

## ДОДАТОК А

Графічний матеріал атестаційної роботи

ГЮИК. 502610.036

(позначення документу)

ЗАТВЕРДЖЕНО  
ГЮИК. 502610.036 – ЛУ

Розробка алгоритму колаборативної фільтрації для роботи з цільовою бізнес-аудиторією та його імплементація в інформаційних системах

Графічний матеріал

ГЮИК. 509000.005 – ЛУ

АРКУШІВ   10  

2020 р.

Міністерство освіти і науки України  
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»  
керівник атестаційної роботи  
проф. Ситніков Д.Е.

Розробка алгоритму колаборативної фільтрації для роботи з цільовою бізнес-аудиторією та його імплементація в інформаційних системах

Графічний матеріал

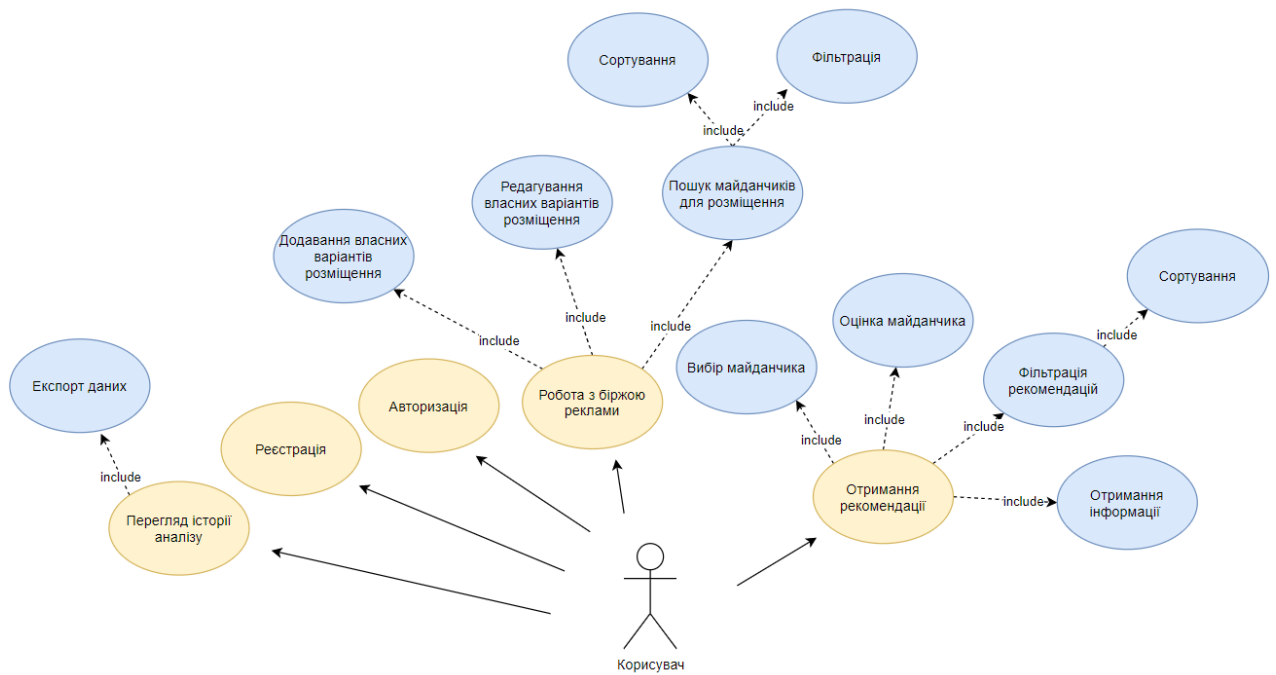
ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 509000.005 – ЛУ

РОЗРОБИВ:  
ст. гр. СПРМ-18-2  
Мамонтов Ю.В.

2018 р.

