# Solution for Genentech Cervical Cancer Screening, 2nd place

Kohei Ozaki[1], Dmitry Efimov[2], Lucas Silva[3], and Gilberto Titericz[4]

[1]eowner@gmail.com
[2]diefimov@gmail.com
[3]lucas.eustaquio@gmail.com
[4]titericz@yahoo.com

## 1 Approach summary

The goal of this competition is to predict which women will not be screened for cervical cancer on the recommended schedule. Identifying at-risk populations will make education and other intervention efforts more effective, ideally ultimately reducing the number of women who die from this disease. Our solution mostly based on the feature engineering. The document is constructed in the following way: first 4 sections describe the features engineered by each team player, the last sections describe the model and the way to obtain the predictions.

## 2 Gilberto: feature description

Three types of features are constructed (241 features in total): counts by feature, count by level (CountVectorizer) and likelihood features. Also there is one special feature built using Vowpal Wabbit [2]

## 2.1 Counts

**Dataset: patient_activity_head.csv**:

- **f1_0**: sum(activity_type==R) by patient_id
- **f1_1**: sum(activity_type==A) by patient_id
- **f1_2**: max(date) by patient_id
- **f1_3**: (max(date)-min(date))/(1+.N) by patient_id
- **f1_4**: maximum number of registers by year by patient_id
- **f1_5**: max(date)-min(date) by patient_id
- **f1_6**: min(date) by patient_id
- **f1_N**: number of registers by patient_id
- **f1_7**: mean number of registers by year by patient_id
- **f1_8**: mean number of registers by month by patient_id

**Dataset: surgical_head.csv**:

- **f2_N**: number of registers by patient_id
- **f2_0**: sum(procedure_type_code==HXPR) by patient_id
- **f2_1**: sum(procedure_type_code==HX05) by patient_id
- **f2_2**: sum(procedure_type_code==HX01) by patient_id
- **f2_3**: sum(procedure_type_code==HX01) by patient_id
- **f2_4**: sum(procedure_type_code==HX02) by patient_id
- **f2_5**: sum(procedure_type_code==HX03) by patient_id
- **f2_6**: sum(procedure_type_code==0001) by patient_id
- **f2_7**: sum(procedure_type_code==0002) by patient_id
- **f2_8**: sum(procedure_type_code==0003) by patient_id

- **f2_9**: sum(procedure_type_code==0004) by patient_id
- **f2_10**: sum(procedure_type_code==0005) by patient_id
- **f2_11**: sum(procedure_type_code==0006) by patient_id
- **f2_12**: sum(place_of_service==INPATIENT) by patient_id
- **f2_13**: sum(place_of_service==OUTPATIENT) by patient_id
- **f2_14**: sum(place_of_service==OTHER) by patient_id
- **f2_15**: sum(place_of_service==CLINIC) by patient_id
- **f2_16**: sum(place_of_service==UNKNOWN) by patient_id
- **f2_17**: sum(plan_type==COMMERCIAL) by patient_id
- **f2_18**: sum(plan_type==MEDICARE) by patient_id
- **f2_19**: sum(plan_type==MEDICAID) by patient_id
- **f2_20**: sum(plan_type==GOVERNMENT) by patient_id
- **f2_21**: sum(plan_type==CASH) by patient_id
- **f2_22**: sum(plan_type==UNKNOWN) by patient_id
- **f2_23**: sum(primary_physician_role==ATG) by patient_id
- **f2_24**: sum(primary_physician_role==‘’) by patient_id
- **f2_25**: sum(primary_physician_role==OTH) by patient_id
- **f2_26**: sum(primary_physician_role==OPR) by patient_id
- **f2_N2**: number of registers by procedure_type_code, place_of_service
- **f2_N3**: number of registers by procedure_type_code, place_of_service, plan_type
- **f2_N4**: number of registers by procedure_type_code, place_of_service, plan_type, primary_physician_role

