

ДОДАТОК А

Фрагменти коду програми

```

export default class App extends React.Component {

  constructor(props) {
    super(props);
    this.state = {
      isModalShown: false,
      isImageSearch: false,
      searchItems: [],
      selectedItem: null,
      isMostPopularFilter: false,
      isBestPriceFilter: false
    }
    this.items = [];
  }

  componentDidMount() {
    fetch('/get-items').then(res => res.json()).catch(() =>
      console.error('DB is not available')
    ).then(result => {
      console.log(result);
      this.setState({ searchItems: result.filter(el => el.type ===
"Jacket"))});
      this.items = result;
    })
  }

  toggleModal = (item) => {
    console.log(item);
    this.setState((prevState) => ({
      isModalShown: !prevState.isModalShown,
      selectedItem: item
    })))
  }

  toggleImageSearch = () => {
    this.setState((prevState) => ({
      isImageSearch: !prevState.isImageSearch
    })))
  }

  onLoad = (fileParams) => {
    console.log(fileParams);
  }

  onFileRevert = (fileParams) => {
    console.log(fileParams);
    this.toggleImageSearch()
  }

  selectFilter = (filterType) => {

```

```

    const prop = filterType === 'popular' ? 'isMostPopularFilter' :
    'isBestPriceFilter';

    return () => {
      this.setState({
        [prop]: !this.state[prop]
      })
    }
  }

  onSearchInput = (e) => {
    const targetValue = e.target.value;
    this.setState({
      searchItems: this.items.filter((el) => {
        return
        el.title.toLocaleLowerCase().search(targetValue.toLocaleLowerCase()) > 0;
      })
    })
  }

  render() {
    const { selectedItem, isMostPopularFilter, isBestPriceFilter } =
    this.state;
    const popularClass = `label-filter ${isMostPopularFilter ?
    'selected': ''}`;
    const priceClass = `label-filter ${isBestPriceFilter ? 'selected' :
    ''}`;
    return (
      <div className="App">
        <h1> Search aggregator</h1>
        <div className='search-block'>
          <input type='text' placeholder='Search'
          className='search' onChange={this.onSearchInput}/>
          {
            !this.state.isImageSearch &&
            <FcAddImage size='3rem'
            onClick={this.toggleImageSearch}></FcAddImage>
          }
          <span className={popularClass}
            onClick={this.selectFilter('popular')}>
            Most Popular</span>
          <span className={priceClass}
            onClick={this.selectFilter('price')}>
            > Best Price</span>

        </div>
        {
          this.state.isImageSearch && <FilePond
            labelIdle='Drag & Drop your picture or <span
            class="filepond--label-action">Browse</span>'
            acceptedFileTypes={['image/*']}
            ref={ref => (this.pond = ref)}
            allowMultiple={false}
            maxFiles={1}
            server="/search-picture"
            onupdatefiles={this.onLoad}
            onprocessfilerevert={this.onFileRevert}

```

```

        />
    }
    <div className='main'>
        {this.state.searchItems.map((item) => <div
className='item' key={item._id}>
            <div className='content' onClick={() =>
this.toggleModal(item)}>
                <img src={item.photos[0]} className='item-
img' />
                <div className='pre-view-name'>
                    <b> $ {item.price}</b> <br/>
                    {item.title}
                </div>
            </div>
        </div>
    )}
</div>

<div className={
    `modal ${ this.state.isModalShown ? 'display' : ''}`
}>
    { selectedItem && <div className='modal-content'>
        <div className='img-content'>
            <img src={selectedItem.photos[0]}
className='item-img' />
        </div>
        <div className='description-content'>
            <h2>{selectedItem.title}</h2>
            <h3> $ {selectedItem.price} </h3>
            <article>
                {selectedItem.description}
            </article>
            <a className='btn' href={selectedItem.url}
target='_blank'>Go to source</a>
        </div>
        <div className='close' onClick={() =>
this.toggleModal(null)} >x</div>
    </div>
}
</div>
);
}
}

async function downloadSearchPage(page = 1) {
    const options = {
        uri: `https://www.asos.com/men/jumpers-
cardigans/cat/?cid=7617&nlid=mw|clothing|shop%20by%20product&page=${page}`,
        headers: {
            cookie: 'geocountry=UA; check=true;
AMCVS_C0137F6A52DEAFCC0A490D4C%40AdobeOrg=1; browseCountry=UA;
browseCurrency=GBP; browseLanguage=en-GB; browseSizeSchema=EU;
storeCode=ROW; currency=1; asos-gdpr22=true; s_cc=true;
plp_columnsCount=fourColumns;
gig_bootstrap_3_Gl66L3LpFTiwZ8jWQ9x_4MLyUUHPRmPtRni0hzJ9RH5WA2Ro6tUv47yNXtK
n3HQ8=_gigya_ver3;
_abck=F0834C860224375A3380551EC0CD74B0~0~YAAQrDxRaFGe6L1xAQAAltSQ0QNh51oGkT

