**Розпізнавання Шахрайських Транзакцій**

**Команда:**

Цюра Павло, Равінський Дмитро, Чанков Даниїл, Дуда Віктор,

Храмов Олександр, Хомин Юрій, Куклюк Богдан, Коломєйцев Ігор

**Ментор**

Ткаченко Павло

**Звіт**

**∙ Дослідження розпочалося 01.10.2019**

**Зміст**

[**TEРМІНОЛОГІЯ**](#_6vtaegyzvo2) **2**

[**Опис документу**](#_kdmyfhwekzb3) **2**

[**Технічне завдання**](#_3dy6vkm) **2**

[**Загальні вимоги до системи**](#_mt2pkyk7tswi) **2**

[**Впорядкування даних**](#_4d34og8) **4**

[**Розгортання сервера**](#_y4rs5jsu57ea) **4**

[**Скрипт для розрахунку описової статистики**](#_4bhudwwu0727) **5**

[**Скрипт, для аналізу якості вибірки та видалення нерелевантних фрагментів.**](#_vu5awilhvf9g) **6**

[**Процедура перетворення категорійних вхідних змінних**](#_ex2bowna5ytz) **9**

[**Нормалізація входів.**](#_vmayja587i14) **9**

[**Факторний аналіз**](#_b5436nxijoez) **10**

[**Формування рівномірної вибірки**](#_y3cepvjh0w0v) **13**

[**Моделювання**](#_3rdcrjn) **13**

[**Імовірнісна нейронна мережа**](#_cizwxp2ngsqu) **13**

[**Результати тестування для мережі із 87 входами**](#_x7nlpy2hx5zl) **15**

[**Інтерпретація результатів**](#_dlpktodfzlk1) **15**

[**Очищенням даних 2**](#_39fruh4qp3yb) **15**

[**Факторний аналіз 2**](#_7j28x8aw9xy) **15**

[**Інтерпретація результатів**](#_qpr3ogllpzam) **16**

[**Імплементація моделі**](#_2cn5asb6gdiy) **17**

[**Додатки**](#_t2dcllpfisyn) **18**

# 

# 

# TEРМІНОЛОГІЯ

|  |  |
| --- | --- |
| **Tермін** | **Пояснення** |
| AI, ШІ | Artificial intelligence, Штучний інтелект |
| МL,МН | Machine learning, Машинне навчання |
| NN,ШНМ, НМ | Neural network, Штучна нейронна мережа, Нейронна мережа |
| GTM | Geometrical Transformation Machine |
| DB,БД | Database, База данних |
| PNN | probabilistic neural network - ймовірнісна нейронна мережа |
| RBF NN | Radial Basis Function neural network |

## Опис документу

Цей документ створений для запису коротких аналітичних звітів про виконану роботу щодо вирішення проблеми побудови інтелектуальної системи розпізнавання шахрайських дій. Документ поділений на такі секції: Впорядкування даних, Моделювання, Імплементація моделі, Інтерпретація результатів

## Технічне завдання

**Об’єкт дослідження:** Транзакції в системах електронних платежів

**Предмет дослідження:** нейромережеві методи в задачах класифікації транзакцій в системі електронних платежів.

**Мета:** побудова класифікатора для виявлення випадків здійснення шахрайських операцій в системах електронних платежів.

### Загальні вимоги до системи

**Вимоги до вхідних даних :**

Структуровані дані в форматі csv(Comma Separated Values).

**Системні вимоги до часу обробки:**

Час відповіді – 10ms (для одного вектора)

**Методи оцінки результатів:**

Генеральна сукупність даних поділена на тренувальну та тестову вибірки. Тестова вибірка становить 1\3 від генеральної сукупності.

Для кожного із класів - fraud та nonfraud визначається кількість правильно та неправильно класифікованих векторів.

Приклад матриці з результатами тестування моделі:

|  |  |  |
| --- | --- | --- |
|  | fraud | not fraud |
| fraud | 2% | 0.2% |
| not fraud | 98% | 99.8% |

**Вимоги до точності моделі:**

Близька до нуля кількість не шахрайських операцій розпізнаних як шахрайські.