**Dataset: diagnosis_head.csv**:

- **f3_N**: number of registers by patient_id
- **f3_0**: sum( primary_physician_role==ATG ) by patient_id
- **f3_1**: sum( primary_physician_role==RND ) by patient_id
- **f3_2**: sum( primary_physician_role==‘’) by patient_id
- **f3_3**: sum( primary_physician_role==PRV ) by patient_id
- **f3_4**: sum( primary_physician_role==ORD ) by patient_id
- **f3_5**: sum( primary_physician_role==UNK ) by patient_id
- **f3_6**: sum( primary_physician_role==OTH ) by patient_id
- **f3_7**: sum( primary_physician_role==OPR ) by patient_id
- **f3_8**: sum( claim_type==HX ) by patient_id
- **f3_9**: sum( claim_type==MX ) by patient_id
- **f3_10**: number of levels of claim_id by patient_id
- **f3_11**: number of levels of diagnosis_code by patient_id

**Dataset: prescription_head.csv**:

- **f4_N**: number of registers by patient_id
- **f4_1**: number of levels of payment_type by patient_id
- **f4_2**: sum( payment_type==COMMERCIAL ) bypatient_id
- **f4_3**: sum( payment_type==CASH ) by patient_id
- **f4_4**: sum( payment_type==MANAGED MEDICAID ) by patient_id
- **f4_5**: sum( payment_type==MEDICAID ) by patient_id
- **f4_6**: sum( payment_type==MEDICARE ) by patient_id
- **f4_7**: sum( payment_type==ASSISTANCE PROGRAMS ) by patient_id
- **f4_8**: number of levels of claim_id by patient_id

- **f4_9**: number of levels of drug_id by patient_id
- **f4_10**: number of levels of refill_code by patient_id
- **f4_11**: mean(days_supply) by patient_id

**Dataset: procedure_head.csv**:

- **f5_1**: number of registers by patient_id
- **f5_2**: number of levels of claim_id bypatient_id
- **f5_3**: number of levels of claim_line_item bypatient_id
- **f5_4**: number of levels of claim_type by patient_id
- **f5_5**: number of levels of procedure_code by patient_id
- **f5_6**: number of levels of place_of_service by patient_id
- **f5_7**: number of levels of plan_type by patient_id
- **f5_8**: number of levels of primary_practitioner_id bypatient_id
- **f5_9**: sum(units_administered) by patient_id
- **f5_10**: sum(charge_amount) by patient_id
- **f5_11**: number of levels of primary_physician_role by patient_id
- **f5_12**: number of levels of attending_practitioner_id by patient_id
- **f5_13**: number of levels of referring_practitioner_id by patient_id
- **f5_14**: number of levels of rendering_practitioner_id by patient_id
- **f5_15**: number of levels of ordering_practitioner_id by patient_id
- **f5_16**: number of levels of operating_practitioner_id by patient_id
- **f5_17**: sum(claim_type==HX) by patient_id
- **f5_18**: sum(claim_type==MX) by patient_id
- **f5_19**: sum(primary_physician_role==ATG) by patient_id

- **f5_20**: sum(primary_physician_role==RND) by patient_id
- **f5_21**: sum(primary_physician_role==‘’) by patient_id
- **f5_22**: sum(primary_physician_role==PRV) by patient_id
- **f5_23**: sum(primary_physician_role==ORD) by patient_id
- **f5_24**: sum(primary_physician_role==UNK) by patient_id
- **f5_25**: sum(primary_physician_role==OTH) by patient_id
- **f5_26**: sum(primary_physician_role==OPR) by patient_id
- **f5_27**: min(procedure_date) by patient_id
- **f5_28**: max(procedure_date) by patient_id
- **f5_29**: (f5_28-f5_27)/f5_1
- **F5_30**: max(charge_amount) by patient_id
- **F5_32**: max(units_administered) by patient_id

## 2.2 Likelihood features

All likelihood features are calculated using out-of-fold predictions with the formula:

$$LL = \frac{L \cdot \bar{y} + \sum_{i \in G} y_i}{L + |G|},$$

where $G$ is the set of indices for the training examples such that set of chosen raw features has unique values for the examples from $G$, $|G|$ is a size of group $G$, $L$ is a number of levels for raw features subset, $\bar{y}$ is an average of output for all training examples.