```

```

39nJG7a3fQKoosejMJglSleVwH9M2Kw5cQw/T8wzb3a9OL5VXKiWaQPJtXv/g26JT5KtNh9RGqs
+D2rvCJYGqhFRgpa9X88KBCE7vxqQzNmz7e2KXh5Bvx+lK04ThbBzj5Axa2G2/V2PgxD9/AqAz7
n7se/zrId3oR64G5tAQltyEviWJ27VVvPjDtj2ERjvLYa14C0m0ifMv7gEJdZL9GDARGGDTjojy
wdmk6guybiUIPcEiHJzFjFjjl+ZcyxtTcMDeZ4iCesK/P5EicBznMpmFRRfyhGh1jWws=~-1~-
1~-1; AsosExecutionEngine=ExemptionTimeout=05/01/2020 19:33;
AsosCustomerAlert=; deliveryCountry=ua;
forterToken=c709cb0f58bf4a3aa7369607f9c6646d_1588360918138_208_UAL9_9ck;
glt_3_Gl66L3LpFTiwZ8jWQ9x_4MLyUUHPRmPtRni0hzJ9RH5WA2Ro6tUv47yNXtKn3HQ8=st2.
s.AcbH2dZzZQ.07IaKiMlGZQ9ryAAEN6t13e10AI2QqY0-
cxjZd9dOmWnoI_LZ7Ysb4PffjV56cjCJq7nB0IeQsWLotlNjmFERl6NUurtwhZZysJRsbS242_g.
L8qS2XSoixE7oGhCKxMIFmolVdE4g1FeIHUPHtGf_aIovBEyIX-
WZtZQjligmAtzdJjrXY3fv3o7mqhQ4WlzbG.sc3%7CUUID%3Df41dcaa6eb9f4550bf98388d13
8d1749; _gig_llp=facebook; _gig_llu=Miroslava; asos-
perx=520893aae0364dba91e47c0ef8fffb20|40a70422917241529a55c73bb5560150|298a
118299f6409085fa707e7e73aa25;
      'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36'
    },
    transform: function (body) {
      return cheerio.load(body);
    }
  };

  let $;
  try {
    $ = await request(options);
  } catch (err) {
    console.error(err)
  }

  fs.writeFileSync(path.join(__dirname, 'downloads', 'pages',
`page_${page}.html`), $.html(), {encoding: 'utf8'});

  let parsedPages;
  if (page === 1) {
    const downloadsQueue = [];

    downloadsQueue.push(downloadSearchPage(2));

    parsedPages = await Promise.all(downloadsQueue);
  }

  const singlePages = [];
  const singlePagesLinks = $('div._3pQmLlY article a');

  for (const prop in singlePagesLinks) {
    if (!isNaN(parseInt(prop))) {
      singlePages.push(singlePagesLinks[prop].attribs.href);
    }
  }
  console.log(`page # ${page}, links ${singlePages.length}`)

  if (parsedPages) {
    return singlePages.concat.apply([], parsedPages);
  }

  return singlePages;
}