Кількість правильно розпізнаних шахрайських операцій - близько 5%

## 

## Впорядкування даних

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Розгортання сервера Нами були виконані наступні завдання:   * Збір фізичного сервера * Встановлення операційної системи * Налаштування проксі-сервера для віддаленого доступу * Налаштування MS SQL бази даних * Формування файлу з даними для зберігання в базі * імпорт даних на сервер   База містить генеральну сукупність даних, а саме 590541 векторів з 394 ознаками (див. таб.1).  Таблиця 1. Ознаки що описують транзакції   |  |  |  | | --- | --- | --- | | Кількість | Змінні | Зміст | | 1 | TransactionDT | дельта час від певної дати | | 1 | TransactionAMT | Сума в доларах США | | 1 | ProductCD | Код продукту | | 6 | card1 - card6 | Інформація про платежі карток, така як тип картки, категорія карти, банк який видав карту, країна і т.д. | | 2 | addr | Адреса | | 1 | dist | Відстань | | 2 | P\_ and (R\_\_) email domain | Домен електронної пошти отримувача (R\_), та покупця (P\_) | | 14 | C1-C14 | Підрахунок, такий як скільки адрес асоційованих з платіжною карткою, і т.д. Справжнє значення приховане | | 9 | M1-M9 | Збіги, такі як імен, адрес і т.д. | | 357 | Vxxx | Підраховані дані від компанії Vesta, такі як збіги, підрахунки, та інші співвідношення. |  Скрипт для розрахунку описової статистики З метою оцінки даних на придатність для обробки, та аналізу нерелевантних фрагментів даних ми розробили скрипт для розрахунку описової статистики вибірки.  Приклад результатів роботи скрипта:       **Скрипт, для аналізу якості вибірки та видалення нерелевантних фрагментів.**  Цей скрипт почергово аналізує колонки (вхідні ознаки) та видаляє ті з них, що містять значну кількість NULL-ів ( >50%), з умовою, що кількість векторів які представляють шахрайські транзакції не повинна зменшитись більше ніж на 20%.   |  | | --- | | #elimination of nulls and removing vectors with missing values  def select(df, maxNA):  ''' returns DataFrame with deleted null values    df:  data frame    maxNA:  int, value between 1 and 0  max rows lost in percentage  '''    if maxNA > 1 or maxNA < 0:  raise ValueError()    f = open("log.txt", 'a+')  import datetime  f.write('\n\n\n' + str(datetime.datetime.now()))  f.write("\nShape before preprocesing:\n")  f.write(str(df.shape))  f.write("\nColumns before preprocesing:\n")  f.write(str(df.columns.tolist()))  max\_NA\_Count = round((1 - maxNA) \* df.shape[0])    df.dropna(axis = "columns", thresh = max\_NA\_Count, inplace = True)    f.write(f"\nColumns with more than {max\_NA\_Count} NA count have been deleted")    f.write("\nShape after first stage:\n")  f.write(str(df.shape))    f.write("\nColumns after first stage:\n")  f.write(str(df.columns.tolist()))    cleanDf = pd.DataFrame()    max\_NA\_Count = round((1 - maxNA) \* df[df['isFraud'] == 1].shape[0])  f.write(f"\nMax\_NA\_Count for frauds = {max\_NA\_Count}\n")    while cleanDf.shape[1] != 1:  cleanDf = df.dropna()  if cleanDf[cleanDf['isFraud'] == 1].shape[0] >= max\_NA\_Count:  f.write('Result DF:\n')  f.write(f'Shape: {cleanDf.shape}\n')  f.write(f'Columns: {cleanDf.columns.tolist()}\n')  f.write('Fraud count: {}'.format((cleanDf['isFraud'] == 1).sum()))  f.close()  return cleanDf  else:  #deleting column greatest amount of nulls  maxNull = max(df[df['isFraud'] == 1].isnull().sum())  for col in df.columns:  if maxNull == df[col][df['isFraud'] == 1].isnull().sum():  f.write(f"\nDeleting {col} column with {df[col].isnull().sum()} nulls\n")  df.drop(labels = col, axis = 'columns', inplace = True)  print("\nNew drop\n")  break  f.write("\n Unable to preproces data")  f.close()  return pd.DataFrame() |   Після видалення нерелевантних Null-ів залишилося 485503 векторів, що cкладаються