The prefix of each feature name can be “mean” or “max”. When it begin with “mean” it means that likelihood feature is the mean of all likelihoods by patient_id. When it begins with “max” it means that likelihood feature is only the max value of all likelihood by patient_id.

### 2.2.1 1-Way Likelihood Features

Dataset: surgical_head.csv:

- **mean_sh_surgical_code**: by surgical_code
- **mean_sh_place_of_service**: by place_of_service
- **mean_sh_plan_type**: by plan_type
- **mean_sh_practitioner_id**: by practitioner_id
- **mean_sh_primary_physician_role**: by primary_physician_role
- **mean_sh_sc1**: by number of characters in surgical_code
- **mean_sh_sc2**: by first word in surgical_code
- **mean_sh_sc3**: by last word in surgical_code
- **mean_sh_pid_1**: by physician_id
- **mean_sh_pid_2**: by state
- **mean_sh_pid_3**: by specialty_code
- **mean_sh_pid_4**: by specialty_description
- **mean_sh_pid_5**: by CBSA
- **mean_sh_claim_id**: by claim_id
- **mean_sh_procedure_type_code**: by procedure_type_code
- **mean_sh_surgical_procedure_date**: by surgical_procedure_date
- **max_sh_surgical_code**: by surgical_code
- **max_sh_place_of_service**: by place_of_service
- **max_sh_plan_type**: by plan_type
- **max_sh_practitioner_id**: by practitioner_id
- **max_sh_primary_physician_role**: by primary_physician_role

- **max_sh_sc1**: by number of characters in surgical_code
- **max_sh_sc2**: by first word in surgical_code
- **max_sh_sc3**: by last word in surgical_code
- **max_sh_pid_1**: by physician_id
- **max_sh_pid_2**: by state
- **max_sh_pid_3**: by specialty_code
- **max_sh_pid_4**: by specialty_description
- **max_sh_pid_5**: by CBSA
- **max_sh_claim_id**: by claim_id
- **max_sh_procedure_type_code**: by procedure_type_code
- **max_sh_surgical_procedure_date**: by surgical_procedure_date

**Dataset: diagnosis_head.csv**:

- **max_dh_diagnosis_code**: by diagnosis_code
- **max_dh_primary_practitioner_id**: by primary_practitioner_id
- **mean_dh_diagnosis_code**: by diagnosis_code
- **mean_dh_primary_practitioner_id**: by primary_practitioner_id
- **max_dh2_diagnosis_code**: by diagnosis_code(letters removed)
- **max_dh_diagnosis_code_sub**: by diagnosis_code(numbers before .)
- **max_dh_diagnosis_code_ind**: by diagnosis_code(numbers after .)
- **mean_dh2_diagnosis_code**: by diagnosis_code(letters removed)
- **mean_dh_diagnosis_code_sub**: by diagnosis_code(numbers before .)
- **mean_dh_diagnosis_code_ind**: by diagnosis_code(numbers after .)
- **max_ph_pid_dh1**: by physician_id

- **mean_ph_pid_dh1**: by physician_id
- **max_ph_pid_dh2**: by state
- **mean_ph_pid_dh2**: by state
- **max_ph_pid_dh3**: by specialty_code
- **mean_ph_pid_dh3**: by specialty_code
- **max_ph_pid_dh4**: by CBSA
- **mean_ph_pid_dh4**: by CBSA