# Діаграма варіантів використання



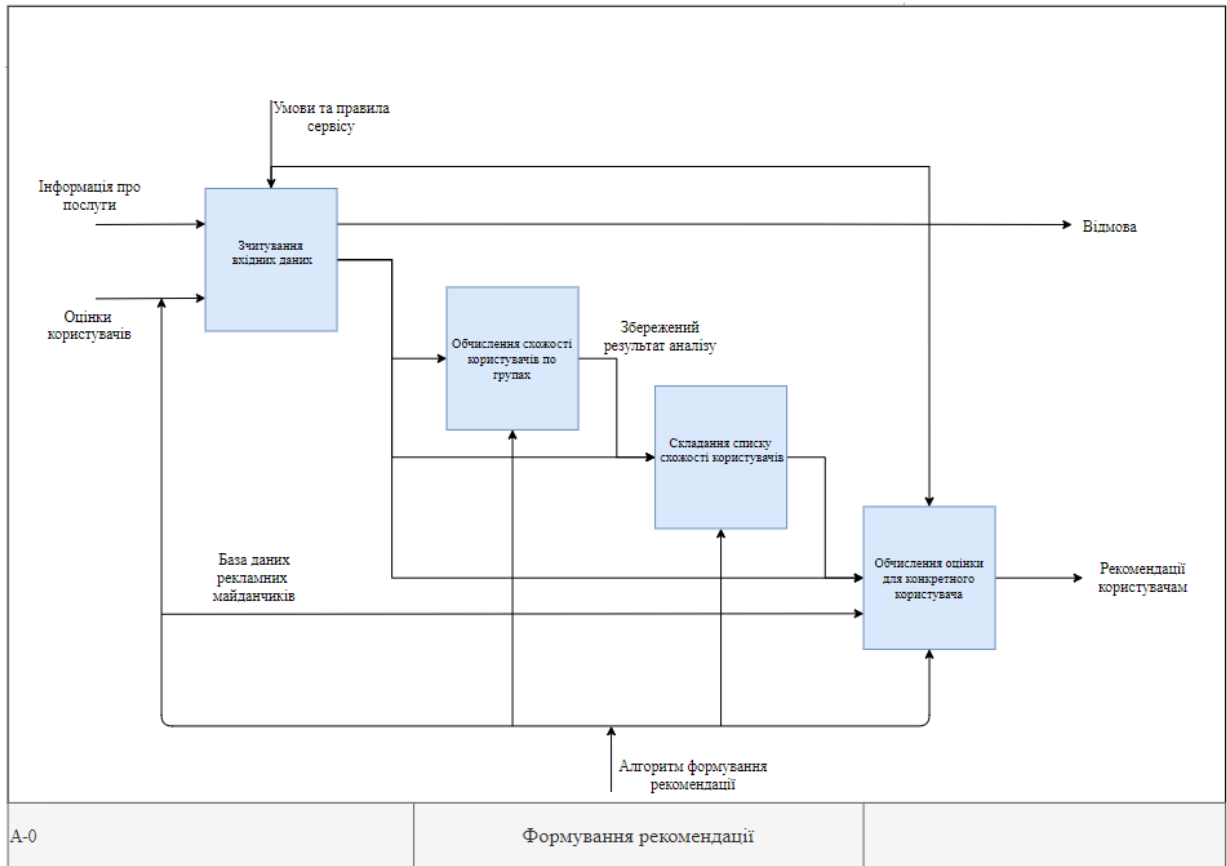
Розроб.	Мамонтов Ю.В.			Діаграма варіантів використання	
Перевір.	Ситніков Д.Е.				
Н. Контр.	Ситніков Д.Е.				
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

# Контекстна діаграма



Розроб.	Мамонтов Ю.В.		Контекстна діаграма	
Перевір.	Ситніков Д.Е.			
Н. Контр.	Ситніков Д.Е.			
			СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.		СТ	Листів 1

