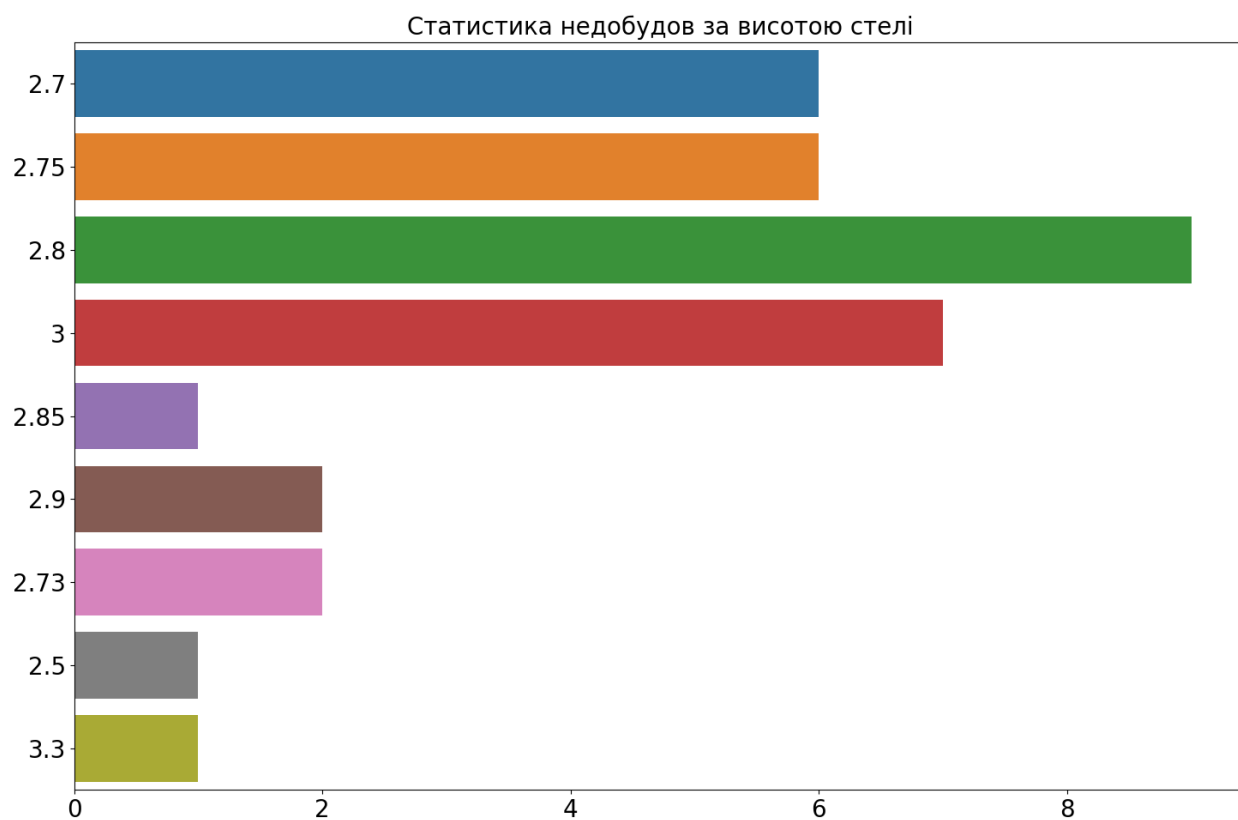
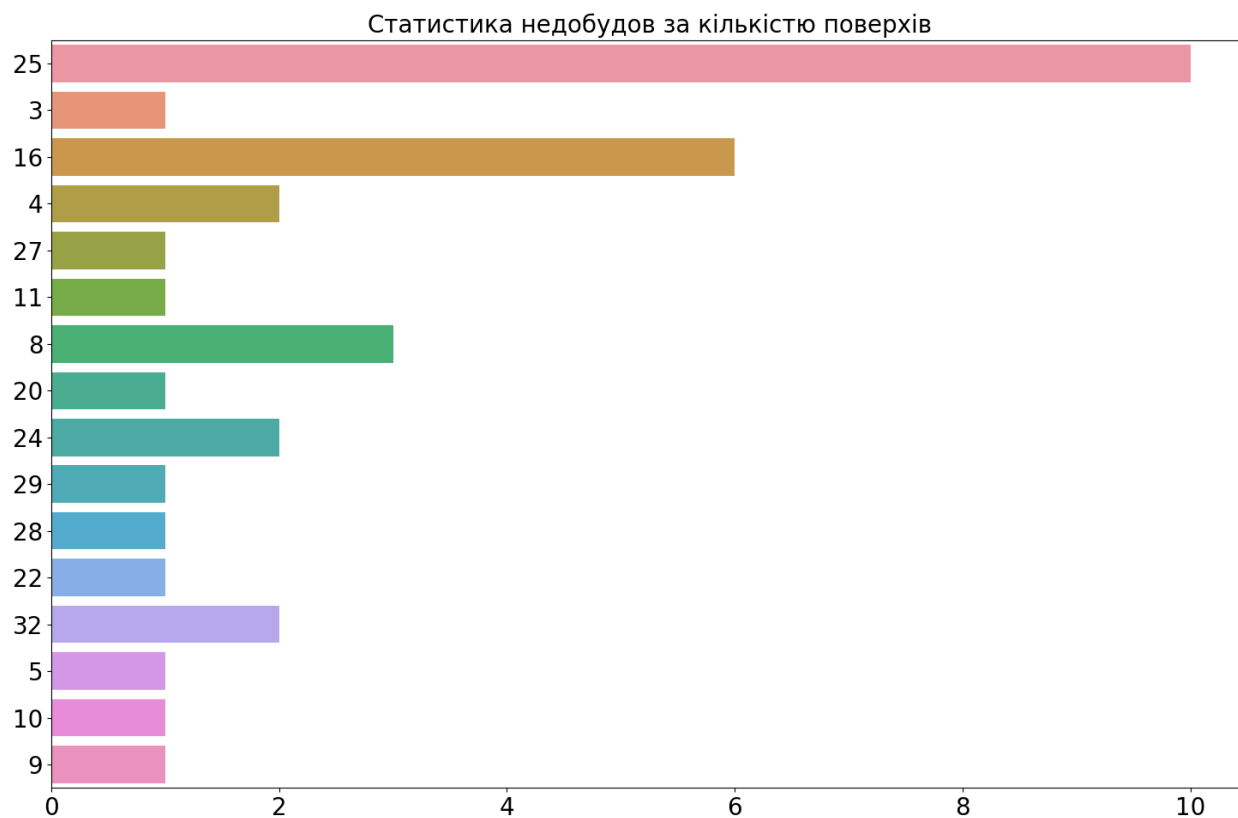
- бульв. Шевченка Тараса
- вул. Корсунь-Шевченківська
- вул. Шевченка (Бортничі)
- вул. Шевченка (Жуляни)
- вул. Шевченка (Троєщина)
- пров. Шевченка (Жуляни)

