

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК, ФІЗИКИ ТА
МАТЕМАТИКИ

КАФЕДРА ІНФОРМАТИКИ, ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА
ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

АЛГЕБРАЇЧНИЙ ПІДХІД ПРИ РОЗРОБЛЕННІ ЕКОНОМІЧНОЇ
МОДЕЛІ

Кваліфікаційна робота (проект)
на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 441 групи

Спеціальності 121 Інженерія
програмного забезпечення

Освітньо-професійної (наукової)
програми Інженерія програмного
забезпечення

Синицин Юрій Валерійович

Керівник кандидат фізико-математичних
наук, доцент кафедри вищої математики
Вейцблїт Олександр Йосипович

Рецензент Senior Software Engineer
(ЕРАМ) Кожевников Дмитро Іванович

Херсон – 2021

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Економічні моделі	5
1.1 Модель Леонтьєва	5
1.2 Модель Марковіца.....	8
1.3 Модель Кобба-Дугласа	10
РОЗДІЛ 2. Інструменти, використані в ході розробки проекту	13
2.1 Мова програмування Python	13
2.1 Бібліотеки chart.js та numpy	18
РОЗДІЛ 3. Розробка економічної моделі на мові python	21
ВИСНОВОК	29
ВИКОРИСТАНІ ДЖЕРЕЛА	30
ДОДАТКИ	32
Додаток А.....	32

ВСТУП

Для чого потрібні економічні моделі? Що вони дають людині? Функціонування багатогалузевого господарства вимагає балансу між галузями. Кожна з галузей є виробником, але і в той же час споживачем тієї продукції, яку випускають інші галузі. Балансова модель Леонтьєва вирішує непросту задачу розрахунку зв'язку між випуском і споживанням продукції галузей різного характеру. Вона побудована на алгебрі матриць і використанні матричного аналізу.

Щоб скористатися економіко – математичними моделями, їх простіше всього запрограмувати. Розробляючи програму я використав мову Python і його бібліотеки, такі як chart.js і NumPy. Про це докладніше розписано в тексті дипломної роботи.

Актуальність роботи. Балансова модель Леонтьєва розв'язує задачу розрахунку міжгалузевих зв'язків в економіці. Аналіз економічних процесів та прийняття відповідних рішень завжди значно вдосконалюється при їх візуалізації. Саме це реалізовано в цій дипломній роботі для моделі Леонтьєва.

Метою роботи є створення програмного додатку для візуалізації процесів міжгалузевих зв'язків згідно з моделлю Леонтьєва.

Об'єктом роботи є моделі економічних процесів та їх програмні реалізації.

Предметом роботи є програмний додаток для візуалізації моделі Леонтьєва на мові Python.

Завдання роботи:

1. Ознайомитися та проаналізувати економіко - математичні моделі, зокрема модель Леонтьєва.

2. Проаналізувати переваги та недоліки інструментів для розробки та реалізації програмного додатку для візуалізації моделі Леонт'єва.
3. Розробити та реалізувати відповідний програмний додаток.
4. Використані **інструменти**: мова Python, Django, HTML (bootstrap), CSS, JavaScript (jQuery) та Python бібліотеки NumPy та chart.js.

РОЗДІЛ 1 ЕКОНОМІЧНІ МОДЕЛІ

1.1 Модель Лєонтєєва

Функціонування багатогалузевого господарства вимагає балансу між господарствами. Кожна з галузей щось виробляє, але і в той же час - потребує ту продукцію, яку виробляють інше господарство. Ця модель вирішує складну задачу розрахунку зв'язку через випуск і споживання продукції галузей різного характеру. Вона ґрунтується на алгебрі матриць і використанні матричного аналізу.

Міжгалузеві балансові відносини

Для того щоб почати виробляти щось своє, знадобиться підтримка ресурсами іншого господарства. Найчастіше процес виробництва розглядається десь рік.

Запишемо:

x_i - єдиний обсяг продукції i -ї галузі.

x_{ij} - обсяг продукції i -ї галузі, споживані j -ї галуззю при виробництві продукції x_j .

y_i - обсяг продукції i -ї галузі, який потрібен для реалізації в невиробничій сфері, до якої ми відносимо: особисті споживання громадян, забезпечення державних інститутів, задоволення суспільних потреб і т.д.

Принцип балансу між різними господарствами базується на тому, що валовий випуск галузі i повинен дорівнювати сумі обсягів, який споживається в виробничих і невиробничих сферах. Балансове співвідношення в її простій формі має вигляд:

$$x_i = x_{i1} + x_{i2} + \dots + x_{in} + y_i, \quad i = 1, 2, \dots, n.$$

Тоді система рівнянь в матричній формі буде:

$$\bar{x} = A\bar{x} + \bar{y}$$

Це співвідношення називається рівнянням лінійного міжгалузевого балансу. Застосовуємо в двох випадках:

1. Коли нам відомий \bar{x} (вектор валового випуску) і ми повинні розрахувати \bar{y} (вектор кінцевого споживання);
2. Коли для T (період певного часу) відомий \bar{y} і потрібно знайти \bar{x} . В даному випадку потрібно вирішити систему лінійних рівнянь, де вже відома матриця A і заданий \bar{y} . І дуже важливо те, що всі елементи матриці A і \bar{x} і \bar{y} повинні бути позитивними.