# Декомпозиція нульового рівня функції «Формування рекомендацій»

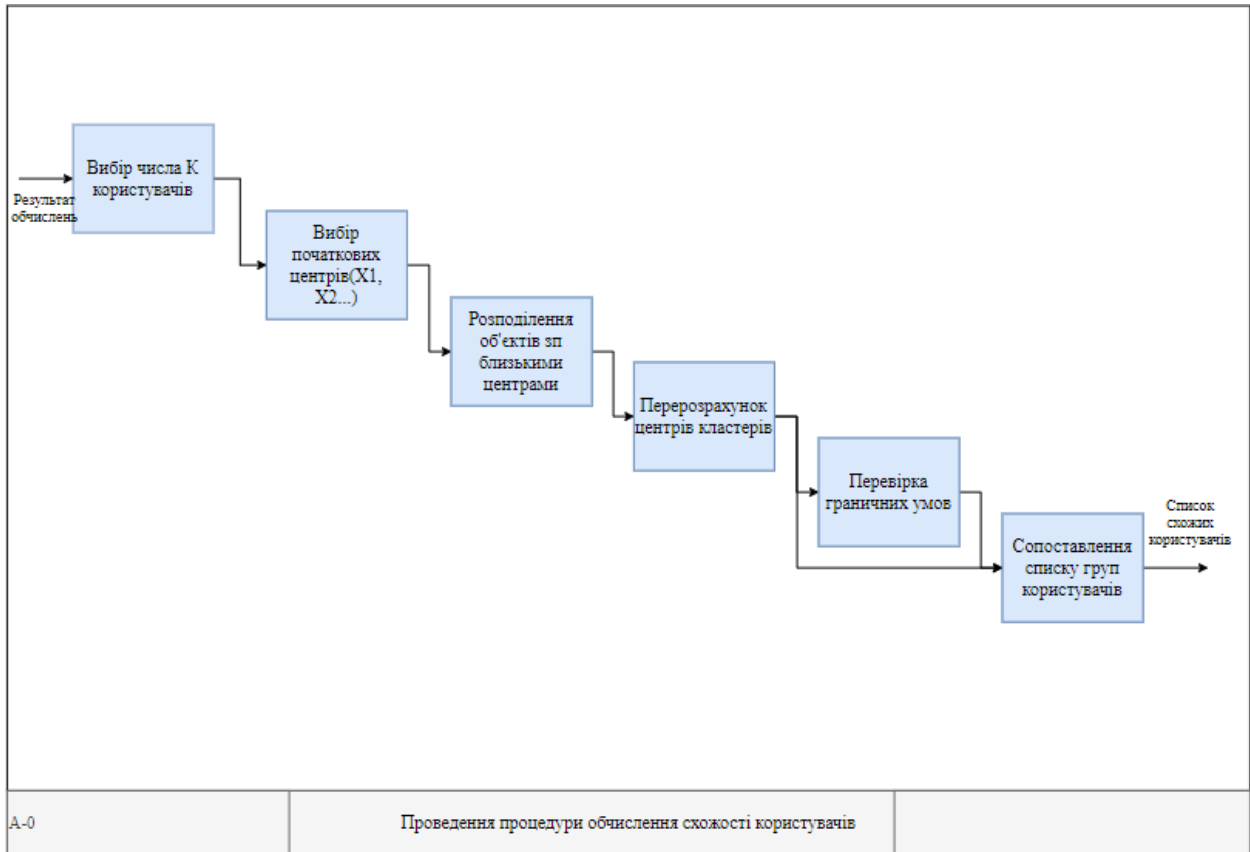


A-0

Формування рекомендації

Розроб.	Мамонтов Ю.В.		Декомпозиція нульового рівня функції «формування рекомендацій»	
Перевір.	Ситніков Д.Е.			
Н. Контр.	Ситніков Д.Е.			
			СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.		СТ	Листів 1