із 179 компонент - вхідних ознак, а саме:  Таб.2 Вхідні ознаки, що залишились після фільтрації   |  |  |  |  | | --- | --- | --- | --- | | TransactionID | me.com | live.com.mx | gmail | | TransactionDT | earthlink.net | yahoo.com.mx | centurylink.net | | TransactionAmt | att.net | embarqmail.com | yahoo.de | | card1-card3 | aol.com | twc.com | suddenlink.net | | mastercard | hotmail.co.uk | optonline.net | yahoo.es | | visa | live.com | windstream.net | suddenlink.net | | american\_express | rocketmail.com | sbcglobal.net | yahoo.es | | discover | verizon.net | roadrunner.com | prodigy.net.mx | | card5 | msn.com | hotmail.es | netzero.com | | debit | ymail.com | cfl.rr.com | yahoo.fr | | credit | icloud.com | charter.net | ptd.net | | c1-c14 | yahoo.co.uk | cableone.net | frontiernet.net | | d1 | aim.com | q.com | servicios-ta.com | | v95-v137 | netzero.net | frontier.com | live.fr | | v279-v321 | outlook.com | mail.com | gmail.com | |  | comcast.net | hotmail.fr | anonymous.com | |  | outlook.es | juno.com | yahoo.com | |  | sc.rr.com | web.de | hotmail.com | |  | cox.net | protonmail.com | hotmail.com | |  | mac.com | gmx.de |  |  Процедура перетворення категорійних вхідних змінних Усі вхідні ознаки, що представленні категорійними змінними ми повинні перетворити у набір окремих входів із значеннями 1 та 0, відповідно до належності окремої реалізації до того чи іншого класу. Класи, котрі у вибірці по кількості мали значення менше 5%, об’єднувались в окремий клас - other.(статистика попередньої вибірки)   |  | | --- | | #categorical inputs transformation  def expand(df):  columns = df.columns.values  for column in columns:  text\_digit\_vals = {}    if df[column].dtype != np.int64 and df[column].dtype != np.float64:    unique = df[column].unique()  for cat in unique:  df[cat] = df[column].apply(lambda x: 1 if x == cat else 0)  df.drop(labels = column, axis = 1, inplace = True)  return df  def handle\_others(df, columns, minOccur, resColName):  other\_columns = list()  for col in columns:  if df[col].sum() <= minOccur:  other\_columns.append(col)  df[resColName] = df[other\_columns].sum(axis = 1).tolist()  df.drop(labels = other\_columns, axis = 1)  return df |  Нормалізація входів. Нормалізація здійснювалась по колонках, кожна комірка із значенням у колонці ділилась на максимальне значення у колонці, - таким чином усі значення в межах окремого входу приводились до діапазону 0 - 1. Як наслідок на формування коефіцієнтів під час факторного аналізу впливатиме зміна вхідних сигналів а не величини їх абсолютних значень.   |  | | --- | | #normalisation by column  df = pd.read\_csv(r'filename.csv', sep = ';')  df\_head = df.head()  for x in df.columns:  df[x] = df[x] / df[x].max()  df.to\_csv('fileName.csv') |   Було проведене повторне використання скрипта описової статистики підтвердження збереження властивостей редукованої вибірки у порівнянні із початковою, відповідно до закону великих чисел.  Результати підтвердили, що властивості вибірки при такому відборі змінюються несуттєво, що підтверджує коректність застосованого підходу. Факторний аналіз З метою оцінки інформативності ознак був проведений факторний аналіз. Для цієї процедури була використана нейромережа прямого поширення, навчена на об'єднаній пронормованій вибірці, та ідентифікаційна матриця в режимі застосування навченої мережі. Сила відгуку мережі на кожен вектор ідентифікаційної матриці свідчить про впливовість окремого вхідного параметра.  Входи з навантаженням меншим від 1% були оцінені як неінформативні, та були виключені із подальшого аналізу.  