Додаток А

«Графічні матеріали атестаційної роботи»

ГЮИК.502520.015

(позначення документу)

Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
Керівник атестаційної
роботи, проф. Ситніков Д. Е.

РОЗРОБКА МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ КЛІЄНТІВ БАНКІВСЬКОЇ СФЕРИ

Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.502520.015 – ЛУ

РОЗРОБИЛА:
Ст. гр. СПРм 19-1
Шемчук В. Н.

2020

ЗАТВЕРДЖЕНО
ГЮИК.502520.015 – ЛУ

РОЗРОБКА МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ КЛІЄНТІВ БАНКІВСЬКОЇ СФЕРИ

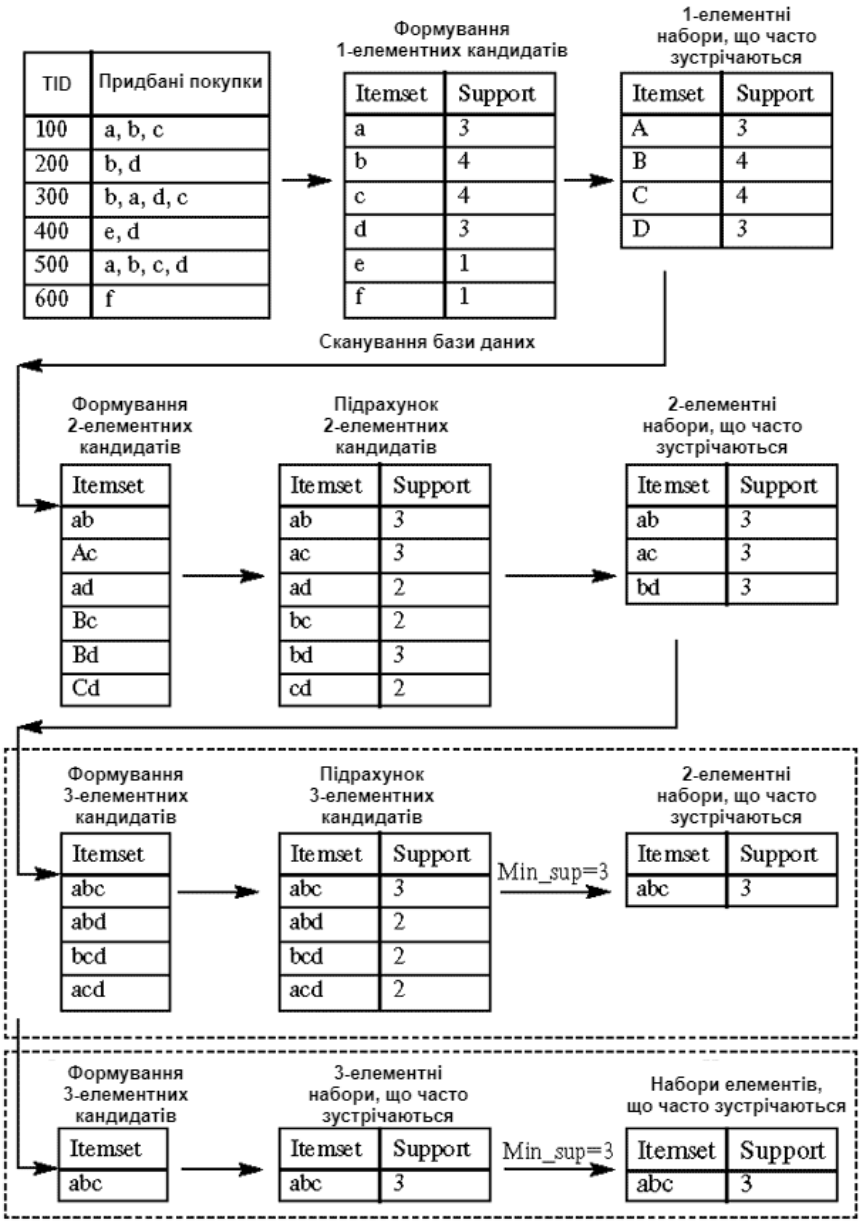
Графічний матеріал

ГЮИК.502520.015 – ЛУ

Аркушів 13

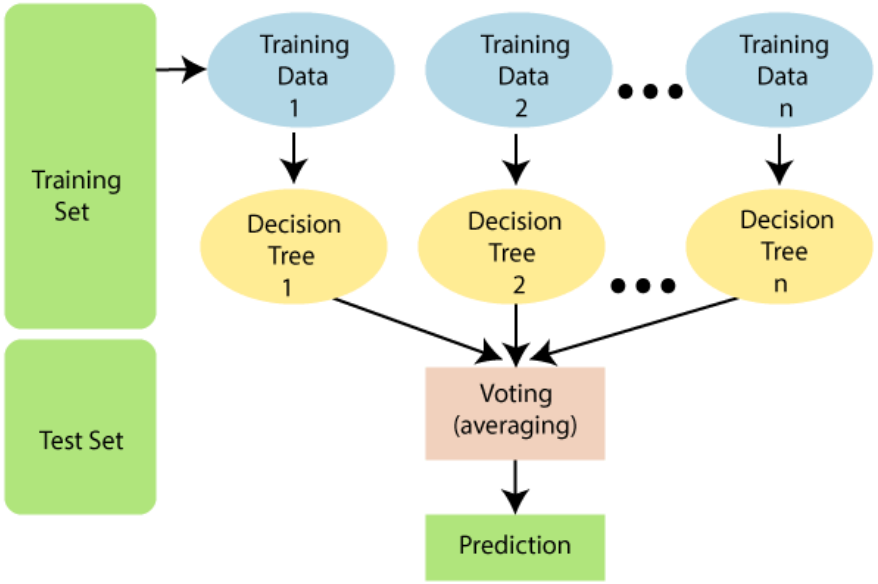
2020

Алгоритм Apriori



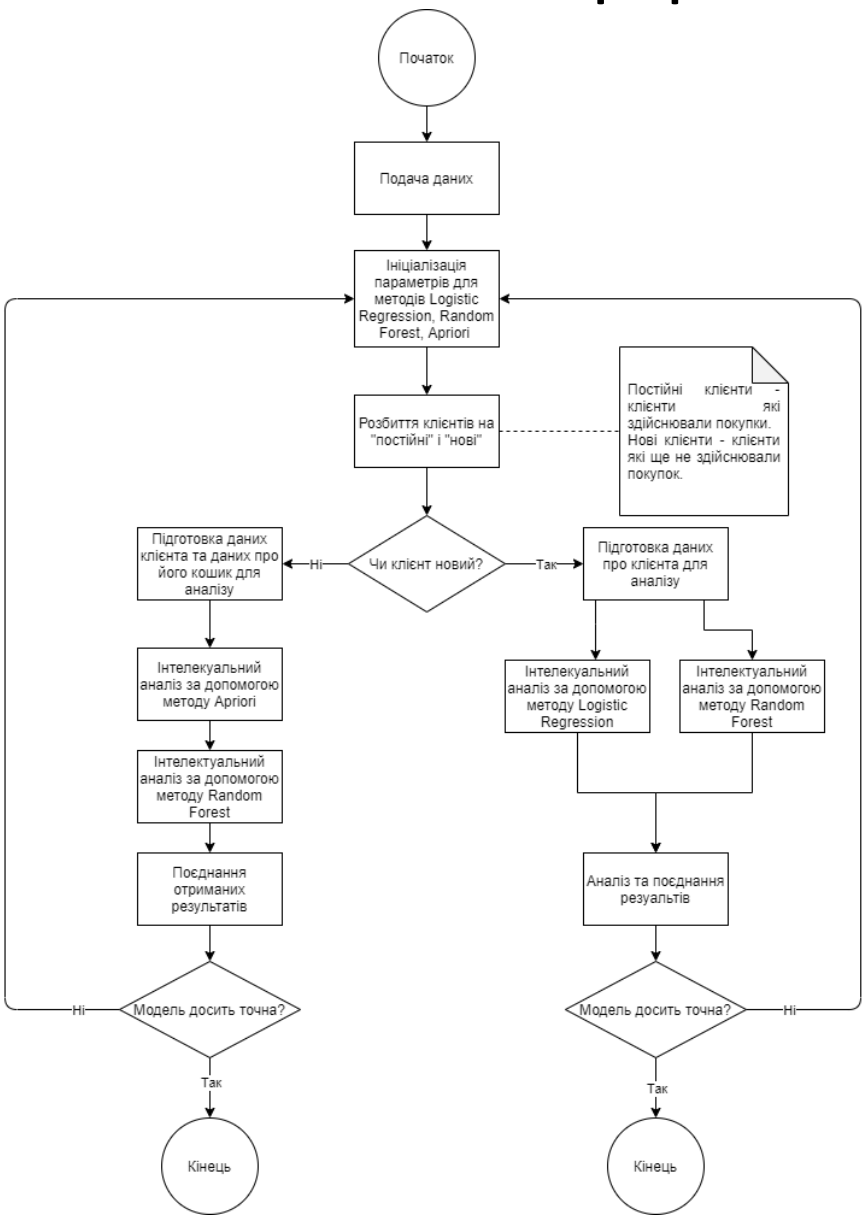
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Навчання алгоритму Random Forest



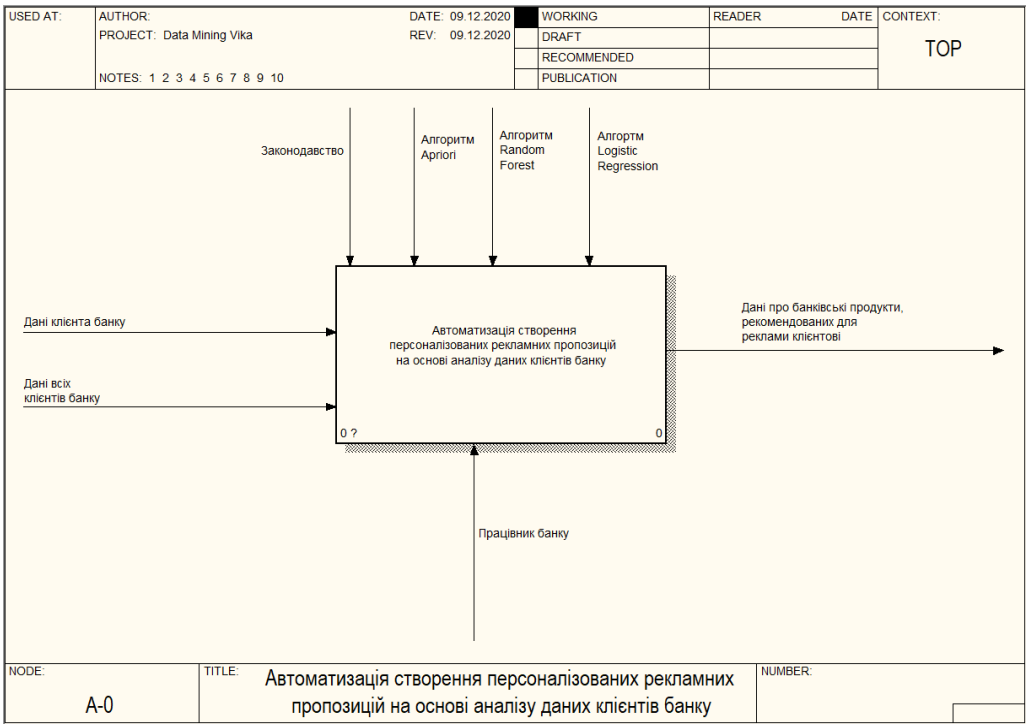
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Алгоритм навчання методу на основі комбінування методів Apriori, Random Forest та логістичної регресії