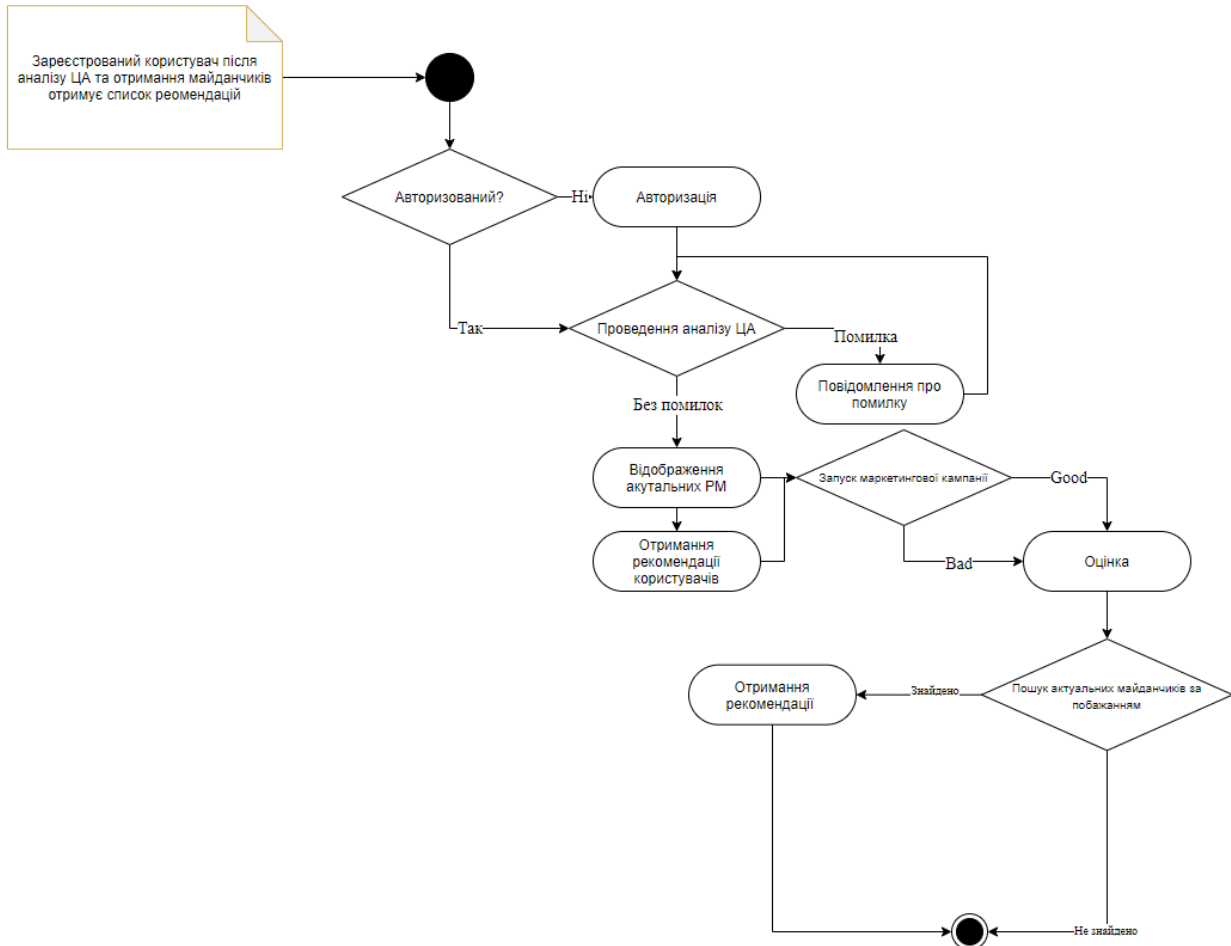
## Декомпозиція функції «Проведення процедури обчислення схожості користувачів»



Розроб.	Мамонтов Ю.В.		Декомпозиція функції «Проведення процедури обчислення схожості користувачів»	
Перевір.	Ситніков Д.Е.			
Н. Контр.	Ситніков Д.Е.			
			КН-14-2	Лист 1
Затверд.	Гребеннік І.В.		СТ	Листів 1