Кнопки «+» та «-» додають або видаляють із таблиці записи, назва вулиці яких співпадає із вказаною у відповідному полі. Для реалізації back-end частини проекту ми використовуємо фреймворк Django, який дозволяє виконувати sql-запити, а також містить можливості додавання бібліотек Python для зручного використання інструментів машинного навчання.

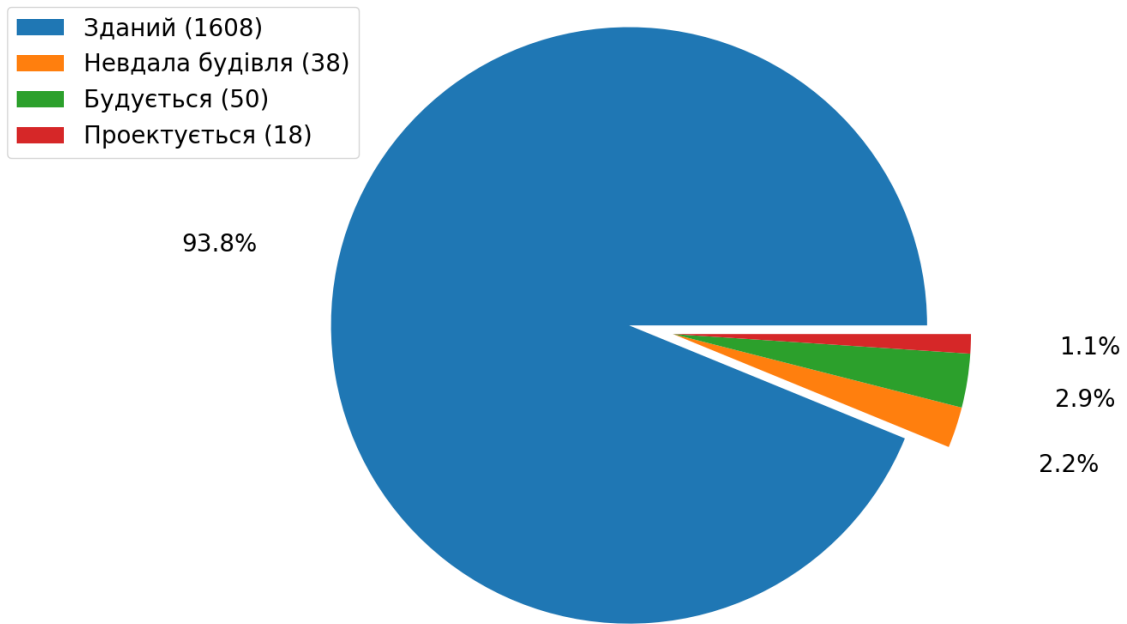
Отже, розроблений нами програмний продукт дозволяє працювати як з існуючими даними про об'єкти житлового будівництва, так і генерувати випадкові, які можуть бути дуже наближеними до реальних. При отриманні інформації у зручному для обробки вигляді вона може бути легко вбудованою в створену нами систему. Окрім вищезазначених функціональних можливостей при натисканні на кнопку «Спрогнозувати стан» наш веб-додаток створює 5 графіків та 2 діаграми у папці проекту, які будуються засобами бібліотеки matplotlib для мови Python. У більшості випадків вони ілюструють ситуацію лише з даними про невдалі об'єкти, оскільки вони є достовірними та, на нашу думку, немає сенсу детально аналізувати випадкові дані. Така інформація отримується шляхом підрахунку кількості входжень певного параметра і має наступний вигляд:



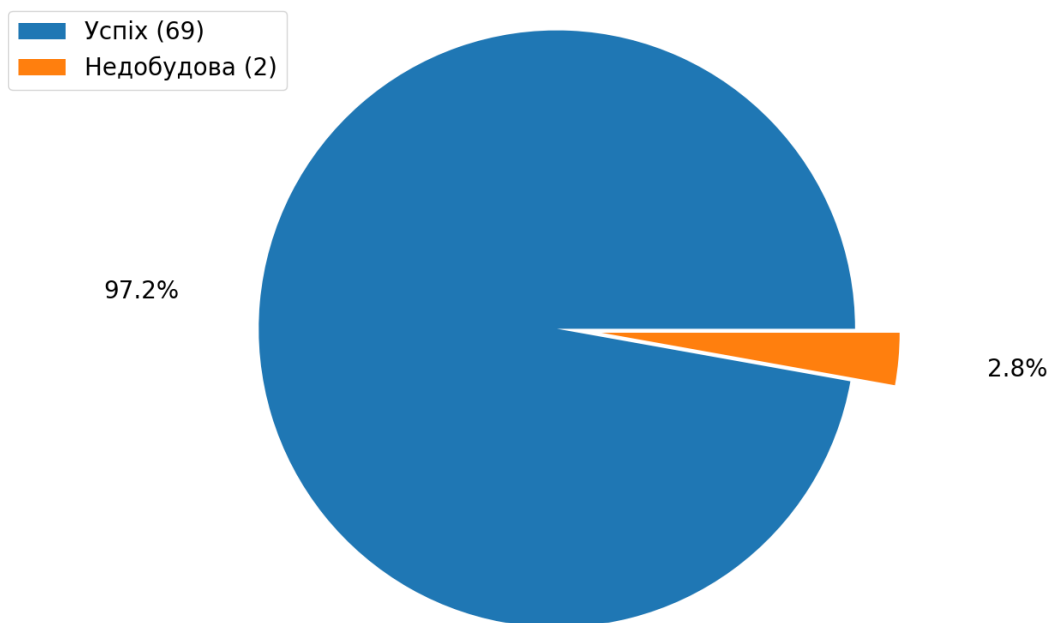




Співвідношення зданих в експлуатацію будівель та недобудов



Спрогнозоване співвідношення зданих в експлуатацію будівель та недобудов



Перед прогнозуванням стану розроблена програма здійснює попередню підготовку даних, яка включає в себе завантаження значень існуючих полів та додавання нових стовбців, які містять кількість входжень того чи іншого параметра. Здійснюється підрахунок недобудов на тій чи іншій вулиці,

формується статистика за роками, рейтингом головних будівельних матеріалів та станом. Отримані прогнози ми також зображуємо на круговій діаграмі.

1.3. Постановка задачі

Головна мета нашої роботи – отримання програми для ручного та автоматичного уведення даних про існуючі та майбутні об'єкти будівництва, збереження відомостей про них у базі даних, аналіз, редагування, доповнення, візуалізація та прогнозування стану майбутніх будинків шляхом використання декількох моделей. Функція прогнозування стану має викликатись як через інтерфейс веб-додатку (у цьому разі результати потрапляють у відповідні комірки таблиці), так і за допомогою консольної програми, для якої вхідними та вихідними даними є CSV-файли строго заданої структури. Також консольний додаток має виводити точність роботи реалізованих моделей. Проведене дослідження повинне дати відповідь лише на одне запитання: чи буде майбутній об'єкт будівництва успішним, тобто чи введуть його в експлуатацію та чи не виявиться він збитковим для інвесторів.