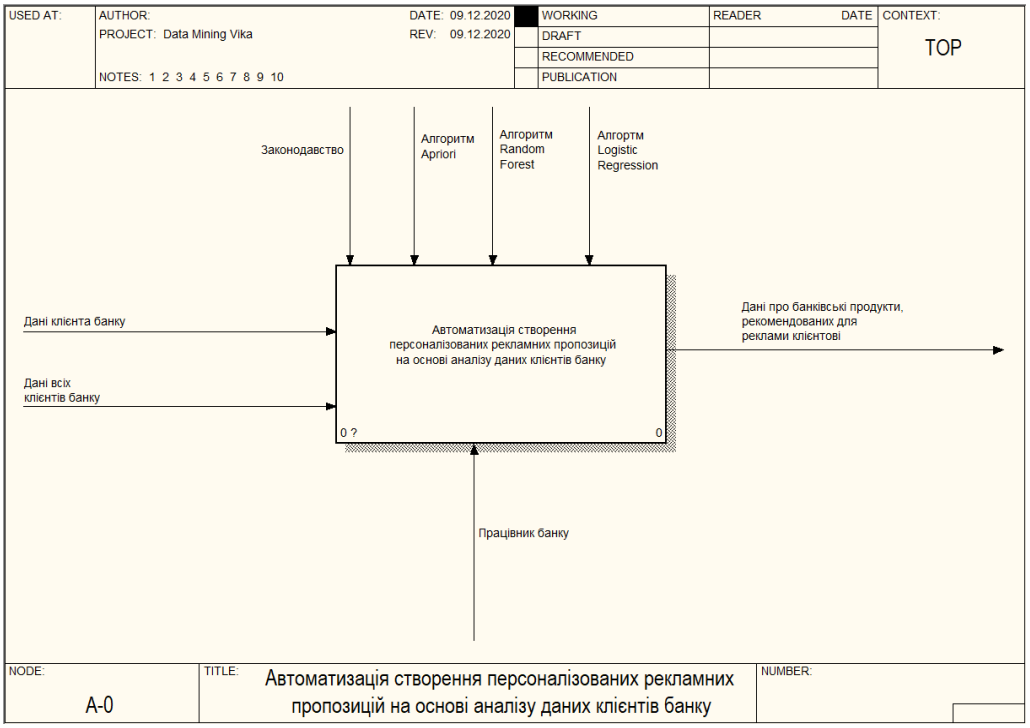
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Концептуальна діаграма функції «Автоматизація створення персоналізованих пропозицій на основі аналізу даних клієнтів банку»



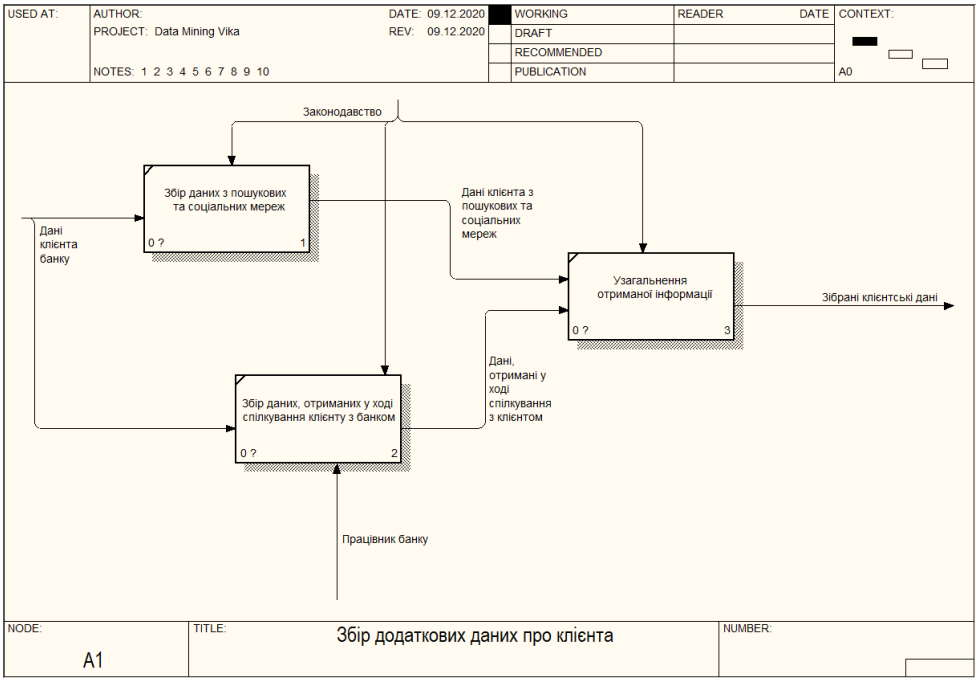
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Діаграма декомпозиції, що представляє основні функції системи