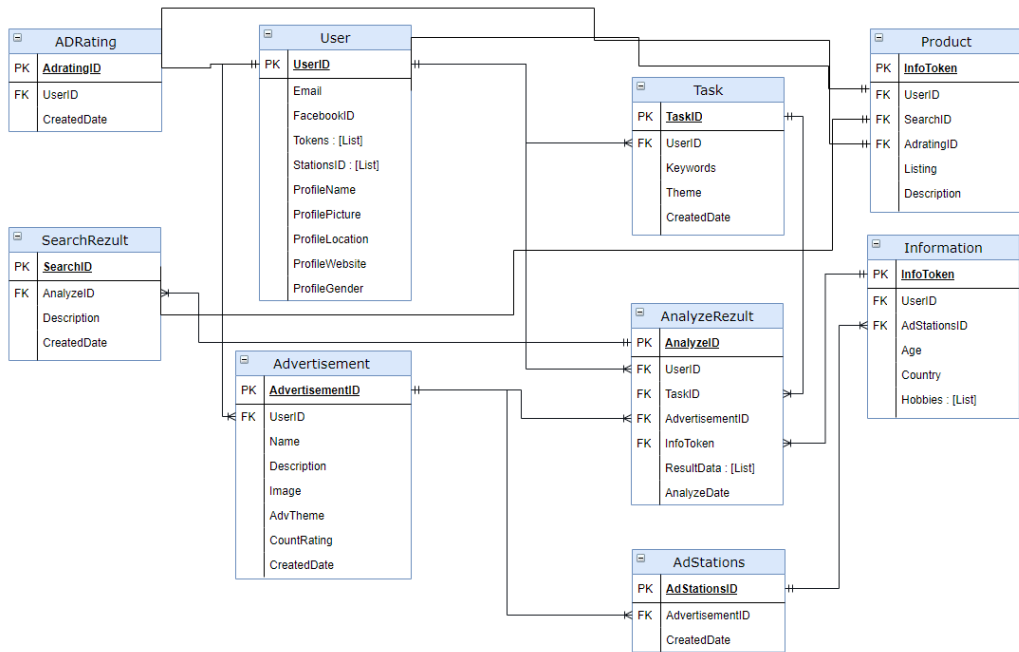


# Діаграма діяльності



Розроб.	Мамонтов Ю.В.			Діаграма діяльності функції	
Перевір.	Ситніков Д.Е.				
Н. Контр.	Ситніков Д.Е.				
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

# Схема бази даних



Розроб.	Мамонтов Ю.В.		Схема бази даних	
Перевір.	Ситніков Д.Е.			
Н. Контр.	Ситніков Д.Е.			
			СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.		СТ	Листів 1

## Сторінка рекомендацій

Україна

18 - 62 р.

Чоловіча

ІНШІ КОРИСТУВАЧІ ПРІЮЦЮЮТЬ ТУТ:



Instagram  
★★★★★



Google ADV  
★★★★★



Facebook  
★★★★★



YouTube  
★★★★★

Розроб.	Мамонтов Ю.В.			Сторінка рекомендацій	
Перевір.	Ситніков Д.Е.				
Н. Контр.	Ситніков Д.Е.				
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ДОДАТОК Б

Текст програми

ГЮИК. 509000.005 – 01 12 01

(позначення документу)

ЗАТВЕРДЖЕНО

ГЮИК. 509000.005 – 01 12 01 – ЛУ

Розробка алгоритму колаборативної фільтрації для роботи з цільовою бізнес-аудиторією та його імплементація в інформаційних системах

Текст програми

ГЮИК. 509000.005 – 01 12 01 – ЛУ

АРКУШІВ 11

2020 р.

Міністерство освіти і науки України  
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»  
керівник атестаційної роботи  
проф. Ситніков Д.Е.

Розробка алгоритму колаборативної фільтрації для роботи з цільовою бізнес-аудиторією та його імплементація в інформаційних системах

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 509000.005 – 01 12 01 – ЛУ

РОЗРОБИВ:  
ст. гр. СПРМ-18-2  
Мамонтов Ю.В.



2020 р.

```

const stem = require('stem-porter');

const process = function(sentences, numberOfTopics, numberOfTermsPerTopic, languages,
alphaValue, betaValue, randomSeed) {
  var result = [];
  var documents = new Array();
  var f = {};
  var vocab=new Array();
  var vocabOrig = {};
  languages = languages || Array('en');

  if (sentences && sentences.length > 0) {
    var stopwords = new Array();

    languages.forEach(function(value) {
      var stopwordsLang = require('./stopwords_' + value + ".js");
      stopwords = stopwords.concat(stopwordsLang.stop_words);
    });

    for(var i=0;i<sentences.length;i++) {
      if (sentences[i]== "") continue;
      documents[i] = new Array();

      var words = sentences[i].split(/[\s,\\"]+/);

      if(!words) continue;
      for(var wc=0;wc<words.length;wc++) {
        var w=words[wc].toLowerCase().replace(/[^a-z\A-Z0-9\u00C0-\u00ff ]+/g, "");
        var wStemmed = stem(w);
        if (w=="" || !wStemmed || w.length==1 || stopwords.indexOf(w.replace("", "")) > -1 ||
stopwords.indexOf(wStemmed) > -1 || w.indexOf("http")==0) continue;
        if (f[wStemmed]) {
          f[wStemmed]=f[wStemmed]+1;
        }
        else if(wStemmed) {
          f[wStemmed]=1;
          vocab.push(wStemmed);
          vocabOrig[wStemmed] = w;
        }
      };
    }
  }