2.1. Визначення моделей, що можуть бути використані

У процесі створення продукту, який має виконувати вищезазначені задачі, ми маємо вирішувати задачі регресії та класифікації. Регресія – це форма зв'язку між випадковими величинами, закон зміни математичного сподівання однієї випадкової величини залежно від значень іншої. Розрізняють прямолінійну, криволінійну, ортогональну, параболічну та інші регресії, а також лінію і площину регресії. Класифікація об'єкта – це номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування. У машинному навчанні завдання класифікації вирішується, як правило, за допомогою методів штучної нейронної мережі при постановці експерименту в вигляді навчання з учителем.

Наша задача полягає в розробці засобів для отримання відповіді на чітке запитання: чи стане майбутній об'єкт будівництва успішним. Отже, в якості результату розроблений код має повернути нуль або одиницю.

Проаналізувавши існуючі моделі машинного навчання, ми виділили три з них, які в найбільшій мірі підходять для вирішення нашої проблеми. Це логістична регресія, випадковий ліс та метод опорних векторів.

Логістична регресія – це статистичний регресійний метод, що застосовують у випадку, коли залежна змінна є бінарною, тобто може набувати тільки двох значень (0 або 1). Прикладом може слугувати класифікація електронних листів на «спам» або «не спам». Метод також використовується у медицині, наприклад, для визначення чи є пухлина злоякісною, чи доброякісною. Логістична регресія є розширенням лінійної регресії для вирішення завдань класифікації. Її основні припущення:

- у бінарній логістичній регресії потрібна залежна змінна, а порядкова логістична регресія вимагає, щоб залежна змінна була порядковою;
- спостереження не повинно виходити з повторюваних вимірювань або збіганих даних;
- логістична регресія відкидає мультиколінеарність між незалежними змінними;
- ця модель працює на основі припущення незалежної лінійності. Він вимагає, щоб незалежні змінні були лінійно пов'язані з переліком шансів.

Логістичну регресію можна використовувати для будь-якої кількості незалежних змінних, але неможливо представити результат у більше ніж у трьох вимірах.

Випадковий ліс – це ансамблевий метод машинного навчання для класифікації, регресії та інших завдань, який працює за допомогою побудови численних дерев прийняття рішень під час тренування моделі й продукує моду для класів (класифікацій) або усереднений прогноз (регресія) побудованих дерев. Недоліком є схильність до перенавчання. Усі дерева комітету будуються незалежно один від одного за такою процедурою:

1. Згенеруємо випадкову підвибірку з **повторенням** розміром n з навчальної вибірки. (Таким чином, деякі приклади потраплять в неї кілька разів, а приблизно $N/3$ прикладів не ввійдуть у неї взагалі).
2. Далі випадково обираємо m предикторів (ознак) із M .

3. Побудуємо дерево рішень, яке класифікує приклади даної підвибірки, причому в ході створення чергового вузла дерева будемо вибирати ознаку, на основі якої проводиться розбиття, не з усіх M ознак, а лише з m випадково вибраних. Вибір найкращого з цих m ознак може здійснюватися різними способами. В оригінальному коді Брейман використовується критерій Джині, що застосовується також в алгоритмі побудови вирішальних дерев CART. У деяких реалізаціях алгоритму замість нього використовується критерій приросту інформації.
4. Розділимо ознаку X на два класи, $X_i \geq S_i$ та $X_i < S_i$.
5. Виміряємо гомогенність у двох нових класах за допомогою критерію Джині.
6. Оберемо таке значення «спліт-поінту» S_i ознаки X , для якого досягнуто максимальної гомогенності класу.
7. Дерево будується до повного вичерпання підвибірки і не піддається процедурі відсікання (на відміну від дерев рішень, побудованих за таким алгоритмом, як CART або C4.5).
8. Повертаємося до пункту 1. генеруємо нову вибірку і повторюємо пункти 2. — 4. будуючи наступне дерево. Чим більше дерев побудовано, тим меншою буде помилка класифікатора на тестовій вибірці.

Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету відносить об'єкт, який класифікується до одного з класів, і перемагає клас, за який проголосувало найбільше число дерев.

Оптимальне число дерев підбирається таким чином, щоб мінімізувати помилку класифікатора на тестовій вибірці. У разі її відсутності, мінімізується оцінка помилки out-of-bag: частка прикладів навчальної вибірки, неправильно класифікованих комітетом, якщо не враховувати голоси дерев на прикладах, що входять в їх власну навчальну підвибірку.

Метод опорних векторів — це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням з пов'язаними алгоритмами навчання, які називаються опорно-векторними машинами (ОВМ). Для заданого набору тренувальних зразків, кожен із яких відмічено як належний до однієї чи іншої з двох категорій, алгоритм

тренування ОВМ будує модель, яка відносить нові зразки до однієї чи іншої категорії, роблячи це неймовірно бінарним лінійним класифікатором. Модель ОВМ є представленням зразків як точок у просторі, відображених таким чином, що зразки з окремих категорій розділено чистою прогалиною, яка є щонайширшою. Нові зразки тоді відображаються до цього ж простору, й робиться передбачення про їхню належність до категорії на основі того, на який бік прогалини вони потрапляють.