**Dataset: prescription_head.csv**:

- **max_pch_drug_id**: by drug_id
- **mean_pch_drug_id**: by drug_id
- **max_pch_practitioner_id**: by practitioner_id
- **mean_pch_practitioner_id**: by practitioner_id
- **max_pch_refill_code**: by refill_code
- **mean_pch_refill_code**: by refill_code
- **max_pch_days_supply**: by days_supply
- **mean_pch_days_supply**: by days_supply
- **max_dg_drug_1**: by drug_name
- **mean_dg_drug_1**: by drug_name
- **max_dg_drug_2**: by BGI
- **mean_dg_drug_2**: by BGI
- **max_dg_drug_3**: by BB_USC_name
- **mean_dg_drug_3**: by BB_USC_name
- **max_dg_drug_4**: by drug_strength

- **mean_dg_drug_4**: by drug_strength

**Dataset: procedure_head.csv**:

- **max_proch_claim_line_item**: by claim_line_item
- **mean_proch_claim_line_item**: by claim_line_item
- **max_proch_procedure_code**: by procedure_code
- **mean_proch_procedure_code**: by procedure_code
- **max_proch_place_of_service**: by place_of_service
- **mean_proch_place_of_service**: by place_of_service
- **max_proch_plan_type**: by plan_type
- **mean_proch_plan_type**: by plan_type
- **max_proch_primary_practitioner_id**: by primary_practitioner_id
- **mean_proch_primary_practitioner_id**: by primary_practitioner_id
- **mean_proch_attending_practitioner_id**: by attending_practitioner_id
- **max_proch_referring_practitioner_id**: by rendering_practitioner_id
- **max_proch_rendering_practitioner_id**: by rendering_practitioner_id
- **max_proch_ordering_practitioner_id**: by ordering_practitioner_id
- **mean_proch_ordering_practitioner_id**: by ordering_practitioner_id
- **max_proch_operating_practitioner_id**: by operating_practitioner_id
- **mean_proch_operating_practitioner_id**: by operating_practitioner_id

### 2.2.2 2-Way Likelihood Features

Dataset: diagnosis_head.csv:

- **mean___diagnosis_code_claim_type**: by diagnosis_code, claim_type
- **max___diagnosis_code_claim_type**: by diagnosis_code, claim_type
- **mean_dh_diagnosis_code_diagnosis_date**: by diagnosis_code, diagnosis_date
- **max_dh_diagnosis_code_diagnosis_date**: by diagnosis_code, diagnosis_date
- **mean_dh_diagnosis_code_primary_practitioner_id**: by diagnosis_code, primary_practitioner_id
- **max_dh_diagnosis_code_primary_practitioner_id**: by diagnosis_code, primary_practitioner_id
- **mean_dh_diagnosis_code_primary_physician_role**: by diagnosis_code, primary_physician_role
- **max_dh_diagnosis_code_primary_physician_role**: by diagnosis_code, primary_physician_role
- **mean_dh_diagnosis_code_pid_dh1**: by diagnosis_code, physician_id
- **max_dh_diagnosis_code_pid_dh1**: by diagnosis_code, physician_id
- **mean_dh_diagnosis_code_pid_dh2**: by diagnosis_code, state
- **max_dh_diagnosis_code_pid_dh2**: by diagnosis_code, state
- **mean_dh_diagnosis_code_pid_dh3**: by diagnosis_code, specialty_code
- **max_dh_diagnosis_code_pid_dh3**: by diagnosis_code, specialty_code
- **mean_dh_diagnosis_code_pid_dh4**: by diagnosis_code, CBSA
- **max_dh_diagnosis_code_pid_dh4**: by diagnosis_code, CBSA
- **mean_dh_primary_physician_role_pid_dh1**: by primary_physician_role, physician_id