Таб.3. Входи які ввійшли в результуючу вибірку   |  |  |  |  | | --- | --- | --- | --- | | Відносна вага входу у відсотках (max=100%) | Порядковий номер та назва вхідної ознаки | Відносна вага входу у відсотках (max=100%) | Порядковий номер та назва вхідної ознаки | | 100 | 124)belsouth.com | 3,304446 | 172)prodigy.net.mx | | 48,56033 | 125)me.com | 3,301488 | 163)protonmail.com | | 47,39577 | 122)yahoo.com | 3,280544 | 168)suddenlink.net | | 32,51522 | 92)v294 | 3,276241 | 171)yahoo.es | | 20,07197 | 113)v315 | 3,276143 | 143)cox.net | | 19,60517 | 117)v319 | 3,258675 | 173)netzero.com | | 17,08475 | 126)earthlink.net | 3,243561 | 175)ptd.net | | 14,05968 | 121)anonymous.com | 3,243005 | 165)gmail | | 12,91005 | 118)v320 | 3,242982 | 174)yahoo.fr | | 9,697012 | 111)v313 | 3,241977 | 178)servicios-ta.com | | 8,962554 | 112)v314 | 3,102678 | 1)TransactionID | | 8,125036 | 123)hotmail.com | 2,517864 | 50)v111 | | 7,662195 | 109)v311 | 2,513349 | 49)v110 | | 7,612577 | 35)v96 | 2,475788 | 48)v109 | | 7,503861 | 6)C | 2,042263 | 22)c4 | | 7,085161 | 115)v317 | 3,947673 | 142)sc.rr.com | | 7,026606 |  | 3,900342 | 138)netzero.net | | 7,01444 | 116)v318 | 3,889672 | 144)mac.com | | 6,889746 | 133)msn.com | 3,872373 | 129)hotmail.co.uk | | 6,354556 | 127)att.net | 3,850105 | 149)optonline.net | | 6,202123 | 4)W | 3,694017 | 145)live.com.mx | | 6,193457 | 110)v312 | 3,683077 | 146)yahoo.com.mx | | 5,038123 | 137)aim.com | 3,649942 | 136)yahoo.co.uk | | 4,795235 | 5)R | 3,628742 | 46)v107 | | 4,738725 | 134)ymail.com | 3,627836 | 47)v108 | | 4,511546 | 131)rocketmail.com | 3,623072 | 45)v106 | | 4,236015 | 130) live.com | 3,557928 | 150)windstream.net | | 4,198555 | 139)outlook.com | 3,556899 | 140)comcast.net | | 4,156612 | 132)verizon.net | 3,493977 | 152)roadrunner.com | | 4,103017 | 141)outlook.es | 3,459016 | 154)cfl.rr.com | | 4,082863 | 3)TransactionAmt | 3,451831 | 158)frontier.com | | 4,059968 | 135)icloud.com | 3,424292 | 147)embarqmail.com | | 3,342919 | 148)twc.com | 3,389284 | 162)web.de | | 3,334414 | 167)yahoo.de | 3,387649 | 159)mail.com | | 3,323696 | 160)hotmail.fr | 3,374979 | 151)sbcglobal.net | | 3,317997 | 161)juno.com | 3,369088 | 156)cableone.net | | 3,315424 | 157)q.com | 3,363546 | 164)gmx.de | | 3,315049 | 166)centurylink.net | 3,356996 | 153)hotmail.es | | 3,312066 | 128)aol.com | 3,3503 | 155)charter.net | | 3,310461 | 176)frontiernet.net | 3,344169 | 170)suddenlink.net | | 3,306754 | 169)yahoo.es | 1,656357 | 114)v316 |   Велика вага деяких входів що представляють поштовий сервер ініціатора транзакції вимагають окремого дослідження, що покликане прояснити знайдений феномен.  Узагальнені висновки проведеного дослідження по поштових серверах:   * Hotmail є старим і недостатньо зв’язаним з технологіями,щоб отримувати більш сучасного постачальника. * Hotmail має поганий захист від спаму. * Hotmail,MSN і Outlook були взломані та Microsoft попередила своїх користувачів про спам.Як наслідок,ми можемо сказати,що ця пошта може бути небезпечною. * Earthlink може бути підозрілою через свого попереднього власника.   https://www.reddit.com/r/computerscience/comments/96h5n1/whats\_wrong\_with\_hotmail/  Інформація про погану безпеку персональних даних:  https://www.forbes.com/sites/kateoflahertyuk/2019/04/15/microsoft-issues-hotmail-msn-and-outlook-phishing-warning/#60399cd7e543  Інформація про Earthlink:  https://www.nytimes.com/2003/09/03/business/earthlink-founder-sentenced-in-fraud.html Формування рівномірної вибірки Оскільки генеральна сукупність даних є виразно нерівномірною 12331 шахрайських і 468210 нешахрайських в початковій вибірці, для уникнення ефекту череди при навчанні НМ, ми сформували два файли, - тренувальний і тестувальний, з однаковою кількістю векторів обох класів, не шахрайські вектори вибрані випадковим чином, оскільки згідно із законом великих чисел така вибірка повністю відтворюватиме статистичні властивості початкової вибірки. |
|  |
|  |