```

```

module.exports = downloadSearchPage;

export class Kmeans {
  constructor(points, clustersCount) {
    this.data = clone(points);
    this.length = this.data.length;
    this.clusters = this.initClusters();
  }

  step() {
    for (let i = 0; i < length; ++i) {
      this.addToNearestCluster(data[i]);
    }
    const changedClusters = recomputeCentroids();
    const hasMoreWork = changedClusters > 0;

    return hasMoreWork;
  }

  recomputeCentroids() {
    let changedClusters = 0;
    for (let i = 0; i < clustersCount; ++i) {
      changedClusters += this.clusters[i].recomputeCentroid();
    }
    return changedClusters;
  }

  addToNearestCluster(pt) {
    let min = Number.POSITIVE_INFINITY;
    let idx = -1;

    for (let i = 0; i < clustersCount; ++i) {
      const dist = this.clusters[i].distanceTo(pt);
      if (dist < min) {
        min = dist;
        idx = i;
      }
    }

    if (idx !== pt.cluster) {
      if (pt.cluster !== undefined) {
        this.clusters[pt.cluster].remove(pt);
      }
      this.clusters[idx].add(pt);
      pt.cluster = idx;
    }
  }

  getClusters() {
    return this.clusters;
  }

  initClusters() {
    let clusters = [];
    let maxLength = length - 1;
    let selectedCentroids = Object.create(null);
  }
}

```

```

    while (clustersCount !== clusters.length) {
      const idx = (Math.random() * maxLength) | 0;
      const point = data[idx];

      this.swap(idx, maxLength);

      if (maxLength === 0) {
        throw new Error('Could not find ' + clustersCount + '
unique data points');
      }

      maxLength -= 1;
      const key = this.getKey(point);

      if (selectedCentroids[key]) {
        continue;
      }

      selectedCentroids[key] = 1;
      clusters.push(new Cluster(point));
    }

    return clusters;
  }

  getKey(p) {
    return '' + p.x + p.y;
  }

  swap(a, b) {
    const temp = data[a];
    data[a] = data[b];
    data[b] = temp;
  }
}

```

ДОДАТОК Б

Слайди презентації

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

АТЕСТАЦІЙНА РОБОТА МАГІСТРА

Методи кластеризації даних
Е-магазинів з підтримки економічно вигідних рішень для клієнтів за
модю та найкращою ціною

Виконала:
ст. гр. ІПЗм-18-4
Яременко М.М.

Науковий керівник:
доц. Ревенчук І.А.

МЕТА РОБОТИ

- Дослідити методи кластеризації даних для формування економічно вигідних рішень для Е-магазинів за модю та найкращою ціною.
- Програмна реалізація клієнт-серверного застосунку агрегатору, що дозволить користувачу бачити товари із різних ресурсів та знаходити схожі товари за зображенням з урахуванням моди та найкращої ціни.

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

- Негативний вплив на пошук за критеріями має занадто велика складність фільтрів, що використовується. Також суттєвим недоліком є те, що пошук товару за зображенням має невелика кількість платформ з продажу товарів. Основним засобом впливу на властивості об'єкта в комп'ютерних системах є курсор мишки та пункти меню, що дозволяють змінювати ці властивості.
- Існуючі програмні рішення не мають зручного рішення, яке би дозволило знаходити товар із різних джерел за зображенням та найкращою пропозицією (ціною та модою).

3

ПОСТАНОВКА ЗАДАЧІ

- Дослідити методи кластеризації даних для роботи із зображеннями для пошуку потрібних аналогів
- Реалізувати програмну систему, що задовольняє наступним вимогам:
 - Має можливість збору даних із різних джерел
 - Має можливість пошуку аналогів за зображенням за допомогою кластеризації на аналізу зображення