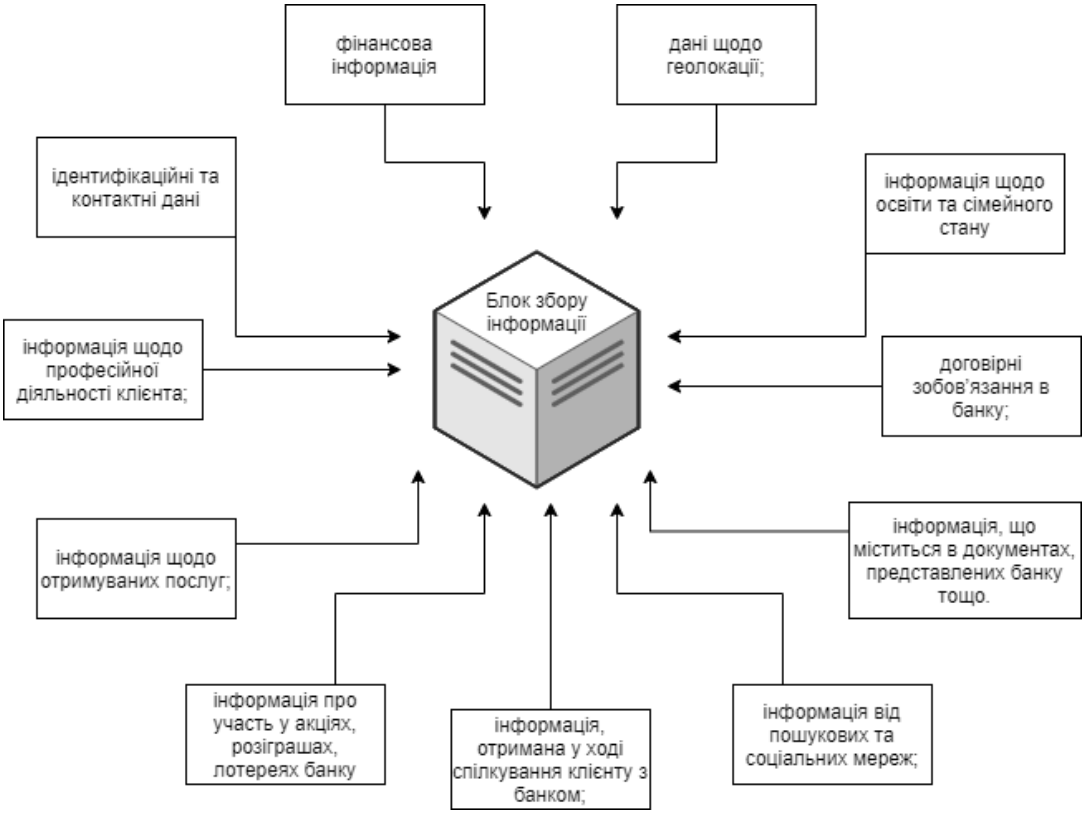
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Діаграма декомпозиції функції «Збір додаткових даних про клієнта»



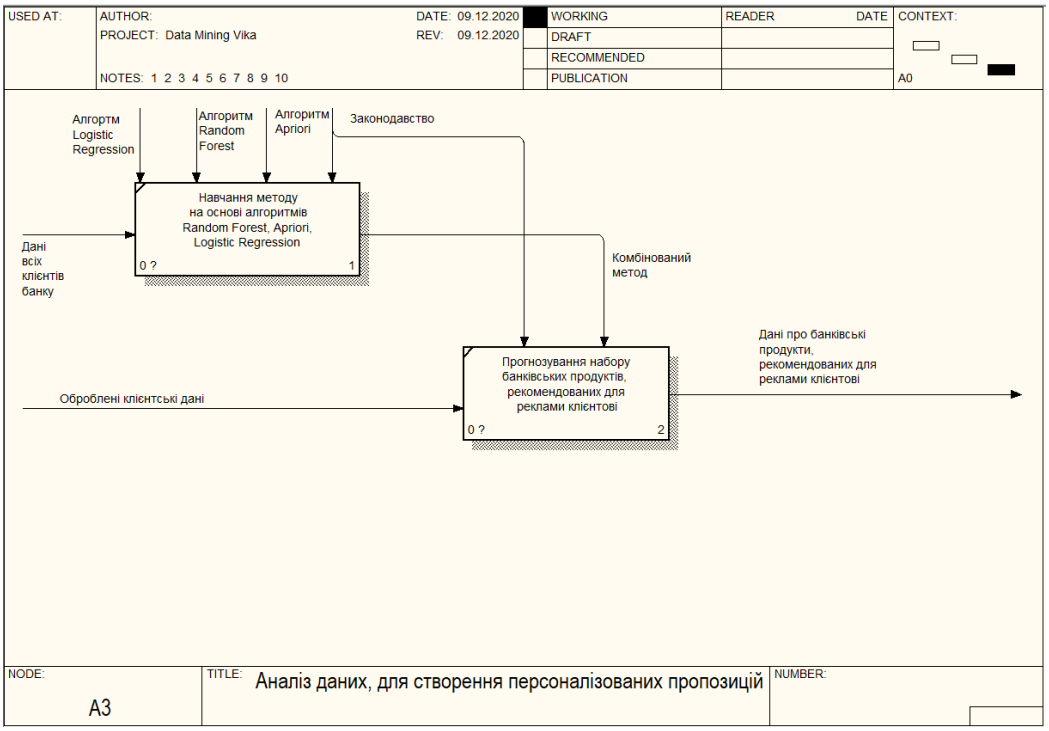
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Блок збору інформації