ОВМ можуть застосовуватися для розв'язання різноманітних практичних задач:

- ОВМ є корисними для категоризації текстів та гіпертекстів, оскільки їхнє застосування може значно знижувати потребу в мічених тренувальних зразках як у стандартній індуктивній, так і в трансдуктивній[en] постановках.
- Із застосуванням ОВМ може виконуватися й класифікація зображень. Експериментальні результати показують, що ОВМ можуть досягати значно вищої точності пошуку, ніж традиційні схеми уточнення запиту, всього лише після трьох-чотирьох раундів зворотного зв'язку про відповідність. Це є вірним і для систем сегментування зображень, включно з тими, які використовують видозмінену версію ОВМ, яка застосовує привілейований підхід, запропонований Вапником.[4][5]
- За допомогою ОВМ може здійснюватися розпізнавання рукописних символів.
- Алгоритм ОВМ широко застосовується в біологічних та інших науках. Їх було використано для класифікації білків з правильною класифікацією до 90 % складу. Як механізм інтерпретування моделей ОВМ було запропоновано пермутаційний тест на основі вагових коефіцієнтів ОВМ.[6][7] Вагові коефіцієнти опорно-векторних машин використовувалися для інтерпретування моделей ОВМ і в минулому. Ретроспективне інтерпретування моделей опорно-векторних машин з метою ідентифікації ознак, які використовує модель для здійснення передбачень, є відносно новою областю досліджень з особливим значенням у біологічних науках.

Бібліотека sklearn для мови програмування Python має дуже зручні засоби для реалізації описаних моделей. Наведемо код отримання даних з двовимірному масиву та використання цих трьох методів:

```
120 df=pd.DataFrame(dfl,columns=['Вулиця','Кількість недобудов на
вулиці','№ буд.','Рік','Кількість недобудов за роком','Матеріали',
'Рейтинг матеріалів','Кількість поверхів','Висота стелі','Стан',
'Стан (bool)'])
121 y = df.iloc[:,10].values
122 df.drop(df.columns[[0,2,3,5,9,10]],axis=1,inplace=True)
123 x = df.iloc[:,0:]
124 x_train, x_test, y_train, y_test = train_test_split(x,y,random_state
=0)
125 lgc = LogisticRegression(solver='lbfgs',random_state=0)
126 lgc.fit(x_train, y_train)
127 dfp=pd.DataFrame(dfp,columns=['Вулиця','Кількість недобудов на
вулиці','№ буд.','Рік','Кількість недобудов за роком','Матеріали',
'Рейтинг матеріалів','Кількість поверхів','Висота стелі','Стан',
'Стан (bool)'])
128 dfp.drop(dfp.columns[[0,2,3,5,9,10]],axis=1,inplace=True)
129 LGPredictions = lgc.predict(dfp)
130 LGStatesDict={}
131 RFPredictions = rfc.predict(dfp)
132 SVMStatesDict={}
133 rfc = RandomForestClassifier()
134 rfc.fit(x_train,y_train)
135 RFPredictions = rfc.predict(dfp)
136 SVM = svm.SVC()
137 SVM.fit(x_train, y_train)
138 SVMPredictions = SVM.predict(dfp)
139
140
```

python file length: 8496 lines:183 Ln:143 Col:1 Pos:5203 Windows (CR LF) UTF-8 INS

2.2. Вибір ознак, що будуть використані для аналізу

Набір ознак, на основі яких буде зроблений висновок про успішність або неуспішність певної будівлі повністю залежить від змісту доступних таблиць. Проаналізувавши вищезазначені джерела та систематизувавши дані з них, ми сформувавши такий перелік вхідних параметрів: назва вулиці, кількість недобудов на ній, № будинку, рік та кількість недобудов, які виникли в зазначений період, основні будівельні матеріали (цегла, утеплювач, керамічний блок, газоблок тощо) та сума їх рейтингів, кількість поверхів, висота стелі, поточний стан. Звичайно, не всі, особливо не числові, ознаки приймають участь в аналізі та прийнятті рішень. Такі параметри як назва вулиці, № будинку, рік, перелік матеріалів немає сенсу та часто неможливо прямо зв'язувати з результатами прогнозування. Замість них ми використовуємо кількість проблемних об'єктів з наведеними ознаками. Всі вони відображаються у першому рядку відповідних CSV-файлів як назви стовпців.