```

```

    documents[i].push(vocab.indexOf(wStemmed));
  }
}

var V = vocab.length;
var M = documents.length;
var K = parseInt(numberOfTopics);
var alpha = alphaValue || 0.1; // per-document distributions over topics
var beta = betaValue || .01; // per-topic distributions over words

lda.configure(documents,V,10000, 2000, 100, 10, randomSeed);
lda.gibbs(K, alpha, beta);

var theta = lda.getTheta();
var phi = lda.getPhi();

var text = "";

//topics
var topTerms=numberOfTermsPerTopic;
for (var k = 0; k < phi.length; k++) {
  var things = new Array();
  for (var w = 0; w < phi[k].length; w++) {
    things.push(""+phi[k][w].toFixed(2)+"_"+vocab[w] + "_" + vocabOrig[vocab[w]]);
  }
  things.sort().reverse();
  //console.log(things);
  if(topTerms>vocab.length) topTerms=vocab.length;

  //console.log("Topic ' + (k + 1));
  var row = [];

  for (var t = 0; t < topTerms; t++) {
    var topicTerm=things[t].split("_")[2];
    var prob=parseFloat(things[t].split("_")[0]*100);
    if (prob<2) continue;

    //console.log("Top Term: ' + topicTerm + ' (' + prob + '%)');
    var term = {};
    term.term = topicTerm;
    term.probability = parseFloat(things[t].split("_")[0]);
    row.push(term);
  }
}

```

```
    }
    result.push(row);
  }
}
return result;
}
function makeArray(x) {
  var a = new Array();
  for (var i=0;i<x;i++) {
    a[i]=0;
  }
  return a;
}
function make2DArray(x,y) {
  var a = new Array();
  for (var i=0;i<x;i++) {
    a[i]=new Array();
    for (var j=0;j<y;j++)
      a[i][j]=0;
  }
  return a;
}
var lda = new function() {
  var documents,z,nw,nd,nwsum,ndsum,thetasum,phisum,V,K,alpha,beta;
  var THIN_INTERVAL = 20;
  var BURN_IN = 100;
  var ITERATIONS = 1000;
  var SAMPLE_LAG;
  var RANDOM_SEED;
  var dispcol = 0;
  var numstats=0;
  this.configure = function (docs,v,iterations,burnIn,thinInterval,sampleLag,randomSeed) {
    this.ITERATIONS = iterations;
    this.BURN_IN = burnIn;
    this.THIN_INTERVAL = thinInterval;
    this.SAMPLE_LAG = sampleLag;
    this.RANDOM_SEED = randomSeed;
    this.documents = docs;
    this.V = v;
    this.dispcol=0;
    this.numstats=0;
  }
}
```

```

this.initialState = function (K) {
    var i;
    var M = this.documents.length;
    this.nw = make2DArray(this.V,K);
    this.nd = make2DArray(M,K);
    this.nwsum = makeArray(K);
    this.ndsum = makeArray(M);
    this.z = new Array(); for (i=0;i<M;i++) this.z[i]=new Array();
    for (var m = 0; m < M; m++) {
        var N = this.documents[m].length;
        this.z[m] = new Array();
        for (var n = 0; n < N; n++) {
            var topic = parseInt(""+(this.getRandom() * K));
            this.z[m][n] = topic;
            this.nw[this.documents[m][n]][topic]++;
            this.nd[m][topic]++;
            this.nwsum[topic]++;
        }
        this.ndsum[m] = N;
    }
}

this.gibbs = function (K,alpha,beta) {
    var i;
    this.K = K;
    this.alpha = alpha;
    this.beta = beta;
    if (this.SAMPLE_LAG > 0) {
        this.thetasum = make2DArray(this.documents.length,this.K);
        this.phisum = make2DArray(this.K,this.V);
        this.numstats = 0;
    }
    this.initialState(K);
    //document.write("Sampling " + this.ITERATIONS
    // + " iterations with burn-in of " + this.BURN_IN + " (B/S="
    // + this.THIN_INTERVAL + ").<br/>");
    for (i = 0; i < this.ITERATIONS; i++) {
        for (var m = 0; m < this.z.length; m++) {
            for (var n = 0; n < this.z[m].length; n++) {
                var topic = this.sampleFullConditional(m, n);
                this.z[m][n] = topic;
            }
        }
    }
}