## Моделювання

Для моделювання була підготована рівномірна вибірка (з приблизно рівною к-стю векторів обох класів), та була розділена дві частини:

* Тренувальна - 23 057 векторів
* Тестова - 11 527 векторів

### Імовірнісна нейронна мережа

Нами була реалізована Імовірнісна нейронна мережа, але через необхідність перебору всього набору даних для кожного вектора, обрахунки були надто затратні по часу.

|  |
| --- |
| #probabilistic neural network  @inline  def euqlideanDist(vec1, vec2):  return math.sqrt(sum((vec1 - vec2))\*\*2)  def pnn(train, use):  sigma = 0.5  # list with 0 value, which length = count of classes  numerator = np.array([\*map(float, np.array(['0'] \* len(set(train[:,1]))))])    for tr in train:  E = math.exp(-1 \* (euqlideanDist(use, tr[:-1])\*\*2) / sigma\*\*2)  numerator[int(tr[-1])] += E  # To exclude case of devison by 0  denumerator += E if E != 0 else 0.00000001    return numerator/denumerator |

Вибрано GTM NN, яка здатна значно швидше обробити такий великий масив інформації, оскільки вона є неітеративною. Також був використаний метод передобробки даних -”гіперциліндр”, що здатен показати хороші результати при побудові складних розділяючих поверхонь в задачах класифікації. Цей метод передбачає розміщення векторів на гіперколі, та їх проекцію на розділяючу гіперплощину утворену під час навчання мережі (проходить через початок координат та сегмент верхнього кола гіперциліндра).

|  |
| --- |
| #Normalisation by row  def RBF(ar, a1 = 1, a2 = 1):  ‘’’  ar - numpyArray:  input data  a1 - float:  optimisation parameter 1  a2 - float:  optimisation parameter 2  ‘’’  ar = ar.T  def zeros(x):  return x if x != 0 else 0.000001    def fermi(y):  return 1 / (1 + e - (a2 \* (y - a1)))    return fermi(ar / np.array(list(map(zeros, np.sqrt(sum(ar\*ar)))))) |

## 

### Результати тестування для мережі із 87 входами

|  |  |  |
| --- | --- | --- |
|  | fraud | not fraud |
| fraud | 1.437066% | 98.562934% |
| not fraud | 0.148662% | 99.851338% |

### Інтерпретація результатів

Вибірка містить суперечливі дані, як наслідок хибні спрацювання системи розпізнавання шахраїв відбуваються досить часто: на 10 виявлених шахраїв маємо одне хибне спрацювання. Подальші кроки - ще одна ітерація з очищенням даних.

### Очищенням даних 2

Під час другого етапу очищення даних були видалені ознаки, що містили значну кількість констанстантних значень та мали малу вагу. Під час експериментів із різними наборами вхідних ознак ми з'ясували, що комбінація 15-ти з них забезпечує 100% точність розпізнавання шахрайських операцій та 0% хибних спрацювань. Результати факторного аналізу цього набору вхідних ознак наведено нижче:

### Факторний аналіз 2

|  |  |  |
| --- | --- | --- |
| Номер | коефіцієнт факторного навантаження | Назва ознаки |
| 1 | 1,00000000000001 | v314 |
| 2 | 1,00000000000006 | v315 |
| 3 | 0,999999999999932 | v316 |
| 4 | 0,999999999999903 | v317 |
| 5 | 7,17551018603046E-14 | v318 |
| 6 | 7,52869988573934E-14 | v319 |
| 7 | 0 | v320 |
| 8 | -1,00000000000002 | anonymous |
| 9 | 6,95856133131612E-14 | yahoo |
| 10 | -1,00000000000013 | hotmail.com |
| 11 | -0,999999999999971 | me.com |
| 12 | -1,69847642894627E-13 | earthlink.net |
| 13 | 3,19581600766172E-14 | att.com |
| 14 | 2,74257409859681E-16 | aol.com |
| 15 | 1,24872985216018E-15 | hotmail.uk |

### Інтерпретація результатів

Нам вдалось знайти лінійну залежність для семи вхідних ознак що із 100% точністю відтворюють правильні значення цільової змінної на тестовій вибірці.

Можливою причиною цього факту є штучний підхід до формування генеральної сукупності організаторами конкурсу. Існує велика імовірність того, що анонсовані організаторами шуми, що містяться у вибірці мають не природне походження, а є завадами, інкорпорованими у вибірку для ускладнення завдання побудови моделі.

Найкращі моделі що були представлені на конкурсі забезпечували точність близько 95%. Наша модель продукує результати кращої якості, отже поставлене у дослідженні завдання можна вважати виконаним.

### Імплементація моделі

Спродукована нами модель може бути представлена у вигляді лінійного полінома, і, відповідно, імплементована в такому вигляді у довільній системі підтримки прийняття рішень.

y = a1\*x1+a2\*x2+a3\*x3+a4\*x4+a5\*x5+a6\*x6+a7\*x7

|  |  |
| --- | --- |
| x1= v314 | a1=1 |
| x2= v315 | a2=1 |
| x3= v316 | a3=1 |
| x4= v317 | a4=1 |
| x5 = anonymous | a5=-1 |
| x6 = hotmail.com | a6=-1 |
| x7 = me.com | a7=-1 |

# **Додатки**

**Технічний стек:**

Python

matplotlib

pandas

numpy

Excel

Ubuntu server

MS SQL server

GTM RBF NN

NN Specto

NN Equo