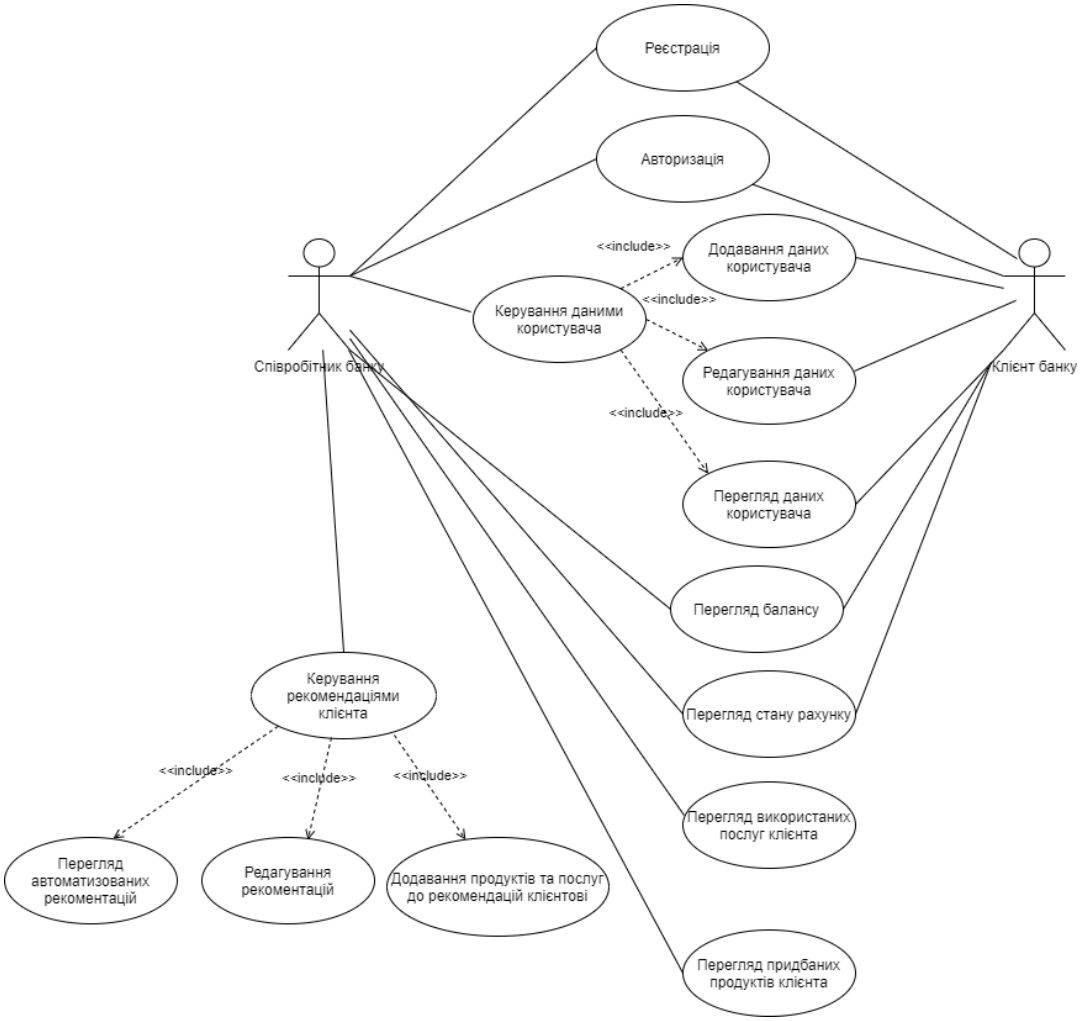
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Декомпозиція функції «Аналіз даних для створення персоналізованих пропозицій»



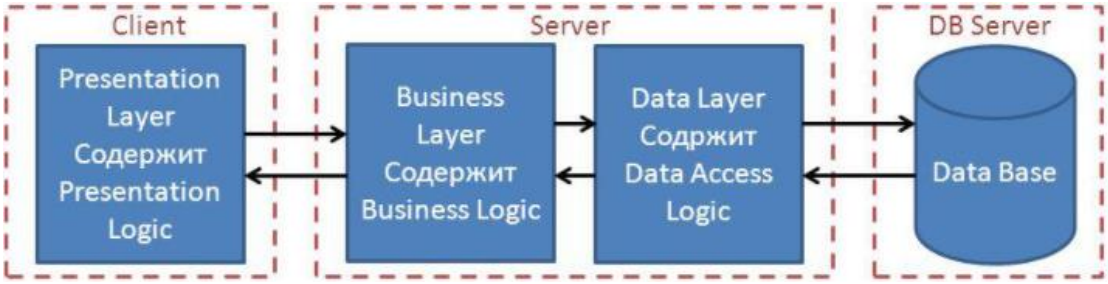
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Діаграма прецедентів системи