2.3. Підготовка даних для навчання та верифікації моделей

Під час пошуку інформації про об'єкти нерухомості нами було встановлено, що перелік всіх існуючих адрес може бути знайдений на офіційних сторінках поштових служб, наприклад, Нової пошти, у вигляді CSV-файлу. Він містить такі стовпці: область, населений пункт, вулиця, № будинку. Для їх додавання в базу даних, яка реалізована засобами ПЗ PostgreSQL 14, нами був розроблений спеціальний скрипт на мові Python. Він також включає в себе ініціалізуючі запити, які формують загальну структуру БД, створюючи пусті таблиці з необхідними полями та ключами, організує їх зв'язок. Крім того, ми включили в цей скрипт додавання достовірної інформації про недобудови, яка була зібрана із офіційних джерел (на порталах Київської міської ради та нерухомості). Веб-сайт <https://my-realty.kiev.ua/houses/kyev-1/> містить дані про дуже велику кількість будівельних об'єктів, але, на жаль, нам не вдалося отримати їх у зручному для автоматизованого використання вигляді. Тому інформацію про інші успішні будинки ми були вимушені згенерувати випадковим чином, максимально наблизивши параметри до реальних. У результаті впровадження такого підходу ми отримуємо повністю заповнену таблицю, яка не містить пустих полів та не потребує додаткових операцій з даними, таких як нормалізація значень. Це спрощує роботу алгоритмів збору, систематизації та збереження даних, а, отже, не потребує включати код для перевірки коректності значень деяких полів. Ми розуміємо, що випадкові дані в такому дослідженні майже повністю унеможливають отримання достовірних результатів аналізу. Тому вони можуть бути легко замінені за допомогою вищезазначеного скрипту та при наявності необхідних текстових файлів.

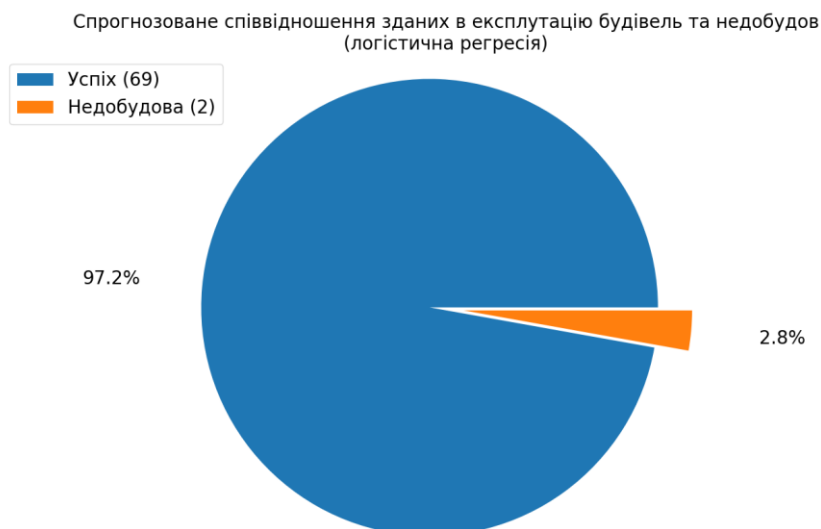
У процесі збору даних розроблений нами програмний продукт розділяє їх на навчальні та ті, для яких потрібно спрогнозувати стан. Таке розподілення виконується за однією ознакою: роком будівництва. Об'єкти, які мають бути здані в експлуатацію в 2022 році або пізніше, потрапляють до окремого масиву прогнозованих, а всі інші використовуються для уможливлення цього прогнозу й покращення результатів.