- **max_dh_primary_physician_role_pid_dh1**: by primary_physician_role, physician_id
- **mean_dh_primary_physician_role_pid_dh3**: by primary_physician_role, specialty_code
- **max_dh_primary_physician_role_pid_dh3**: by primary_physician_role, specialty_code
- **mean_dh_primary_physician_role_pid_dh4**: by primary_physician_role, CBSA
- **max_dh_primary_physician_role_pid_dh4**: by primary_physician_role, CBSA
- **mean_dh_primary_physician_role_primary_practitioner_id**: by primary_physician_role, primary_practitioner_id
- **max_dh_primary_physician_role_primary_practitioner_id**: by primary_physician_role, primary_practitioner_id
- **mean_dh_primary_practitioner_id_diagnosis_date**: by diagnosis_code, diagnosis_date
- **max_dh_primary_practitioner_id_diagnosis_date**: by diagnosis_code, diagnosis_date
- **mean_dh_primary_practitioner_id_pid_dh1**: by primary_practitioner_id, physician_id
- **max_dh_primary_practitioner_id_pid_dh1**: by primary_practitioner_id, physician_id

The following list of features takes into account some leaks found in datasets construction regarding some procedures removed. To generate them we used the following derivative features:

- **flag1**: number of claim_id in diagnosis_head.csv
- **flag2**: number of claim_id in procedure_head.csv
- **flag3**: maximum(claim_line_item) - number of claim_id in procedure_head.csv

- **flag4**: flag1 + 8 · flag2 + 64 · flag3

**Dataset: diagnosis_head.csv and procedure_head.csv**:

- **mean___diagnosis_code_flag3**: by diagnosis_code, flag3
- **max___diagnosis_code_flag3**: by diagnosis_code, flag3
- **mean___diagnosis_code_flag4**: by diagnosis_code, flag4
- **max___diagnosis_code_flag4**: by diagnosis_code, flag4
- **mean___diagnosis_code_flag2**: by diagnosis_code, flag2
- **max___diagnosis_code_flag2**: by diagnosis_code, flag2
- **mean___diagnosis_code_flag1**: by diagnosis_code, flag1
- **max___diagnosis_code_flag1**: by diagnosis_code, flag1
- **mean___flag1_flag2**: by flag1, flag2
- **max___flag1_flag2**: by flag1, flag2
- **mean___flag1_flag3**: by flag1, flag3
- **max___flag1_flag3**: by flag1, flag3
- **mean___flag2_flag3**: by flag2, flag3
- **max___flag2_flag3**: by flag2, flag3
- **mean___year_flag1**: by year, flag1
- **max___year_flag1**: by year, flag1

### 2.2.3 3-Way Likelihood Features

The following list of features takes into account some leaks found in datasets construction regarding some procedures removed. To generate them we used the following derivative features:

- **flag1**: mumber of claim_id in diagnosis_head.csv

- **flag2**: number of claim_id in procedure_head.csv
- **flag3**: maximum(claim_line_item) - number of claim_id in procedure_head.csv

**Dataset: diagnosis_head.csv and procedure_head.csv**:

- **mean___flag1_flag2_flag3**: by flag1, flag2, flag3
- **max___flag1_flag2_flag3**: by flag1, flag2, flag3
- **mean___diagnosis_code_flag1_flag2**: by diagnosis_code, flag1, flag2
- **max___diagnosis_code_flag1_flag2**: by diagnosis_code, flag1, flag2
- **mean___diagnosis_code_flag1_flag3**: by diagnosis_code, flag1, flag3
- **max___diagnosis_code_flag1_flag3**: by diagnosis_code, flag1, flag3
- **mean___diagnosis_code_flag2_flag3**: by diagnosis_code, flag2, flag3
- **max___diagnosis_code_flag2_flag3**: by diagnosis_code, flag2, flag3

#### 2.2.4 4-Way Likelihood Features

The following list of features takes into account some leaks found in datasets construction regarding some procedures removed. To generate them we used the following derivative features:

- **flag1**: number of claim_id in diagnosis_head.csv
- **flag2**: number of claim_id in procedure_head.csv
- **flag3**: maximum(claim_line_item) - number of claim_id in procedure_head.csv