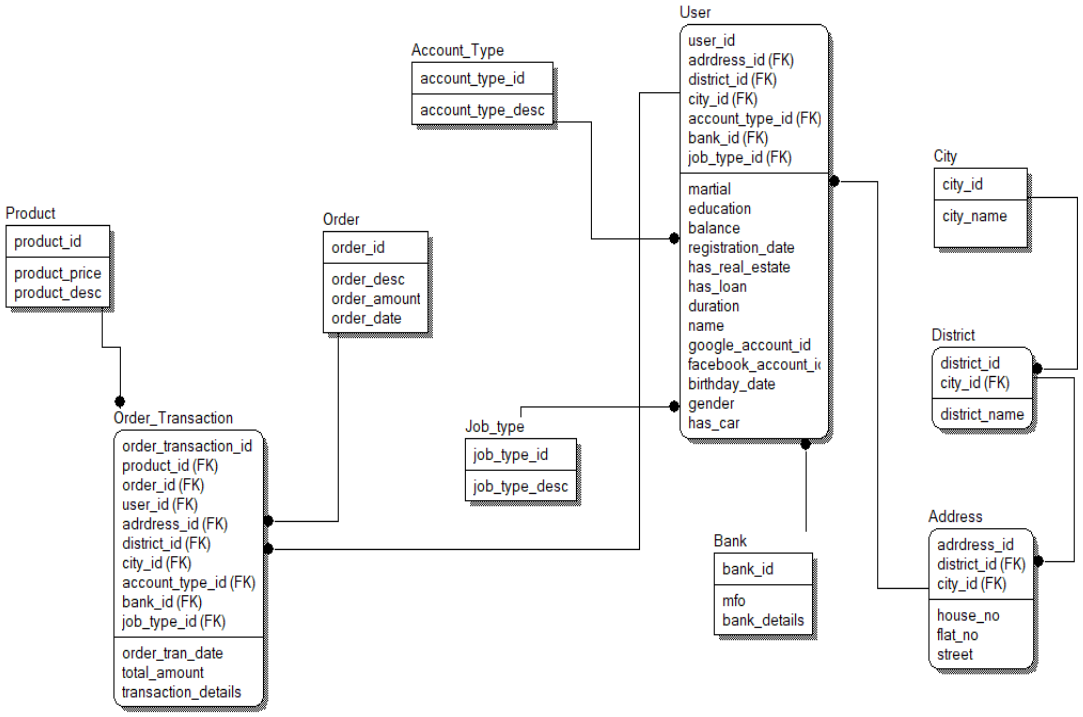
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Схема 3-рівневої архітектури



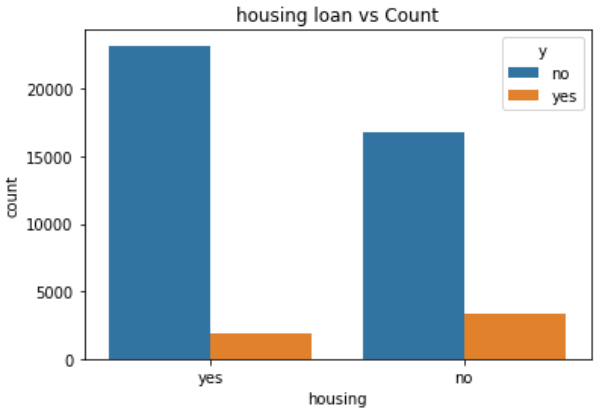
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

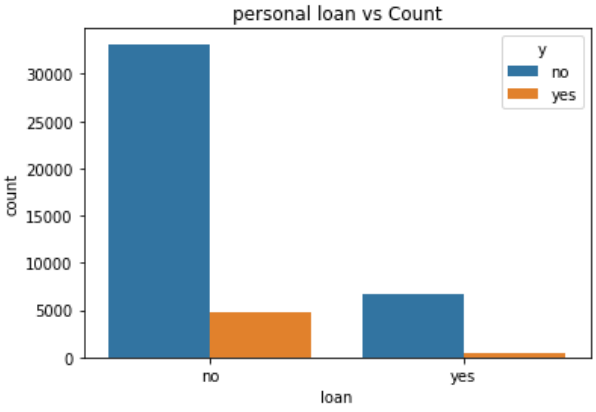
Логічна модель бази даних

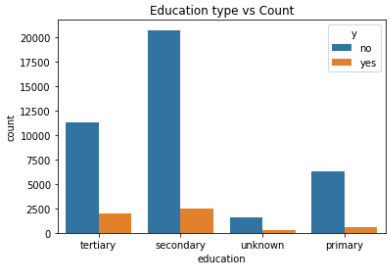


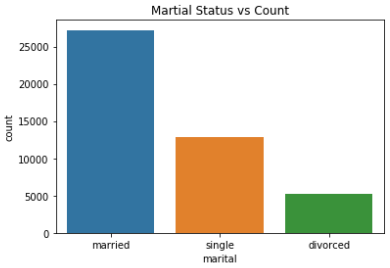
Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

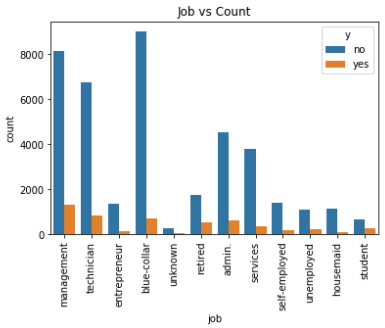
Діаграми залежностей для різних ознак клієнта