```

```

if ((i < this.BURN_IN) && (i % this.THIN_INTERVAL == 0)) {
    //document.write("B");
    this.dispcol++;
}
if ((i > this.BURN_IN) && (i % this.THIN_INTERVAL == 0)) {
    //document.write("S");
    this.dispcol++;
}
if ((i > this.BURN_IN) && (this.SAMPLE_LAG > 0) && (i % this.SAMPLE_LAG == 0))
{
    this.updateParams();
    //document.write("|");
    if (i % this.THIN_INTERVAL != 0)
        this.dispcol++;
}
if (this.dispcol >= 100) {
    //document.write("*<br/>");
    this.dispcol = 0;
}
}
}

this.sampleFullConditional = function(m,n) {
    var topic = this.z[m][n];
    this.nw[this.documents[m][n]][topic]--;
    this.nd[m][topic]--;
    this.nwsum[topic]--;
    this.ndsum[m]--;
    var p = makeArray(this.K);
    for (var k = 0; k < this.K; k++) {
        p[k] = (this.nw[this.documents[m][n]][k] + this.beta) / (this.nwsum[k] + this.V * this.beta)
            * (this.nd[m][k] + this.alpha) / (this.ndsum[m] + this.K * this.alpha);
    }
    for (var k = 1; k < p.length; k++) {
        p[k] += p[k - 1];
    }
    var u = this.getRandom() * p[this.K - 1];
    for (topic = 0; topic < p.length; topic++) {
        if (u < p[topic])
            break;
    }
    this.nw[this.documents[m][n]][topic]++;
}

```

```

this.nd[m][topic]++;
this.nwsum[topic]++;
this.ndsum[m]++;
return topic;
}

this.updateParams =function () {
  for (var m = 0; m < this.documents.length; m++) {
    for (var k = 0; k < this.K; k++) {
      this.thetasum[m][k] += (this.nd[m][k] + this.alpha) / (this.ndsum[m] + this.K *
this.alpha);
    }
  }
  for (var k = 0; k < this.K; k++) {
    for (var w = 0; w < this.V; w++) {
      this.phisum[k][w] += (this.nw[w][k] + this.beta) / (this.nwsum[k] + this.V * this.beta);
    }
  }
  this.numstats++;
}

this.getTheta = function() {
  var theta = new Array(); for(var i=0;i<this.documents.length;i++) theta[i] = new Array();
  if (this.SAMPLE_LAG > 0) {
    for (var m = 0; m < this.documents.length; m++) {
      for (var k = 0; k < this.K; k++) {
        theta[m][k] = this.thetasum[m][k] / this.numstats;
      }
    }
  } else {
    for (var m = 0; m < this.documents.length; m++) {
      for (var k = 0; k < this.K; k++) {
        theta[m][k] = (this.nd[m][k] + this.alpha) / (this.ndsum[m] + this.K * this.alpha);
      }
    }
  }
  return theta;
}

this.getPhi = function () {
  var phi = new Array(); for(var i=0;i<this.K;i++) phi[i] = new Array();
  if (this.SAMPLE_LAG > 0) {

```

```

    for (var k = 0; k < this.K; k++) {
      for (var w = 0; w < this.V; w++) {
        phi[k][w] = this.phisum[k][w] / this.numstats;
      }
    }
  } else {
    for (var k = 0; k < this.K; k++) {
      for (var w = 0; w < this.V; w++) {
        phi[k][w] = (this.nw[w][k] + this.beta) / (this.nwsum[k] + this.V * this.beta);
      }
    }
  }
  return phi;
}