**Dataset: diagnosis_head.csv and procedure_head.csv**:

- **mean___diagnosis_code_primary_practitioner_id_flag3_claim_type**: by diagnosis_code, primary_practitioner_id, flag3, claim_type
- **max___diagnosis_code_primary_practitioner_id_flag3_claim_type**: by diagnosis_code, primary_practitioner_id, flag3, claim_type
- **mean___diagnosis_code_primary_practitioner_id_flag3_primary_physician_role**: by diagnosis_code, primary_practitioner_id, flag3, primary_physician_role

- **max___diagnosis_code_primary_practitioner_id_flag3_primary_physician_role**: by diagnosis_code, primary_practitioner_id, flag3, primary_physician_role
- **mean___diagnosis_code_primary_practitioner_id_flag2_claim_type**: by diagnosis_code, primary_practitioner_id, flag2, claim_type
- **max___diagnosis_code_primary_practitioner_id_flag2_claim_type**: by diagnosis_code, primary_practitioner_id, flag2, claim_type
- **mean___diagnosis_code_primary_practitioner_id_flag2_primary_physician_role**: by diagnosis_code, primary_practitioner_id, flag2, primary_physician_role
- **max___diagnosis_code_primary_practitioner_id_flag2_primary_physician_role**: by diagnosis_code, primary_practitioner_id, flag2, primary_physician_role

### 2.3 Vowpal Rabbit feature

**Datasets: surgical_head.csv, surgical_code.csv and physicians.csv**:

- **vw1**: Vowpal Wabbit Meta feature

This feature is generated as follows: features from surgical_code and physicians are merged in surgical_head, then features are concatenated as text and run over Vowpal Wabbit to build an out-of-fold logistic meta feature.

## 3 Kohei: feature description

Meta feature is trained by XGBoost [1]. The feature set has dense (low-dimensional) features and sparse (high-dimensional) features. Most of categorical features on this feature set is encoded by LabelEncoder [7]. For each patient, dense features are calculated by the following definition:

| source table | definition |
|---|---|
| patients | age_group encoded by LabelEncoder |
| patients | state encoded by LabelEncoder |
| patients | ethinicity encoded by LabelEncoder |
| patients | household_income encoded by LabelEncoder |
| patients | education_level encoded by LabelEncoder |
| prescription | statistic values (min, max) of days_supply |
| patient_activity_head | highest value of activity_year |
| procedure | existance of practitioner_id (0 or 1) |
| procedure | existance of referring_id (0 or 1) |
| procedure | existance of operating_id (0 or 1) |
| surgical | unique count of claim_id |
| procedure | number of records on procedur table |
| procedure | unique count of claim_id |
| prescription | unique count of claim_id |
| diagnosis | number of records on diagnosis table |
| diagnosis | unique count of claim_id |
| prescription | unique count of diagnosis_date |

To handle categorical variables of transaction table, this feature set uses 1-of-K coding. For each patient, sparse features on kohei_type1 are calculated by the following definition:

| source table | definition |
|---|---|
| diagnosis | term counts of drug_id |
| diagnosis, phycisian | term counts of phycisian's specialty_description |
| diagnosis, phycisian | term counts of phycisian's state |
| prescription, drug | term counts of bb_usc_code |
| prescription, drug | term counts of drug_generic_name |
| prescription, drug | term counts of drug_manufacturer |
| procedure, phycisian | term counts of phycisian's specialty_description |
| diagnosis | term counts of diagnosis_code |
| diagnosis | term counts of primary_physician_role |
| procedure | term counts of procedure_code |
| procedure | term counts of place_of_service |

| source table | definition |
|---|---|
| procedure | term counts of plan_type |
| procedure | term counts of physician_state |
| diagnosis, diagnosis_code | 1-gram of diagnosis_description |
| procedure, procedure_code | 1-gram of procedure_description |

# 4 Lucas: feature description

To generate some features additional tables have been created by merging the provided tables. For example, tables procedure_head2 and diagnosis_head2 are generated from procedure_head and diagnosis_head by removing claim_id without missed rows. Mainly four types of features were used.