Продуктивні моделі Леонтьєва

Якщо у матриці A елементи від'ємні, вона є *продуктивною*. Для \bar{y} з позитивними елементами існує рішення рівняння - \bar{x} , елементи якого так само позитивні, тоді модель Леонтьєва так само вважається *продуктивною*.

Розглянемо систему матриці E :

$$(E - A)\bar{x} = \bar{y}$$

Беремо до уваги існування матриці $(E - A)^{-1}$ (зворотної матриці), то в такому випадку є тільки одне рішення рівняння:

$$\bar{x} = (E - A)^{-1}\bar{y}$$

Ця матриця називається *матрицею повних витрат*.

Продуктивність матриці A можна розділити на два критерії:

1. Матриця A продуктивна тільки в тих випадках, коли присутня зворотна матриця з тільки позитивними елементами.

2. Якщо сума компонентів по стовпцю або рядку не перевищує одиниці, то матриця A є продуктивною.

1.2 Модель Марковіца

Зараз теорії фінансового ризику, які пов'язані з процентними ставками або фінансовими інструментами, тримаються на портфельній теорії Марковіца. Ризик в моделі визначається за допомогою побудови оптимального портфеля. Теоретична складова будується з теорії ймовірності та математичної статистики. Вигода від цінного паперу розглядається як випадкова величина, а прибутковість портфеля визначається завдяки випадковому вектору.

Так само можна розглянути пасивне володіння портефелем, тобто коли цінний папір купується на тривалий термін. Вкладник бере якийсь показник і створює портфель, який буде показувати прибутковість цінного паперу відповідно до динаміки цього показника.

Модель припускає об'єднання цінних паперів в портфель, який забезпечить нам мінімальний ризик при заданій прибутковості. Портфель, що має мінімальний рівень ризику при заданій прибутковості називається ефективним множеством.

Вкладення в ризиковий папір j -го виду позначається x_j , ($j = 1, \dots, n$), де n - кількість видів цінних паперів в портфелі. Для того, щоб обчислити дисперсію і очікуваний прибуток портфеля, ми формуємо завдання вибору оптимальної структури ризикового портфеля, яка формується таким чином:

Знаходимо вектор $X = (x_1, x_2, \dots, x_n)$ який згладжує дисперсію портфеля

$$V_p = \sum_{i=1}^n \sum_{j=1}^n V_{ij} x_i x_j ,$$

при наявності обмежень:

$$\bar{m}_p = \sum_{i=1}^n \bar{m}_i x_i,$$

$$\sum_{i=1}^n x_i = 1$$

Так задача Марковіца схожа на умовну задачу екстремуму і вирішується методом Лагранжа.

Далі ми представимо рівняння у вигляді матриці, використовуючи такі позначення:

$V = (V_{ij})$ – коваріаційна матриця розмірністю $n \times n$.

m – вектор стовпчик передбачуваної прибутку розмірністю $n \times 1$.

I – одиничний вектор стовпчик розмірністю $n \times 1$.

X – вектор стовпчик невідомих часткою розмірністю $n \times 1$.

Метою завдання Марковіца в матричній формі – знайти \bar{X} , який пом'якшить дисперсію портфеля:

$$V_p = X^T V X,$$

З огляду на обмеження:

$$m^T X = \bar{m}_p, \quad I^T X = 1.$$

Тепер напишемо функцію Лагранжа для задачі:

$$L(X, \lambda_1, \lambda_2) = X^T V X + \lambda_1 (m^T X - \bar{m}_p) + \lambda_2 (I^T X - 1),$$

де $X^T V X$ - цільова функція, екстремум якої можна визначити;

$\lambda = (\lambda_1, \lambda_2)^T$ - вектор стовпчик множників Лагранжа.

Щоб визначити екстремум цієї функції, потрібно продифференціювати всі змінні і прирівняти їх похідні до 0

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial x_1} = 2x_1V_{11} + 2x_2V_{12} + \dots + 2x_nV_{1n} + \lambda_1\bar{m}_1 + \lambda_2 = 0; \\ \frac{\partial L}{\partial x_2} = 2x_1V_{21} + 2x_2V_{22} + \dots + 2x_nV_{2n} + \lambda_1\bar{m}_2 + \lambda_2 = 0; \\ \dots \dots \dots \\ \frac{\partial L}{\partial x_n} = 2x_1V_{n1} + 2x_2V_{n2} + \dots + 2x_nV_{nn} + \lambda_1\bar{m}_n + \lambda_2 = 0; \\ \frac{\partial L}{\partial \lambda_1} = x_1\bar{m}_1 + x_2\bar{m}_2 + \dots + x_n\bar{m}_n - \bar{m}_p = 0; \\ \frac{\partial L}{\partial \lambda_2} = x_1 + x_2 + \dots + x_n - 1 = 0. \end{array} \right.$$

Тепер це рівняння напишемо в матричній формі:

$$\begin{pmatrix} 2V_{11} & 2V_{12} & \dots & 2V_{1n} & \bar{m}_1 & 1 \\ 2V_{21} & 2V_{22} & \dots & 2V_{2n} & \bar{m}_2 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 2V_{n1} & 2V_{n2} & \dots & 2V_{nn} & \bar{m}_n & 1 \\ \bar{m}_1 & \bar{m}_1 & \dots & \bar{m}_n & 0 & 0 \\ 1 & 1 & \dots & 1 & 0 & 0 \end{pmatrix} x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \bar{m}_p \\ 1 \end{pmatrix}$$

Далі наше завдання вирішити систему: $AX = B$, де $X = A^{-1}B$,

A^{-1} – зворотна матриці A . Це дозволить визначити структуру кращого портфеля n видів цінних паперів, який покаже кращій прибуток при меншій дисперсії.