2.4. Формування моделей. Вибір оптимального класу складності

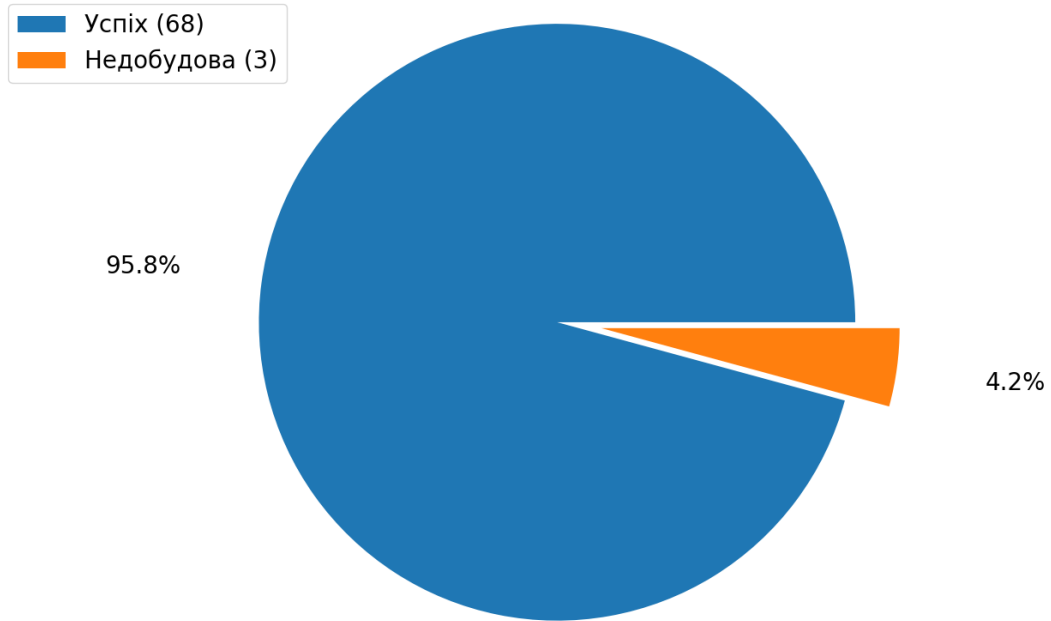
Деякі моделі машинного навчання, реалізовані у вигляді готових бібліотек для мови Python, мають параметр *cp*, який відповідає за складність. Основна його роль полягає в збереженні обчислювального часу шляхом відсічення розщеплень, які марно використовують апаратні ресурси. Вказуючи даний параметр, користувач повідомляє програмі, що будь-яке розподілення, яке покращує адаптацію за складністю, скоріш за все буде вилучене перехресною перевіркою, і тому програмі не треба його виконувати. Моделі, які базуються на дереві рішень мають схожий параметр *maxdepth*, який регулює максимальну глибину будь-якого вузла кінцевого дерева, причому корінний вузол вважається глибиною 0. У процесі побудови проекту ми не побачили необхідності вказувати такі обмеження, оскільки боялись погіршити точність роботи моделі. Проблема вибору оптимального класу складності стає актуальною при проблемах з продуктивністю та при бажанні покращити її, не нехтуючи точністю.

2.5. Верифікація моделей

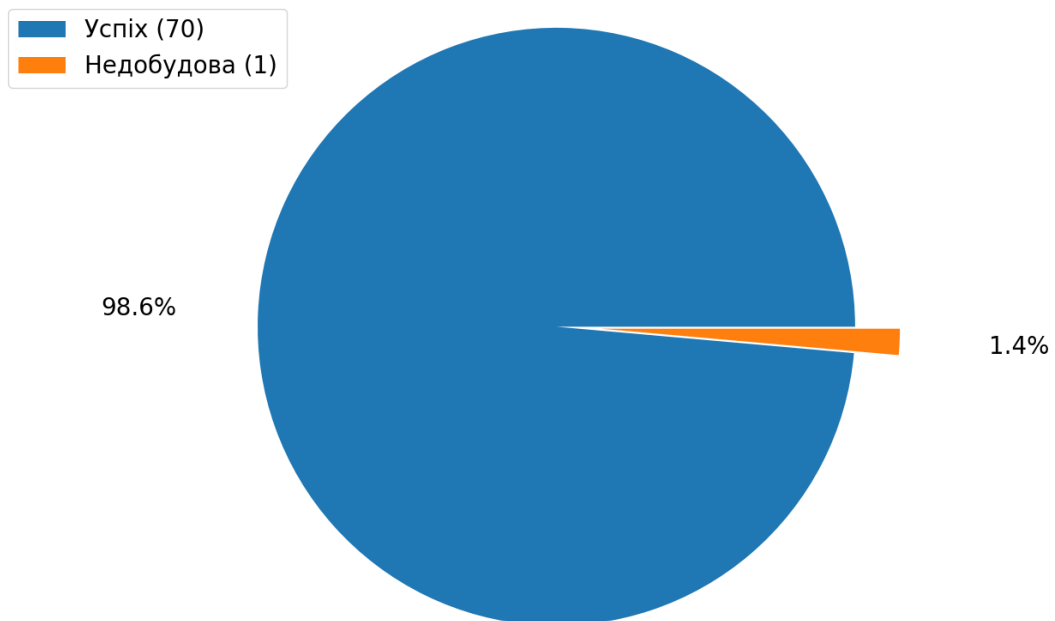
У ході проведення аналізу за допомогою трьох різних моделей ми помітили, що часто вони надають зовсім різні прогнози. Це пов'язано з принциповою відмінністю випробуваних нами алгоритмів. Консольний додаток ілюструє результати роботи моделей у вигляді трьох діаграм:



Спрогнозоване співвідношення зданих в експлуатацію будівель та недобудов
(випадковий ліс)



Спрогнозоване співвідношення зданих в експлуатацію будівель та недобудов
(метод опорних векторів)