1. Raw information given in patient file
2. Count of ocurrences of a certain condition
3. Smoothed likelihood of calculated based on the subset of raw categorical features
4. A few extra features that do not fit in any of the previous categories

More information on each type of features, as well as a description of each will be described in the subsections below.

## 4.1 Raw information on patient file

All features were transformed using ordinal encoding using lexical order. When sorting the values of those features, the smallest one becomes the ordinal value, the second smallest 2, and so on. The description of features from this class is following:

- **patient_age_group**
- **patient_gender**
- **patient_state**
- **ethinicity**
- **household_income**
- **education_level**

## 4.2 Count features

- **act_r_count**: count by patient of all activities with activity_type value equals 'R' in the patient_activity_head table
- **act_a_count**: count by patient of all activities with activity_type value equals 'A' in the patient_activity_head table
- **diag_count**: count by patient of the number of diagnosis per patient in table diagnosis_head
- **diag_desc_infect**: count by patient of diagnostics with 'infection' in its description
- **diag_desc_malig**: count by patient of diagnostics with 'malignant' in its description
- **diag_desc_carcin**: count by patient of diagnostics with 'carcinoma' in its description
- **diag_desc_oma**: count by patient of diagnostics with 'oma' in its description
- **diag_desc_hodk**: count by patient of diagnostics with 'hodgkins' in its description
- **diag_desc_leuk**: count by patient of diagnostics with 'leukemia' in its description
- **diag_desc_benig**: count by patient of diagnostics with 'benign' in its description
- **diag_desc_tuberc**: count by patient of diagnostics with 'tuberculosis' in its description
- **diag_desc_virus**: count by patient of diagnostics with 'virus' in its description
- **diag_desc_diab**: count by patient of diagnostics with 'diabetes' in its description
- **diag_desc_vag**: count by patient of diagnostics with 'vagina' in its description

- **diag_desc_smr**: count by patient of diagnostics with 'smear' in its description
- **diag_desc_cvc**: count by patient of diagnostics with 'cervical' in its description
- **diag_desc_cvx**: count by patient of diagnostics with 'cervix' in its description
- **diag_desc_ppl**: count by patient of diagnostics with 'papiloma' in its description
- **diag_desc_chm**: count by patient of diagnostics with 'chlamy' in its description
- **diag_desc_gyn**: count by patient of diagnostics with 'gyneco' in its description
- **diag_desc_hiv**: count by patient of diagnostics with 'hiv' in its description
- **diag_phys_nrs_mw**: count by patient of diagnostics whose pratictioner specialty description contains 'NURSE MIDWIFE'
- **diag_phys_mfm**: count by patient of diagnostics whose pratictioner specialty description contains 'MATERNAL AND FETAL MEDICINE'
- **diag_phys_pc**: count by patient of diagnostics whose pratictioner specialty description contains 'PATHOLOGY, CYTOPATHOLOGY
- **diag_phys_gy**: count by patient of diagnostics whose pratictioner specialty description contains 'GYNECOLOGY'
- **diag_phys_ap**: count by patient of diagnostics whose pratictioner specialty description contains 'ANATOMIC PATHOLOGY'
- **diag_phys_og**: count by patient of diagnostics whose pratictioner specialty description contains 'OBSTETRICS AND GYNECOLOGY'
- **diag_phys_pac**: count by patient of diagnostics whose pratictioner specialty description contains 'PATHOLOGY, ANATOMIC/CLINICAL'