1.3 Модель Кобба-Дугласа

Модель Кобба-Дугласа була сформульована за допомогою перетворення простої виробничої функції $Y = F(L, K)$. Ця модель показує, якою часткою нагороджується людина, що бере участь в створенні сукупного продукту. Її можна представити у вигляді:

$$Y = A K^\alpha L^\beta,$$

де α має значення від 0 до 1, а $\beta = 1 - \alpha$.

L – Праця;

K – Капітал;

A – коефіцієнт, що показує рівень технологічної продуктивності.

α і β - коефіцієнти гнучкості обсягу продукції, що випускається за фактором виробництва: α – за капіталом, β – по праці. Так можна помітити, що кожен фактор, який оплачується з кінцевим продуктом, то ці чинники демонструють частки капіталу і праці в спільному доході. Частка капіталу в доході буде дорівнювати αY , а частка праці в доході - βY . Так як $\beta = 1 - \alpha$, то $\alpha + \beta = 1$, з цього робимо висновок, що маємо справу з постійною віддачею від масштабу. Ми можемо розглянути емпіричні параметри моделі Кобба-Дугласа, де $A=1,1$; $\alpha = 1/4$; $\beta = 3/4$, а значить частка капіталу складе 25%, частка праці - 75%.

Коли ми шукаємо найбільшу ефективність виробництва, нас повинні цікавити фактори, які беруть участь в граничному виробництві, які і визначають прийнятний обсяг вживаних ресурсів.

Граничний продукт капіталу пропорційний відношенню частки капіталу в доході до обсягу використаного капіталу: $MP_K = \alpha Y/K$. Таким же чином ми визначаємо граничну продуктивність праці: $MP_L = \beta Y/L$.

Розглянемо кілька властивостей виробничої функції Кобба-Дугласа:

1. **Сталість віддачі від масштабу.** Воно описується наступною формулою $F(nK, nL) = n A K^\alpha L^\beta$, яка демонструє те, що при збільшенні праці і капіталу в n раз, обсяг доходу відповідно зросте в n раз.
2. **Зміна граничної продуктивності факторів.** Припустимо, якщо ми залучимо в наше виробництво додатковий капітал K , а праця L залишиться колишнім, то гранична продуктивність праці MP_L збільшиться, а гранична продуктивність збільшеного обсягу капіталу

MP_K зменшиться. Аналогічно подія відбудеться, якщо ми збільшимо кількість праці. Виходячи з чого, ми можемо зробити висновок, що при порушенні пропорцій капіталу і праці призводить до неефективності виробництва.

З іншого боку, якщо ми збільшимо коефіцієнт A , то ми побачимо зростання MP_K і MP_L , що означає досить швидке економічне зростання.

РОЗДІЛ 2 ІНСТРУМЕНТИ, ВИКОРИСТАНІ В ХОДІ РОЗРОБКИ ПРОЕКТУ

2.1 Мова програмування Python

Python - це інтерпретована, інтерактивна та об'єктно-орієнтована мова сценаріїв високого рівня. Він був створений для високої читабельності.

Python обробляється під час виконання інтерпретатором. Нам не потрібно компілювати нашу програму перед її запуском. Цими функціями він дуже схожий з Perl і PHP.

Ми можемо взаємодіяти з інтерпретатором, який видає нам підказки і вказує на помилки в коді, показуючи потрібно рядок і підсвічуючи місця з помилками.

Мова підтримує об'єктно-орієнтована стиль і техніку програмування, яка інкапсулює код всередині об'єктів.

Так само ця мова дуже добре підходить для новачків-програмістів.

Перерахуємо кілька особливостей мови Пайтон:

Легкий у навчанні: У Python є кілька ключових слів, проста структура і чітко визначена синтаксис, що дозволяє абсолютно будь-якому користувачеві швидко вникнути і вивчити цю мову.

Простий в обслуговуванні: вихідний код на Python досить простий в обслуговуванні.

Широкі стандартні бібліотеки: більшість бібліотек на Python досить таки портативні і кроссплатформени.

Інтерактивний режим: Python підтримує інтерактивний режим, завдяки якому ми можемо інтерактивно тестувати код і робити налагодження різних фрагментів коду.

Переносний: мова працює на більшості апаратних платформ з незмінним інтерфейсом.

Висувний: ми можемо додати будь-які низькорівневі модулі до інтерпретатора Python. Ці модулі дозволяють програмістам додавати і налаштовувати свої інструменти для більшої ефективності.

Робота з базами даних: Python дозволяє працювати з інтерфейсом будь-якої комерційної бази даних.