```

```

this.getRandom = function() {
  if (this.RANDOM_SEED) {
    // generate a pseudo-random number using a seed to ensure reproducible results.
    var x = Math.sin(this.RANDOM_SEED++) * 1000000;
    return x - Math.floor(x);
  } else {
    // use standard random algorithm.
    return Math.random();
  }
}
}

```

```
module.exports = process;
```

Модуль авторизації через соціальну мережу Facebook:

```

const passport = require('passport');
const FacebookStrategy = require('passport-facebook').Strategy;

const User = require('../models/User');

exports.setup = function (config) {
  passport.use(new FacebookStrategy({
    clientID : config.facebook.clientID,
    clientSecret : config.facebook.clientSecret,

```

```

    callbackURL : '/api/v1/auth/facebook/callback',
    profileFields: ['emails', 'name', 'link']
  },
  function (req, accessToken, refreshToken, profile, done) {
    if (req.user) {
      User.findOne({ facebook: profile.id }, (err, existingUser) => {
        if (err) {
          return done(err);
        }
        if (existingUser) {
          req.flash('errors', { msg: 'There is already a Facebook account that belongs to you. Sign in with
that account or delete it, then link it with your current account.' });
          done(err);
        } else {
          User.findById(req.user.id, (err, user) => {
            if (err) {
              return done(err);
            }
            user.facebook = profile.id;
            user.tokens.push({ kind: 'facebook', accessToken });
            user.profile.name = user.profile.name || `${profile.name.givenName}
${profile.name.familyName}`;
            user.profile.gender = user.profile.gender || profile._json.gender;
            user.profile.picture = user.profile.picture ||
`https://graph.facebook.com/${profile.id}/picture?type=large`;
            user.save((err) => {
              req.flash('info', { msg: 'Facebook account has been linked.' });
              done(err, user);
            });
          });
        }
      });
    } else {
      User.findOne({ facebook: profile.id }, (err, existingUser) => {
        if (err) {
          return done(err);
        }
        if (existingUser) {
          return done(null, existingUser);
        }
      });
    }
  }
}

```

```

    }
    User.findOne({email: profile._json.email}, (err, existingEmailUser) => {
      if (err) {
        return done(err);
      }
      if (existingEmailUser) {
        req.flash('errors', {msg: 'There is already an account using this email address. Sign in to that
account and link it with Facebook manually from Account Settings.});
        done(err);
      } else {
        const user = new User();
        user.email = profile._json.email;
        user.facebook = profile.id;
        user.tokens.push({kind: 'facebook', accessToken});
        user.profile.name = `${profile.name.givenName} ${profile.name.familyName}`;
        user.profile.gender = profile._json.gender;
        user.profile.picture = `https://graph.facebook.com/${profile.id}/picture?type=large`;
        user.profile.location = (profile._json.location) ? profile._json.location.name : "";
        user.save((err) => {
          done(err, user);
        });
      }
    });
  });
}
})))
};

module.exports.setTokenCookie = (req, res) => {
  if (!req.user) {
    return res.status(404).json({
      message: 'Something went wrong, please try again.'
    });
  } else {
    const token = this.signToken(req.user._id);
    res.cookie('JWT', token);
    return res.redirect('/index');
  }
};

```

```
/**
 * Returns a jwt token signed by the app secret
 */
this.signToken = (id) => {
  return jwt.sign({
    _id: id
  }, config.secrets.session, {
    expiresIn: 60 * 60 * 24 * 7
  });
};

function isAuthenticated() {
  return compose()
    .use((req, res, next) => {
      if (req.cookies && req.cookies.hasOwnProperty('JWT')) {
        req.headers.authorization = 'Bearer ' + req.cookies.JWT;
      }
      validateJwt(req, res, next);
    })
    .use((req, res, next) => {
      User.findById(req.user._id)
        .exec((err, user) => {
          if (err) {
            return next(err);
          } else if (!user) {
            return res.status(401).send('Unauthorized');
          } else {
            req.user = user;
            return next();
          }
        });
    })
    .use((err, req, res, next) => {
      if (err.name === 'UnauthorizedError') {
        var errors = [];
        errors.push(err);
        return res.status(401).send(errors);
      }
    });
};
```

```
}  
module.exports.isAuthenticated = isAuthenticated;  
  
module.exports.signToken = this.signToken;
```