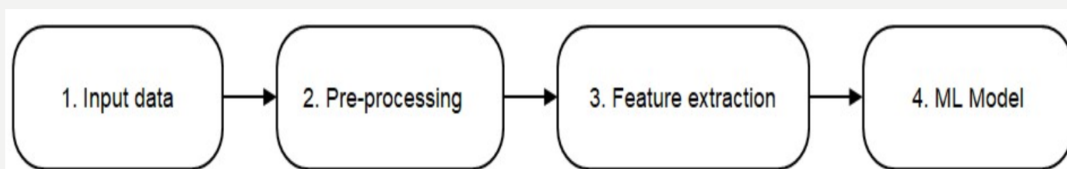
4

ОГЛЯД АНАЛОГІВ

- Hotline.ua
 - має функціонал пошуку за критеріями
 - не має функціоналу пошуку за зображенням
- Price.ua
 - має функціонал пошуку за критеріями
 - не має функціоналу пошуку за зображенням
 - має тільки дані партнерів
- Aliexpress.com»
 - має функціонал пошуку за критеріями
 - має функціонал пошуку за зображенням тільки для мобільного додатку
 - має дані лише від зареєстрованих у системі продавців

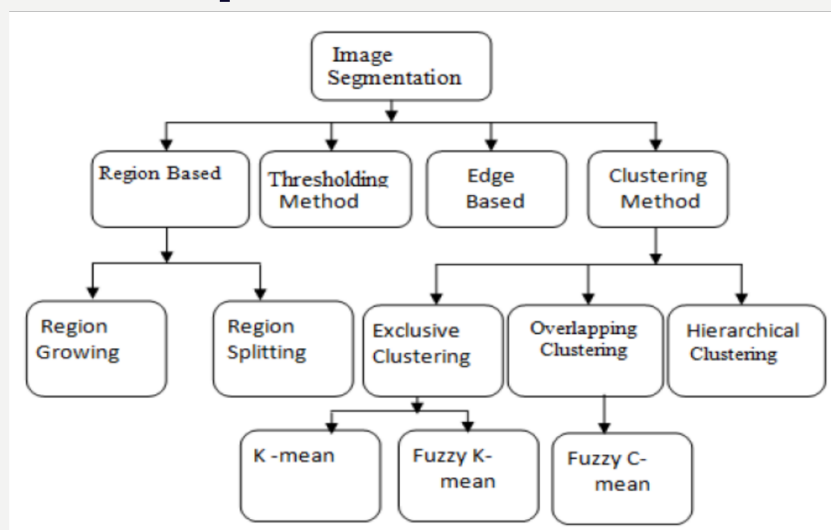
5

СХЕМА КРОКІВ РОЗПІЗНАВАННЯ ОБ'ЄКТУ НА ЗОБРАЖЕННІ



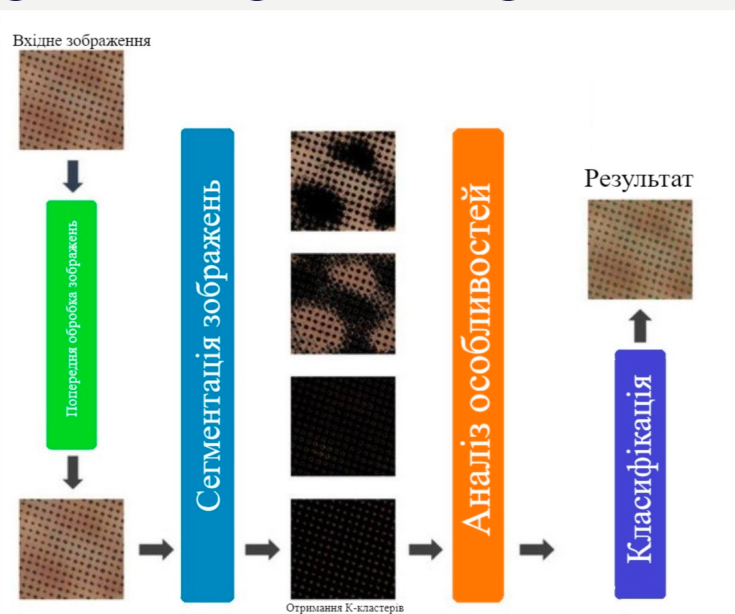
6

КЛАСИФІКАЦІЯ АЛГОРИТМІВ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ



7

ГОЛОВНІ КРОКИ АЛГОРИТМУ



8

ОТРИМАННЯ КЛАСТЕРІВ

1. поділити зображення на 3 канали Red(червоний), Green(зелений), Blue(блакитний);
2. проініціалізувати кількість кластерів k та центр для кожного каналу;
3. для кожного пікселю зображення потрібно вирахувати Евклідову відстань d між центром та кожним пікселем, використовуючи формулу наведену нижче:

$$d_k = \sum_{i=1}^3 \|p^i(x, y) - C_k^i\|$$

4. Віднести пікселі до найближчого з центроїдів;
5. Після того, як всі пікселі будуть віднесені до центрів, вирахувати нові центроїди шляхом усереднення пікселів. Формула усереднення наведена нижче:

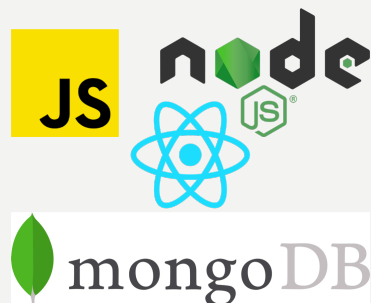
$$C_k^i = \frac{1}{k} \sum_{y \in C_k^i} \sum_{x \in C_k^i} p^i(x, y)$$

6. Повторити процес.

9

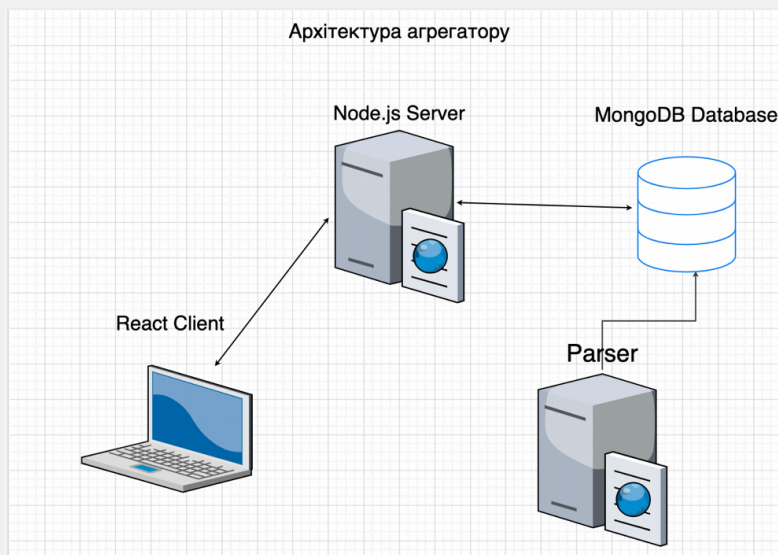
СТЕК ТЕХНОЛОГІЙ

- JavaScript
- Node.js
- React.js
- MongoDB



10

АРХІТЕКТУРА ПО



11

ФРАГМЕНТИ КОДУ ФУНКЦІЇ ЗНАХОДЖЕННЯ КЛАСТЕРІВ ДЛЯ ЗОБРАЖЕННЯ

```

export const findClusters = (img) => {
  const base64Img = toBase64(img.src);

  if (!base64Img) return null;
  const size = [img.height, img.width];

  const gaussianBlurFilterElement = GaussianBlur(img.src, size, 0, 0, BORDER_DEFAULT);
  const laplacianFilterElement = Laplacian(gaussianBlurFilterElement, size, 0, 1, 0, BORDER_DEFAULT);

  const points = generateInitialPoints(laplacianFilterElement);

  const clustersData = new Kmeans(points, { clustersCount: 3, laplacianFilterElement }).getClusters();

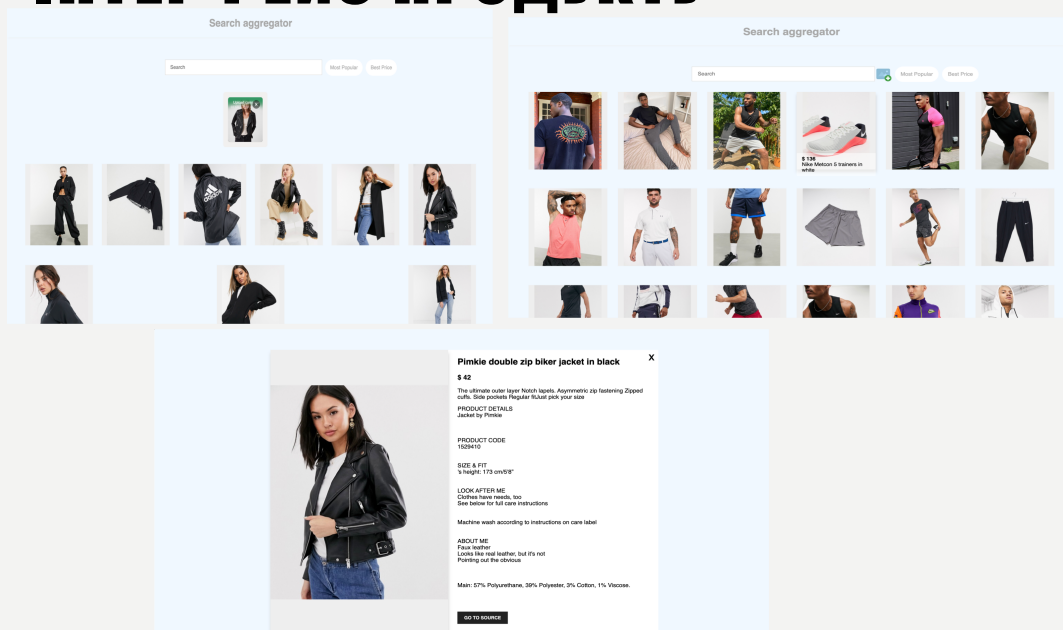
  if (!clustersData.length) return null;

  return clustersData;
}

```

12

ІНТЕРФЕЙС ПРОДУКТУ



13

ВИСНОВКИ

- Було проведено дослідження методів кластеризації даних для роботи із зображеннями для пошуку потрібних аналогів
- Для отримання даних із різних ресурсів було створено парсер, який зчитує дані із DOM та аналізує отриманий текст, витягує потрібну інформацію та перетворює її у документ(об'єкт) для зберігання у базі даних MongoDB
- Для реалізації функції пошуку за зображенням, було використано алгоритм сегментації зображень, який базується на алгоритмі кластеризації даних k-середніх.
- При реалізації сегментації із використанням алгоритму кластеризації k-середніх, був реалізований підхід з розподіленням на кластери, спираючись на кольорові особливості зображення та розподілення на 3 канали RGB спектру, тобто R – червоний, G – зелений, B – блакитний.