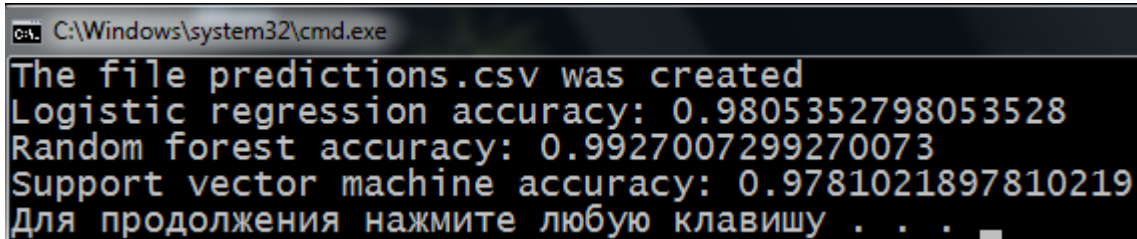


Також причинами таких розбіжностей може бути використання моделей із параметрами за замовчуванням: ми в жодному випадку не вказували додаткові уточнюючі та обмежувальні опції. Отримання повністю однакових результатів дослідження за допомогою різних моделей здається нам не

реальним. Було помічено, що в кількох випадках два алгоритми попарно прогнозують однаковий негативний результат, що може в деякій мірі свідчити про достовірність виконаного аналізу.

2.6. Висновки щодо якості побудованих моделей

Наш консольний додаток містить в собі можливість виведення оцінок точності реалізованих моделей:



```
C:\Windows\system32\cmd.exe
The file predictions.csv was created
Logistic regression accuracy: 0.9805352798053528
Random forest accuracy: 0.9927007299270073
Support vector machine accuracy: 0.9781021897810219
Для продовження натисніть будь-яку клавішу . . .
```

Найбільш точним виявився випадковий ліс, який вважається найбільш універсальним, здатним вирішувати задачі класифікації, регресії, кластеризації, пошуку аномалій, селекції ознак тощо. Він є прикладом ансамблевого алгоритму, що ґрунтується на передбаченнях вирішальних дерев. Ідея ансамблевих моделей полягає у побудові великої кількості простих моделей, результати яких накопичуються. Інші моделі виявились достатньо точними та, на нашу думку, не потребують заходів для їх оптимізації. Такі показники обумовлені також невеликою кількістю ознак для аналізу, відносно великими розмірами навчальної збірки. Було встановлено, що надмірна кількість вхідних параметрів для аналізу може негативно впливати на точність та швидкість реалізованих моделей. Це може стати причиною реструктуризації вибірок, а також встановлення інших параметрів роботи алгоритмів.

3. Результати аналізу

Ми можемо стверджувати, що повністю реалізували вищеописані завдання. Нами була вирішена складна проблема отримання даних для навчання, в свою чергу ми були вимушені подолати багато перешкод при розробці веб-додатку та його серверної частини. Було розроблено ефективне рішення для швидкого розгортання проекту з нуля на інших ПК, розроблені функції ефективно надають підсумкову інформацію у вигляді графіків та

діаграм. Створений веб-проект включає в себе інструменти для ручного введення даних про будівельні об'єкти, хоча вони не здатні значно спростити процес рутинного поповнення бази даних без використання таблиць у зручному для обробки вигляді. Ми розуміємо, що цей веб-застосунок має величезний простір для розвитку. При отриманні великої кількості достовірних даних він має певний потенціал стати довідковим веб-сайтом, надаючи окрім загальнодоступних відомостей точні прогнози, які здатні врятувати покупців та інвесторів. На поточному етапі розвитку надана нами веб-сторінка відображає прогнози, створені лише однією моделлю машинного навчання: логістичною регресією. Додавання стовпців для двох інших наборів результатів потребує часткового переосмислення існуючих рішень і не може відбутись дуже швидко.

Проаналізувавши достовірні дані про проблемні об'єкти будівництва у місті Києві ми можемо зробити висновок, що ситуація з недобудовами у столиці не є утішною, як і в більшості інших міст нашої держави, хоча прогнозуючі інструменти на перший погляд не обіцяють суттєвого погіршення встановленого тренду. Наявність подібних рішень в арсеналі будівників, інвесторів та покупців житла має спричинити позитивні зміни на ринку нерухомості.

4. Список використаних джерел

1. Теорія ймовірностей та математична статистика [Електронний ресурс]: підручник / Т.А. Ліхоузова. – Київ : НТУУ «КПІ ім. Ігоря Сікорського», 2018, с. 173-210.
3. Бахрушин В.Є. Методи аналізу даних: навчальний посібник для студентів.
4. Роб Дж Хиндман, Джордж Атанасопулос Прогнозирование: принципы и практика регресія для прогнозування часових послідовностей, розділ 5.
5. <https://my-realty.kiev.ua/doma/>
6. <https://kyivcity.gov.ua/>