Програмування графічного інтерфейсу: Python може працювати з додатками, які мають графічний інтерфейс, який ми в свою чергу можемо створювати і переміщати на багато системних бібліотек та системних вікон, наприклад: Windows MFC, Macintosh і система X Window в Unix.

Масштабна: Python забезпечує кращу структуру і підтримку великих програм.

Але, як це завжди буває, скільки б не було позитивних сторін, завжди будуть і негативні.

Нижче перерахуємо недоліки мови:

- Перше, що можна відзначити, це продуктивність. Python не настільки швидкодіючий, на скільки хотілося б. Це пов'язано з тим, що мова, в першу чергу, інтерпретована і в порівнянні з іншими інтерпретованими мовами ця, особливо, помітна.
- Далі можна відзначити синтаксис. Хоч ми і зазначали раніше, що він досить таки простий і легко освоюється, але якщо програміст переходить з іншої мови на цей, то для нього він буде незвичний.

- Динамічна типізація. Через це Python їсть більше ресурсів, ніж нам хотілося б.

У висновку можна сказати, що беручи до уваги все вищесказане, всі ці недоліки перекриваються перевагами мови. Так само з огляду на, що мова програмування розвивається, незабаром ці недоліки зникнуть і переростуть в переваги.

Python Django

Django - високорівневий веб-фреймворк Python, який відкриває можливість створювати безпечні сайти і підтримувати їх. Він бере на себе більшу частину веб-розробки. Django лежить у вільному доступі і має відкритий вихідний код. Завдяки великій кількості ком'юніті, в інтернеті можна знайти багато документації та підтримки продукту, як платної, так і безкоштовної.

Веб-фреймворк дає нам наступне:

Повноту

Django включає в себе абсолютно все - це його основна філософія. Він надає розробникам буквально все, щоб зробити що-небудь з нуля. І оскільки це є частиною одного продукту, все це чудово доповнює один одного та відповідає принципам проектування і має в наявності велику базу документації.

Багатогранність

Django універсальний. Він використовується для створення практично будь-якого виду сайту: від систем, які дозволяють нам управляти контентом сайту, до створення новинних сайтів і навіть соціальних мереж. Ще він може мати справу з різним клієнтським середовищем і відправляти медіа-продукт в будь-якому форматі.

Безпека

Розробники, які використовують даний веб-фреймворк, можуть не хвилюватися про даний пункт. Django надає своїм користувачам надійний спосіб для управління обліковими записами користувачів і пароллями. І ми не зіткнемося з такою поширеною помилкою, як розміщення інформації в файлах cookie, де, власне, ця інформація найбільше вразлива. Так само веб-фреймворк пропонує хороший захист від більшості вразливостей, такі як:

- міжсайтовий скриптинг;
- SQL-ін'єкція;
- підробка запитів між сайтами;
- клікджекінг.

Масштабованість

Django використовує архітектуру shared-nothing, кожен з компонентів якої незалежний від інших компонентів. Це означає, що він масштабується, коли зростає трафік, за допомогою додавання обладнання на будь-якому з рівнів:

- сервер кешування
- сервер додатку
- сервер бази даних

Портативність

Django створений за допомогою мови Python, який в свою чергу є кросплатформним. Це не прив'язує нас до будь-якої серверної платформи і ми можемо спокійно запускати додатки на багатьох версіях. Так само фреймворк підтримують багато веб-хостингів.

Само собою перерахуємо і **мінуси** веб-фреймворку:

1. Django досить складний. Після кожного оновлення працювати з ним стає важче для розробників. Django ORM, на відміну від його аналогів, складніше адаптувати під свій проект і в добавок це потрібно робити відносно часто.
2. Веб-фреймворк занадто нерозривний.
3. Частини програми розгортаються разом.
4. Так само він використовує шаблон маршрутизації із зазначенням URL.

JavaScript (jQuery)

JavaScript - це мова програмування, яка дозволяє розробникам додавати щось складне на веб-сторінку. Під складним можна мати на увазі те, коли, припустимо, користувач заходить на будь-який веб-сайт, шукає на ньому інформацію, в цей час на сайті оновлюється контент. І завдяки цій мові, сайт не залишається статичним, а оновлюється в реальному часі для користувача. У наш час практично кожен сайт написаний з використанням JavaScript.

JavaScript – це інтерпретована мова, яка розроблена для того, щоб взаємодіяти з веб-сайтами.

Мова надає нам такий спектр можливостей:

- анімувати сайт;
- перевіряти введення даних до відправки на сервер;
- визначати інтернет-браузер;
- JavaScript може керувати тегами hltm, змінювати їх, додавати і змінювати стилі;
- зчитувати дані про комп'ютер користувачів через cookie-файли.

Недоліки мови:

- якщо вікна або вкладки були відкриті не за допомогою JavaScript, то він не зможе їх закрити;
- не здатний захистити вихідний код;
- Javascript не може заборонити користувачу копіювати текст або зображення зі сторінок;
- якщо сайт розташований на іншому домені, мова не зможе отримати до нього доступ;
- не діє за рамки веб-сторінки, тобто він не зможе мати доступ до файлів комп'ютера користувача.

jQuery - це досить таки швидка і багатофункціональна JavaScript бібліотека. В її можливості входить наступне:

- уміння обходити і управляти HTML файлами;
- обробляти події;
- включає в себе анімацію;
- Робить Ajax простіше у використанні API.