- Розроблене програмне забезпечення задовольняє усім вимогам поставленої задачі

14

ДОДАТОК В

Апробація результатів роботи

SCI-CONF.COM.UA**PRIORITY DIRECTIONS
OF SCIENCE DEVELOPMENT****ABSTRACTS OF V INTERNATIONAL
SCIENTIFIC AND PRACTICAL CONFERENCE
MARCH 2-3, 2020****LVIV
2020**

- 50 Сайко В. Г., Наритник Т. М., Сивкова Н. М. Телекомунікаційна система впливу електромагнітного випромінювання на супутникові канали радіозв'язку 222
- 51 Сафронов О. М. Основні методологічні положення експериментальних досліджень гальмівної системи вагонів метрополітену з дисковими гальмами 227
- 52 Слюсенко А. М., Пономаренко В. В. Дослідження роботи водоповітряного ежектора з конічно-циліндричною камерою змішування 232
- 53 Соц С. М., Кустов І. О., Кузьменко Ю. Я. Нова сировина як основа розвитку вітчизняних кру'пяних технологій 236
- 54 Стрілець В. В., Шевченко Р. І. Функціональна схема процесу термічної деструкції хімічно-небезпечних речовин математичної моделі локалізації осередку надзвичайної ситуації 241
- 55 Тищенко В. І., Єршов О. В., Шубіна Є. А., Божко Н. В. Аналіз функціонально-технологічних властивостей м'ясо-містких фаршів з регіональної сировини 244
- 56 Хозя П. А., Сафронов А. М., Водяников Ю. Я. Оценка фрикционных свойств композиционных тормозных колодок на соответствие нормативным требованиям по обеспечению тормозной эффективности грузовых вагонов 249
- 57 Цирульник С. М. ІХ всеукраїнська олімпіада з радіоелектроніки 254
- 58 Шевченко С. М. Пружинний маятник як механічна модель дослідження декількох нелінійно зв'язаних систем 259
- 59 Шпакова Г. В., Глущенко І. В. Адаптація будівельної галузі до принципів циркулярної економіки 265
- 60 Яременко М. М., Ревенчук І. А. Використання кластеризації даних для пошуку по критеріям у Е-магазинах 269

ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

- 61 Марусяк А. І., Завада А. В. Лазерне випромінювання та його вплив на людину 273
- 62 Настасенко В. А. 1-е, 2-е и 3-е квантово-механические уравнения В.А. Настасенко и их определение 276

ГЕОГРАФИЧЕСКИЕ НАУКИ

- 63 Бойко З. В., Бойко О. В. Географічні особливості глобалізації і регіоналізації світу 281
- 64 Васенко А. Г., Старко Н. В. Оценка влияния биологической мелиорации на развитие фитопланктона в водоеме-охладителе Змиевской ТЭС 286

АРХИТЕКТУРА

- 65 Медведєва Ю. Л., Тетянін І. О. Архітектура Української Церкви в стилі бароко 289

ВИКОРИСТАННЯ КЛАСТЕРИЗАЦІЇ ДАНИХ ДЛЯ ПОШУКУ ПО КРИТЕРІЯМ У Е-МАГАЗИНАХ

Яременко Мирослава Максимівна

студент

Харківський національний університет радіоелектроніки
м. Харків, Україна

Ревенчук Ілона Анатоліївна

доцент, кандидат технічних наук

Харківський національний університет радіоелектроніки
м. Харків, Україна

Вступ. У наш час усе більшої необхідності набирає обробка даних, а у особливості групування та упорядкування даних.

В епоху розвитку Internet усе більшої популярності набувають Internet-платформи з можливістю здійснити покупки онлайн. Для того, щоб мати змогу при пошуку, потрібних для покупця об'єктів, оперувати великою кількістю даних швидше - з'являється необхідність групування існуючих об'єктів по критеріям. Для роботи із великою кількістю даних задля вирішення проблеми пошуку, або знаходження аналогів доцільно використовувати методи кластеризації даних. Групування об'єктів спрощує пошук та дає можливість аналізувати об'єкти зі схожими критеріями.

Мета роботи. Метою даного дослідження є знаходження відповідного методу кластеризації та класифікації для того щоб мати змогу розділити вибірку об'єктів на такі групи (кластери), які будуть максимально схожі, об'єднуються у межах даної підмножини, але вони будуть відрізнятись настільки ж, як і об'єкти з інших підмножин.

Матеріали та методи. Клієнти хочуть отримати найбільш релевантні результати, або отримати аналогічні схожі результати. Таким чином, задача полягає в тому, щоб дати їм багато варіантів і потім відсортувати те, що для них найбільш актуально. Наприклад, скажімо, у нас є купа фотографій або продуктів, які містять різні речі, такі як взуття, сорочки, сукні, джинси, куртки і

т. д. Ми можемо навчити модель навчання під наглядом на цих фотографіях, щоб вивчити елементи на кожній фотографії, а потім використовувати цю модель, щоб розпізнати ті ж предмети на нових фотографіях. Ці елементи по суті є цільовими змінними, які ми хочемо, щоб модель вивчила. Для того, щоб мати контрольовану навчання, ви повинні мати чітко визначені дані.

Люди називають речі різними речами, а в роздрібній торгівлі найгірше, що ви можете зробити - це пошукова система не доставляє нічого клієнту, тому що вони набрали альтернативне слово чи синонім. Ці типи виходів називаються дискретними змінними виходу. Навчання машини пошуку всіх предметів у конкретному класі вимагає, щоб ваші дані про навчання були чітко визначені. Можна виділити кілька різних алгоритмів машинного навчання для тренування даних.