Розроб.	*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	
Перевір.	*Ситніков Д. Е.*				
Н. Контр.	*Ситніков Д. Е.*				
				СПРм 19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 1*

Порівняння результатів роботи різних моделей

	precision	recall	f1-score	support
0	0.89	0.98	0.94	10938
1	0.66	0.33	0.44	1495
accuracy			0.90	12433
macro avg	0.79	0.65	0.69	12433
weighted avg	0.88	0.90	0.88	12433

Random Forest

	precision	recall	f1-score	support
0	0.93	0.98	0.96	10938
1	0.77	0.49	0.60	1495
accuracy			0.92	12433
macro avg	0.85	0.74	0.78	12433
weighted avg	0.91	0.92	0.91	12433

Combined model

	precision	recall	f1-score	support
0	0.87	0.98	0.95	10938
1	0.73	0.40	0.51	1495
accuracy			0.91	12433
macro avg	0.83	0.69	0.73	12433
weighted avg	0.90	0.91	0.90	12433

Logistic Regression

Додаток В

«Текст програми»

ГЮИК.502520.015 – 01 12 01 – ЛУ
(позначення документу)

Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
Керівник атестаційної
роботи, проф. Ситніков Д. Е.

РОЗРОБКА МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ КЛІЄНТІВ БАНКІВСЬКОЇ СФЕРИ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.502520.015 – 01 12 01 – ЛУ

РОЗРОБИЛА:
Ст. гр. СПРм 19-1
Шемчук В. Н.

2020

ЗАТВЕРДЖЕНО
ГЮИК.502520.015 – 01 12 01 – ЛУ

РОЗРОБКА МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ КЛІЄНТІВ БАНКІВСЬКОЇ СФЕРИ