Завдяки універсальності і частим оновленням спрощує роботу з JavaScript для всіх розробників.

2.1 Бібліотеки **chart.js** та **numpy**

При створенні свого проекту, я використовував дані бібліотеки, для чого вони потрібні і як використовуються, поясню далі.

Бібліотека chart.js

Ця бібліотека відноситься до JavaScript-бібліотек, яка дозволяє нам створювати різного роду графіки та діаграми при використанні canvas (HTML5). Можна створювати:

- лінійні графіки;

- кругові діаграми;
- столбцові графіки і т.д.

Так само `chart.js` підтримує анімацію об'єктів.

Найбільшим плюсом є розмір бібліотеки, вона займає мало місця. Але навіть не дивлячись на це ми можемо зменшити її розмір, беручи тільки ті модулі, які ми збираємося використовувати. Припустимо, нам потрібно створити якусь певну діаграму або графік. Ми підключаємо основну бібліотеку – `chart.js` і сам модуль, за допомогою якого і створюється цей графік або діаграма. Також можна виділити ще одну особливість бібліотеки - динамічність, тобто якщо ми зменшимо розмір вікна браузера, то і вона змінить свій розмір.

Бібліотека безкоштовна і її можна завантажити з офіційного сайту. Після того, як ми підключили її до нашого проекту, ми можемо звертатися до неї через глобальну змінну - `Chart`. Ну і щоб намалювати графік або діаграму, ми повинні розмістити об'єкт `canvas` і ще створити `chart`-об'єкт і передати йому 2д-контекст `canvas`.

Бібліотека `numpy`

`Numpy` міститься в бібліотеці `Python`, вона дозволяє розробникам користуватися матриця і багатовимірними масивами, а так же безліччю математичних функцій, завдяки яким можна виконувати операції над масивами.

Основою цієї бібліотеки є `numpy.ndarray` (однорідний багатовимірний масив).

Цей масив має наступні атрибути:

`ndarray.ndim` – деяке число, яке вимірює масив.

ndarray.shape – показує нам розмір масиву, тобто його довжину на кожній осі

ndarray.size – показує нам елементи масиву, а точніше їх кількість.

ndarray.dtype – функція, яка виводить тип компонентів масиву. Dtype можна визначити за допомогою стандартних типів, які є в Python, але в numpy є вбудовані типи, що дозволяє нам точніше визначити тип, наприклад:

- character;
- bool;
- int16;
- int32;
- int64;
- float16;
- float32;
- float64;
- object.

ndarray.itemsize - розмір кожного елемента масиву в байтах.

ndarray.data - буфер, який містить фактичні елементи масиву. Зазвичай не потрібно використовувати цей атрибут, так як звертатися до елементів масиву найпростіше за допомогою індексів.

РОЗДІЛ 3 РОЗРОБКА ЕКОНОМІЧНОЇ МОДЕЛІ НА МОВІ PYTHON

Запрограмувавши модель Леонтьєва, я вирішив, що легше всі отримані дані у вигляді табличних розрахунків матриць і відображення графіків буде помістити в односторінковому сайті. Для цього я використовував наступні інструменти:

Сервер Python Django;

- Frontend – HTML (bootstrap), CSS, JavaScript (jQuery);
- Бібліотеки – NumPy та chart.js.

Сайт, як я і писав раніше - односторінковий. На ньому користувач може побачити матрицю А (технологічну матрицю):

Таблиця А

N	1	2	3	4	5	6	7	8	9
1	0.3095	0.0065	0.0026	0.0405	0.0008	0.0318	0.0024	0.0451	0.0004
2	0.0003	0.0242	0.0003	0.0002	0	0.0024	0	0.0034	0
3	0.0077	0.0027	0.0303	0.1474	0.3578	0.7004	0.0011	0.0033	0.07
4	0.2197	0.1017	0.0813	0.3787	0.1579	0.8201	0.077	0.3563	0.2229
5	0.0138	0.0053	0.0383	0.0286	0.0988	0.2344	0.0075	0.0336	0.0418
6	0.0006	0.0003	0.0018	0.0011	0.006	0.3057	0.0012	0.0051	0.0028
7	0.136	0.0863	0.0328	0.2122	0.002	0.1696	0.0485	0.0272	0.0155
8	0.0006	0.0009	0.0011	0.0016	0.0046	0.1096	0.0073	0.0091	0.0068
9	0.0468	0.0316	0.0649	0.0587	0.0237	0.7816	0.0847	0.0182	0.1097

Рисунок 3.1 - Технологічна матриця А (вхідні дані)

Та матрицю Е, яка також містить в собі вхідні дані, завдяки яким і буде розраховуватись далі наша модель.

Таблиця Е

N	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

Рисунок 3.2 - Матриця Е

Інші ж дані не підраховуються. Якщо користувачеві потрібно дізнатися результати підрахунків в інших таблицях, йому знадобиться ввести 9 галузей і їх кінцевий попит вручну на сайті. Тому що це також вхідні дані. Так як матриці розмірностей $9 * 9$, то і галузей потрібно ввести рівно 9, не більше не менше. Інакше підрахунку інших таблиць не буде.