Алгоритм k-Найближчих сусідів (KNN) дуже простий і дуже ефективний. Прогнози робляться для нової точки даних шляхом пошуку всього навчального набору для K-подібних екземплярів (сусідів) та узагальнення вихідної змінної для цих екземплярів K. Найбільший випадок використання k-Найближчих сусідів - це системи рекомендацій, в яких, якщо ми знаємо, що користувачеві подобається певний предмет, ми можемо рекомендувати подібні елементи для них.

Дерева рішень - ще один важливий тип методики класифікації, що використовується для машинного навчання прогнозування. Представлення моделі дерева рішень є бінарним деревом. Кожен вузол являє собою одну вхідну змінну (x) та точку розбиття на цій змінній (якщо вважати, що змінна є числовою). Листові вузли дерева містять вихідну змінну (y), яка використовується для прогнозування. Прогнози робляться до тих пір, поки не надійде до вузла і виведе значення класу в цьому вузлі.

Логістична регресія - це метод переходу, коли наша цільова змінна категорична з двома або більше рівнями.

Модель Naive Bayes складається з двох типів ймовірностей, які можна обчислити безпосередньо з ваших навчальних даних: ймовірність кожного

класу та умовна ймовірність для кожного класу з урахуванням кожного значення x . Після розрахунку модель ймовірності може бути використана для прогнозування нових даних за допомогою теореми Байеса. Ключова відмінність від класифікації полягає в тому, що в класифікації ми знаємо, що шукаємо. Це не так у кластеризації. Кластеризацію іноді називають непідтримуваною класифікацією, оскільки вона дає такий же результат, як класифікація, але не маючи попередньо визначених класів.

Один широко використовуваний алгоритм кластеризації - k -means, де k - кількість кластерів для створення. Алгоритм кластеризації k -means починається з k -випадкових центрів кластерів, відомих як центроїди.

Далі алгоритм обчислює відстань від кожної точки до центрів кластерів. Кожна точка присвоюється найближчому центру кластера. Потім центри кластерів перераховуються на основі нових точок кластера. Цей процес повторюється, поки центри кластерів більше не рухаються. Цей простий алгоритм досить ефективний, але чутливий до початкового розміщення кластера.

Для кращого кластеризації може бути використаний другий алгоритм, який називається $bisecting$ k -means. Розбиття k -means починається з усіх точок в одному кластері, а потім розбиває кластери, використовуючи k -means. При наступній ітерації кластер з найбільшою помилкою вибирається розділити. Цей процес повторюється, поки не буде створено k кластерів. Взагалі, ділення k -means створює кращі кластери, ніж оригінальний k -means.

Результати роботи. Отже у результаті дослідження, ми отримуємо робочий сценарій для реалізації конкретного пошуку з використанням кластеризації даних. Цей сценарій має такі пункти:

Кластеризація: набір даних має безліч функцій, які можна класифікувати за профілем товару (типи, марки, опис, колір, розмір). Застосовуючи алгоритм кластеризації k -означає до цього набору даних, ми закінчуємо N різними кластерами. На даний момент ми не знаємо, що являє собою кожен з цих кластерів; тому ми їх довільно називаємо Кластером 1, 2... N .

Класифікація: на даному етапі здійснюється контрольоване навчання, використовуючи методи класифікацій відповідно до кожного кластеру.

Рекомендація: на даному етапі можна рекомендувати товари замовнику, знаючи, що вони дуже актуальні згідно з попереднім аналізом сегментації. Ми можемо просто використовувати алгоритм k-Найближчий сусід, щоб знайти елементи, які можуть бути рекомендовані.

Висновки. Отже у даному дослідженні розглянуто знаходження відповідного методу кластеризації та класифікації для реалізації пошуку у E-магазинах із можливістю запропонувати рекомендовані товари, які будуть актуальні згідно з критеріями пошуку користувача.