Текст програми

ГЮИК.502520.015 – 01 12 01 – ЛУ

Аркушів 9

2020

```
#Importing required libraries

import numpy as np

import pandas as pd

import seaborn as sns.


from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix, accuracy_score,
classification_report

from imblearn.over_sampling import SMOTE

from sklearn.preprocessing import MinMaxScaler


import matplotlib.pyplot as plt

%matplotlib inline


#Reading training and testing data as dataframes

train_df = pd.read_csv("train.csv",sep =";")

test_df = pd.read_csv("test.csv", sep =";")

train_df.head()


#Dimensions of dataset

train_df.shape


#Information about the data types of features

train_df.info()


#Statistical summary of training dataset

train_df.describe(include ='all')
```

```
#Checking if there are any missing values
train_df.isnull().sum()

sns.countplot(x="education",data=train_df, hue = "y")
plt.title("Education type vs Count")

sns.countplot(x="marital", data = train_df)
plt.title("Martial Status vs Count")

sns.countplot(x="job", data = train_df, hue ="y")
plt.title("Job vs Count")
plt.xticks(rotation=90)

sns.countplot(x="loan", data = train_df, hue ="y")
plt.title("personal loan vs Count")

sns.countplot(x="housing", data = train_df, hue ="y")
plt.title("housing loan vs Count")

sns.countplot(x="contact", data = train_df, hue ="y")
plt.title("Contact vs Count")

correlation_matrix = train_df.corr()
sns.heatmap(correlation_matrix, annot =True)

#Combining training and testing data for the purpose of encoding
df = pd.concat([train_df,test_df], ignore_index=True)
df.shape
```

```
df = pd.get_dummies(df,columns =
['job','marital','education','default','housing','month','loan','contact','po
utcome'], drop_first = True)

df.head()

df['y'].replace('yes', 1, inplace=True)

df['y'].replace('no', 0, inplace=True)

df.head()


target = df['y']

df = df.drop('y',axis = 1)

columns = df.columns

scaler = MinMaxScaler()

df = scaler.fit_transform(df)

df = pd.DataFrame(df,columns=[columns])

df.head()


y = np.array(target)

X = df


def train(self, data, Olabels, unique_classes):
          '''
          train the classifier. One classifier per unique label
          '''
          print 'training....'
          debug = self.debug
          regularized = self.regularized
          #print 'train regularized', regularized


          num_iters = self.num_iters
          m,n = data.shape
```

```
            # map labels to program friendly labels
            labels = np.zeros(Olabels.shape)

            uniq_Olabel_names = np.unique(Olabels)

            uniq_label_list = range(len(uniq_Olabel_names))

            for each in zip(uniq_Olabel_names, uniq_label_list):
                  o_label_name = each[0]
                  new_label_name = each[1]
                  labels[np.where(Olabels == o_label_name)] = new_label_name

            labels = labels.reshape((len(labels),1))
            # now labels variable contains labels starting from 0 to
(num_classes -1)
            #print unique_classes
            num_classes = len(unique_classes)

            Init_Thetas = [] # to hold initial values of theta

            Thetas = [] # to hold final values of theta to return

            Cost_Thetas = [] # cost associated with each theta

            Cost_History_Theta = [] # contains list of varying cost thetas

            # if num_classes = 2, then N_Thetas will contain only 1 Theta
            # if num_classes >2, then N_Thetas will contain num_classes number
of Thetas.
```

```
        if(num_classes == 2):
            theta_init = np.zeros((n,1))
            Init_Thetas.append(theta_init)

            # we need only 1 theta to classify class A from class B
            #local_labels = np.zeros(labels.shape)
            #local_labels[np.where(labels == 2)] = 1
            # for i in zip(labels, local_labels):
            #    print i
            # exit()

            local_labels = labels

            assert(len(np.unique(labels)) == 2)

            assert(len(local_labels) == len(labels))

            init_theta = Init_Thetas[0]

            new_theta, final_cost = self.computeGradient(data,
local_labels, init_theta)

            Thetas.append(new_theta)
            Cost_Thetas.append(final_cost)

        elif(num_classes>2):
            for eachInitTheta in range(num_classes):
                theta_init = np.zeros((n,1))
                Init_Thetas.append(theta_init)
                pass
```

```
            for eachClass in range(num_classes):
                # load data local of the init_theta
                # +ve class is 1 and rest are zeros
                # its a one vs all classifier

                local_labels = np.zeros(labels.shape)


                local_labels[np.where(labels == eachClass)] = 1


                # assert to make sure that its true
                assert(len(np.unique(local_labels)) == 2)
                assert(len(local_labels) == len(labels))
                # print eachClass
                # print Init_Thetas
                init_theta = Init_Thetas[eachClass]


                new_theta, final_cost = self.computeGradient(data,
local_labels, init_theta)
                #print final_cost
                Thetas.append(new_theta)
                Cost_Thetas.append(final_cost)

        return Thetas, Cost_Thetas


    def classify(self, data, Thetas):
```

```
            # since it is a one values all classifier, load all classifiers
and pick most likely
            # i.e. which gives max value for sigmoid(X*theta)
            debug = self.debug
            assert(len(Thetas)>0)

            if(len(Thetas) > 1):
                  mvals = []
                  for eachTheta in Thetas:
                        mvals.append(self.sigmoidCalc(np.dot(data, eachTheta)))

                        pass
                  return mvals.index(max(mvals))+1

            elif(len(Thetas) == 1):
                  # either is close to zero or 1
                  # if more than 0.5 classify as 1 and if less than 0.5
classify as 0
                  # print data
                  # print Thetas[0]
                  #print self.sigmoidCalc(np.dot(data, Thetas[0]))

                  cval = round(self.sigmoidCalc(np.dot(data, Thetas[0])))+1.0
                  #print 'classification output: ', cval
                  return cval


      def sigmoidCalc(self, data):
            '''
            calculate the sigmoid of the given data
            '''
```

```
            # if(len(data.flatten()) == 1 ):
            #     data = data.reshape((1,1))
            debug = self.debug
            data = np.array(data, dtype = np.longdouble)
            #g = np.zeros(data.shape, dtype = np.float64)
            g = 1/(1+np.exp(-data))

            return g

#Splitting the data into train and test data
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.25,
random_state = 20)

#Initializing and fitting the logistic regression model
lr_model = LogisticRegression(max_iter=125)
lr_model.fit(X_train,y_train)
y_pred = lr_model.predict(X_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

from sklearn.ensemble import RandomForestClassifier
RFC =  RandomForestClassifier(n_estimators = 500, n_jobs = -1)
RFC.fit(X_train,y_train)
y_pred = RFC.predict(X_test)
print(confusion_matrix(y_test, y_pred))

print(classification_report(y_test, y_pred))

from sklearn.ensemble import VotingClassifier
```

```
VC = VotingClassifier(estimators=[('lr', lr_model), ('rf', RFC)],
voting='hard')

VC.fit(X_train,y_train)

y_pred = VC.predict(X_test)

print(confusion_matrix(y_test, y_pred))


print(classification_report(y_test, y_pred))


newClient = X_test[:1]

VC.predict(newClient)
```

№	Позначення	Найменування	Додаткові Відомості
		Текстові документи	
1.	ГЮИК.502520.015 ПЗ	Пояснювальна записка	76 стор.
2.	ГЮИК.502520.015 – 01 12 01 – ЛУ	Текст програми	9 стор.
		Графічні документи	
3.	ГЮИК.ГЮИК.502520.015 – С10	Алгоритм Apriori	1 аркуш
4.		Навчання алгоритму Random Forest	1 аркуш
5.		Алгоритм навчання методу на основі комбінування методів Apriori, Random Forest та логістичної регресії	1 аркуш
6.		Концептуальна діаграма функції «Автоматизація створення персоналізованих пропозицій на основі аналізу даних клієнтів банку»	1 аркуш
7.		Діаграма декомпозиції, що представляє функції системи	1 аркуш
8.		Діаграма декомпозиції функції «Збір додаткових даних про клієнта»	1 аркуш
9.		Блок збору інформації	1 аркуш
10.		Декомпозиція функції «Аналіз даних для створення персоналізованих пропозицій»	1 аркуш
11.		Діаграма прецедентів системи	1 аркуш
12.		Схема 3-рівневої архітектури	1 аркуш
13.		Логічна модель бази даних	1 аркуш
14.		Діаграми залежностей для різних ознак клієнта	1 аркуш
15.		Порівняння результатів роботи різних моделей	1 аркуш

Зм.	Арк	№ докум	Підп.	Дата	ГЮИК.502520.015 ДЗ		
Розроб.		*Шемчук В. Н.*			*Розробка методу інтелектуального аналізу даних клієнтів банківської сфери*	Аркуш	Аркушів
Перевір.		*Ситніков Д. Е.*				*1*	*1*
Н.контр.		*Ситніков Д. Е.*				*ХНУРЕ*	
Затв.		*Гребеннік І. В.*				*Кафедра СТ*	