Натиснувши на кнопку «Додати галузь», в табличці вибивається пусті поля, де у першому ми вказуємо назву галузі економіки України, а у другому їх кінцевий попит. В колонці «Actions» ми маємо дві активні кнопки «Відредагувати», «Видалити», та зелена кнопка «Додати». Тобто якщо в полі «Назва галузі» буде якась помилка, будь-то орфографічна, або вказана не та галузь, то це можна швидко виправити. Теж саме стосується і поля «Кінцевий попит». Червона кнопка «Видалити» повністю видалить строку з таблички. Ну і звісно зелена кнопка додає до таблички введені дані користувачем.

Вот як виглядають дані, які вводяться користувачем через клавіатуру:

+ Додати галузь

Галузі економіки України	Кінцевий попит Y, тис. грн.	Actions
Риболовство	40000	
<input style="width: 95%; height: 20px;" type="text"/>	<input style="width: 95%; height: 20px;" type="text"/>	

Рисунок 3.3 - Таблиця галузей економіки та їх кінцевого попиту

Якщо говорити з мого боку, я маю справу з вихідними даними таблиці А. Варіацій цих даних може бути кілька. Ці дані зберігаються на диску і можуть бути замінені, якщо цього вимагає випадок.

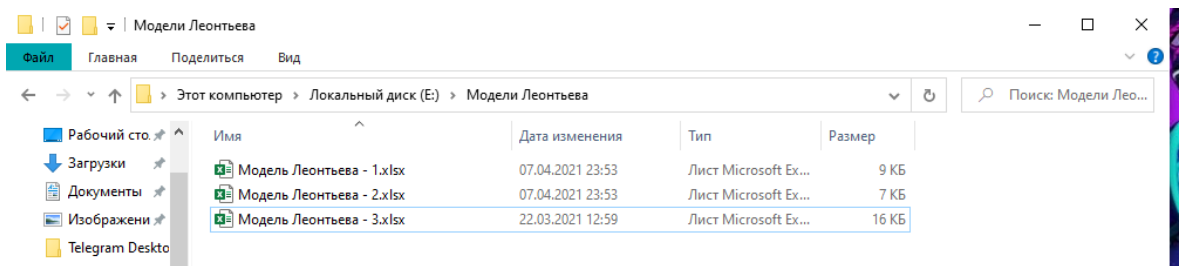


Рисунок 3.4 - Файли з різними вихідними даними матриці А

Вводячи дані в таблицю А я брав значення, взяті з цих таблиць Excel.

У програмі таблиці А і Е введені відразу, а таблиця з галузями і їх кінцевим попитом вводяться вручну. Так ось, чому в ці дві таблиці можна відразу вписувати свої значення. Я порахував, що так як матриці мають розмірності 9*9, створення таблиці з input fields цієї розмірності несло б за собою смислове і логічне навантаження. Тому я прийняв рішення помістити ці таблиці в json file. При потребі значення можна поміняти там.

Повертаючись до таблиці А, приведу приклад, котрий я брав з таблиці Excel.

ПОБУДОВА МОДЕЛІ «ВИТРАТИ-ВИПУСК» ДЛЯ ГАЛУЗЕЙ ЕКОНОМІКИ УКРАЇНИ (модель Леонтьєва)									
A	0,3095	0,0065	0,0026	0,0405	0,0008	0,0318	0,0024	0,0451	0,0004
	0,0003	0,0242	0,0003	0,0002	0	0,0024	0	0,0034	0
	0,0077	0,0027	0,0303	0,1474	0,3578	0,7004	0,0011	0,0033	0,07
	0,2197	0,1017	0,0813	0,3787	0,1579	0,8201	0,077	0,3563	0,2229
	0,0138	0,0053	0,0383	0,0286	0,0988	0,2344	0,0075	0,0336	0,0418
	0,0006	0,0003	0,0018	0,0011	0,006	0,3057	0,0012	0,0051	0,0028
	0,136	0,0863	0,0328	0,2122	0,002	0,1696	0,0485	0,0272	0,0155
	0,0006	0,0009	0,0011	0,0016	0,0046	0,1096	0,0073	0,0091	0,0068
	0,0468	0,0316	0,0649	0,0587	0,0237	0,7816	0,0847	0,0182	0,1097
	0,735	0,2595	0,2534	0,869	0,6516	3,1556	0,2297	0,5013	0,4699

Рисунок 3.5 - Приклад матриці A, побудованої в Excel

Повторює, таких таблиць 3, з різними даними. За необхідністю можна долучатись до них.

Якщо говорити про сервер, на якому знаходиться сайт, я використав локальний сервер Django. Щоб його запустити, в консолі треба прописати наступну команду:

```
python manage.py runserver
```

В браузері через url ми звертаємось до сервера, який в свою чергу видає нам view для url, тобто html template, до якого підвантажується JS script.

Так ось, якщо повертатися до програми, а саме до таблиць, які потрібно розрахувати. Після введення 9 галузей програма зробить підрахунки всіх порожніх таблиць, за якими ми зможемо зробити будь-якої висновок, виходячи з отриманих даних.

Приклад таблиць, які ще не підраховані:

Таблиця $V=(E - A)^{-1}$

N	1	2	3	4	5	6	7	8	9

Таблиця X-Y

N	1

Рисунок 3.6 - Не розраховані таблиці (до введення 9 галузей)

Коли ми введемо всі потрібні нам галузі і значення їх кінцевого попиту, програма розрахує всі таблиці.

+ Додати галузь

Галузі економіки України	Кінцевий попит Y, тис. грн.	Actions
Сільське господарство	67681	 
Рибальство	2926	 
Добувна промисловість	10460	 
Переробна промисловість	372196	 
Виробництво електроенергії	16278	 
Будівництво	2042	 
Торгівля	4524	 
Готельна діяльність	10390	 
Галузь транспорту і зв'язку	42860	 

Рисунок 3.7 - Заповнена таблиця галузей і кінцевого попиту

Далі ми побачимо розрахунки в таблиці В, в котрій ми визначили мультиплікативне зростання випуску в кожній галузі України при умові, що кінцевий попит збільшиться на 1 грн.

Таблиця В=(E - A)^-1

N	1	2	3	4	5	6	7	8	9
1	1.49111	0.02476	0.01826	0.11297	0.03181	0.2912	0.01743	0.11231	0.03396
2	0.00068	1.02489	0.00041	0.00057	0.00033	0.00567	0.00011	0.0038	0.00024
3	0.15608	0.05414	1.09952	0.33656	0.51498	1.9562	0.05479	0.16474	0.20333
4	0.68419	0.23892	0.22519	1.85877	0.45623	3.36999	0.21272	0.74936	0.52483
5	0.06399	0.02159	0.06134	0.0902	1.15705	0.67669	0.02569	0.08061	0.08495
6	0.00474	0.00176	0.00439	0.00636	0.01288	1.4675	0.00322	0.01073	0.0073
7	0.37521	0.15331	0.09359	0.44724	0.13111	1.15902	1.1058	0.22226	0.15025
8	0.00698	0.00332	0.00381	0.00926	0.00994	0.19415	0.00991	1.01468	0.01162
9	0.1766	0.07415	0.11044	0.20377	0.12409	1.80088	0.12785	0.12091	1.19764

Рисунок 3.8 - Таблиця В

Далі за формулою ми обчислюємо матрицю галузевого випуску України, тобто множимо матрицю В на Y (таблиця галузей та їх кінцевого попиту). І отримуємо наступні показники:

1. Сільське господарство - 151230.7588;
2. Рибальство - 3398.01985;
3. Добувна промисловість - 220570.7951;
4. Переробна промисловість - 839225.05696;
5. Виробництво електроенергії - 136904.24662;
6. Будівництво - 27715.0991;
7. Торгівля - 226717.15162;
8. Готельна діяльність - 12598.92064;
9. Галузь транспорту і зв'язку - 140954.53159.

Далі ми розраховуємо величину проміжної продукції по кожній галузі. Формула наступна – $X - Y$. Отримуємо наступні результати:

1. Сільське господарство - 79362,65796;
2. Рибальство - 401,55064;
3. Добувна промисловість - 160081,99795;
4. Переробна промисловість - 414541,88328;
5. Виробництво електроенергії - 47139,98132;
6. Будівництво - 4341,50140;
7. Торгівля - 207012,76215;
8. Готельна діяльність - 5221,83488;
9. Галузь транспорту і зв'язку - 105168,90397.

Наступні дві таблиці результативні. Вони будуть відповідати на питання – «Чи є галузь рентабельною?» та «Чи є галузь продуктивною?».

Таблиця R – таблиця рентабельності галузей. Вона розраховується за

формулою: $R_j = 1 - \sum_{i=1}^n a_{ij}$, $j = 1, 2, \dots, n$.

Отримуємо наступні результати и робимо висновок:

1. Сільське господарство - 0,26500;
2. Рибальство - 0,74050;
3. Добувна промисловість - 0,74660;
4. Переробна промисловість - 0,13100;
5. Виробництво електроенергії - 0,34840;
6. Будівництво - -2,15560;
7. Торгівля - 0,77030;
8. Готельна діяльність - 0,49870;
9. Галузь транспорту і зв'язку - 0,53010.

Дивлячись на показники, можна сказати наступне: Будівництво - не є рентабельною галуззю, іншими словами, вона не відображає того рівня ефективності використання грошових коштів, трудових коштів, матеріалів природного і штучного характеру.

І остання таблиця π , яка обчислює прибуток кожної галузі и дає нам відповідь на запитання, чи є ця галузь прибутковою. Вона розраховується за формулою: $\pi_j = R_j \cdot X_j$, $j = 1, 2, \dots, n$.

Результати:

1. Сільське господарство - 38966,56936;
2. Рибальство - 2464,05125;
3. Добувна промисловість - 127326,65567;
4. Переробна промисловість - 103062,66271;
5. Виробництво електроенергії - 22094,82469;
6. Будівництво - -13760,27561;
7. Торгівля - 162946,76788;

8. Готельна діяльність - 7785,62205;
9. Галузь транспорту і зв'язку - 78470,12199.

Висновок: тут все зрозуміло, як і в попередній таблиці, знов галузь Будівництво має від'ємний показник. Тобто ця галузь потребує великих вкладень коштів, але в замін не пропонує ніякого прибутку, а навіть забирає її.

ВИСНОВОК

Розробка економічних моделей має на увазі собою аналіз і прогнозування ситуації в економіці. Ці моделі можуть застосовувати люди з різних сфер, будь-то підприємець або ж людина, яка просто цікавиться цією темою, до великих підприємств і держав.

Моделюючи економічну складову, ми таким чином спрощуємо схеми процесів в певній області економіки і робимо акцент на важливих чинниках, які так чи інакше впливають на нашу модель. Були виконані поставлені завдання, а саме я

1. Ознайомився та проаналізував економіко - математичні моделі, зокрема модель Леонтьєва.
2. Проаналізував переваги та недоліки інструментів для розробки та реалізації програмного додатку для візуалізації моделі Леонтьєва.
3. Розробив та реалізував відповідний програмний додаток.
4. Використані **інструменти**: мова Python, Django, HTML (bootstrap), CSS, JavaScript (jQuery) та Python бібліотеки NumPy та chart.js.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Модель Леонтьєва. Е. Харламова, Д. Філонович (ГІУСТ БГУ). Науковий керівник: кандидат фіз.-мат. наук, доцент Л.Г. Третьякова. URL: <https://core.ac.uk/download/pdf/38539337.pdf>
2. Василь Леонтьєв у світовій і вітчизняній економічній науці - Гранберг А.Г. URL: https://elib.hse.ru/e-resources/HSE_economic_journal/articles/10_03_06.pdf
3. Міжгалузева методологія як інструмент економічного аналізу складних систем. URL: http://www.vectoreconomy.ru/images/publications/2017/5/mathematicalmethods/Ghazwan_Marakhovsky_Shadchneva3.pdf
4. Гаррі Марковіц: Вибір портфеля. The Journal of Finance, том 7, №1, стор. 77-91, березень 1952 р. URL: <https://assetallocation.ru/portfolio-selection/>
5. Модель Марковіца: математичні аспекти та комп'ютерна реалізація. URL: <https://cyberleninka.ru/article/n/model-markovitsa-matematicheskie-aspekty-i-kompyuternaya-realizatsiya/viewer>
6. Модель оптимального портфельного інвестування – Л. В. Борисова, І.Д. Сагаєва, Навчально-методичний посібник. URL: http://elibrary.sgu.ru/uch_lit/1347.pdf
7. Прогнозування зайнятості за допомогою функції Кобба-Дугласа. В.А. Гневашева. URL: https://mosgu.ru/nauchnaya/ZPU/2005_1/Gnevasheva/index.pdf
8. Виробнича функція Кобба-Дугласа як інструмент політики економічного зростання України в умовах ринкових реформ. Cherkashyna Tetiana Simon Kuznets Kharkiv National University of Economics. URL: <https://www.economyandsociety.in.ua/index.php/journal/article/download/26/25/>

9. Сафрончук М.В. Економічне зростання (гл.25, параграфи 1-6) // Курс економічної теорії - К: АСА, 2004. - С. 605-644
URL:<https://mgimo.ru/upload/iblock/e5e/e5e768174f0636120f106bf3d897006a.pdf>
10. Основи програмування на прикладі мови Python. Д.Ю. Федоров - навчальний посібник. URL: <http://dfedorov.spb.ru/python3/book.pdf>
11. Command line and environment Python. URL: <https://docs.python.org/3/using/cmdline.html>
12. Мова програмування Python: переваги, недоліки та область застосування. URL: https://skillbox.ru/media/code/dlya_chego_nuzhen_python/
13. Django documentation. URL: <https://docs.djangoproject.com/en/3.2/>
14. Робота з базою даних: моделі Django. URL: <https://djbook.ru/ch05.html>
15. Django. Кэшування. URL: <https://djbook.ru/ch13.html>
16. JavaScript. URL: <https://developer.mozilla.org/ru/docs/Learn/JavaScript>
17. Введення в jQuery. URL: <https://html5book.ru/vvedenie-v-jquery/>
18. Бібліотека Chart.js для WEB. URL: <http://java-online.ru/chartjs.xhtml>
19. Chart.js documentation. URL: <https://www.chartjs.org/docs/latest/>
20. Початок роботи з бібліотекою NumPy. URL: <https://pythonworld.ru/numpy/1.html>
21. NumPy documentation. URL: <https://numpy.org/doc/stable/user/whatisnumpy.html>

ДОДАТКИ

Додаток А

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Синицин Юрій Валерійович, учасник(ця) освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

– дотримуватися:

- вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
- принципів та правил академічної доброчесності;
- нульової толерантності до академічного плагіату;
- моральних норм та правил етичної поведінки;
- толерантного ставлення до інших;
- дотримуватися високого рівня культури спілкування;

– надавати згоду на:

- безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
- оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
- використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;
 - самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
 - надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
 - не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
 - своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
 - не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
 - підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
 - поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
 - не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
 - відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;
 - запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
 - не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
 - не підроблювати документи;
 - не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
 - не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
 - не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
 - не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
 - не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
 - не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;
 - не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

_____ (дата)

_____ (підпис)

_____ (ім'я, прізвище)