- **diag_phys_dp**: count by patient of diagnostics whose pratictioner specialty description contains 'DERMATOPATHOLOGY'
- **diag_phys_dt**: count by patient of diagnostics whose pratictioner specialty description contains 'DERMATOLOGY'
- **diag_phys_pccm**: count by patient of diagnostics whose pratictioner specialty description contains 'PULMONARY CRITICAL CARE MEDICINE'
- **diag_phys_nph**: count by patient of diagnostics whose pratictioner specialty description contains 'NEPHROLOGY'
- **diag_phys_pr**: count by patient of diagnostics whose pratictioner specialty description contains 'PEDIATRIC RADIOLOGY'
- **diag_phys_pd**: count by patient of diagnostics whose pratictioner specialty description contains 'PEDIATRICS'
- **diag_no_proc**: count by patient of diagnostics without any precedure associated
- **proc_count**: count by patient of the total number of procedures
- **proc_no_diag**: count by patient of procedures without any diagnostics associated
- **presc_count**: count by patient of the total number of prescriptions
- **surg_count**: count by patient of the total number of surgeries
- **pat_ct_ds**: count by patient of the number of claims that have diagnosis and surgery
- **pat_ct_d**: count by patient of the number of claims that have only diagnosis
- **pat_ct_ps**: count by patient of the number of claims that have only procedures and surgeries
- **pat_ct_dps**: count by patient of the number of claims that have diagnosis, procedures and surgeries

- **pat_ct_dp**: count by patient of the number of claims that have only diagnostics and procedures
- **pat_ct_p**: count by patient of the number of claims that have only procedures

## 4.3 Likelihood features

The likelihood features were calculated using smoothing technique to try tweaking the probabilities for the groups that appeared only a few times:

$$\frac{k \cdot \bar{y} + \sum_{i \in G} y_i}{k + |G|},$$

where $G$ is the set of indices for the training examples such that set of chosen raw features has unique values for the examples from $G$, $|G|$ is a size of group $G$, $k = 30$ (defined empirically), $\bar{y}$ is an average of output for all training examples. The effect of adding this $k$ factor is the same as adding $k$ observations with the value equals to the global average.

Each patient had several likelihoods associated with each group, for instances if the group is diagnosis_id, each patient can have many of them. To transform those list into a number we used the maximum value for each likelihood as a feature. To avoid overfitting all likelihoods were calculated using cross validation. The list of features used are presented below:

| table(s) | features subset |
|---|---|
| diagnosis_head | diagnosis_code |
| diagnosis_head | diagnosis_code, primary_physician_role |
| diagnosis_head | diagnosis_code_prefix |
| diagnosis_head | diagnosis_code_prefix, primary_physician_role |
| diagnosis_head | diagnosis_code_prefix, diagnosis_practitioner_specialty_code |
| diagnosis_head | diagnosis_code, diagnosis_practitioner_specialty_code |
| diagnosis_head | diagnosis_code, ethinicity |
| diagnosis_head | diagnosis_practitioner_id |
| diagnosis_feats2 | diagnosis_code, primary_physician_role |
| diagnosis_head2 | diagnosis_code, primary_physician_role, proc_claim_line_diff |
| diagnosis_head2 | diagnosis_code, primary_physician_role, diag_count_claim |
| diagnosis_head2 | diagnosis_code, primary_physician_role, diag_count_claim, proc_claim_line_diff |
| diagnosis_head2 | diag_count_claim, proc_claim_line_diff, primary_physician_role |
| diagnosis_head2 | diag_count_claim, primary_physician_role |

| table(s) | features subset |
|---|---|
| diagnosis_head2 | proc_claim_line_diff, primary_physician_role |
| diagnosis_head2 | diagnosis_code, primary_physician_role |
| procedure_head | procedure_code, procedure_primary_physician_role |
| procedure_head | procedure_code |
| procedure_head | procedure_primary_practitioner_id |
| procedure_head | procedure_attending_practitioner_id |
| procedure_head | procedure_referring_practitioner_id |
| procedure_head | procedure_rendering_practitioner_id |
| procedure_head | procedure_operating_practitioner_id |
| procedure_head2 | procedure_code |
| procedure_head2 | procedure_primary_practitioner_id |
| procedure_head3 | procedure_code, proc_claim_line_diff |
| procedure_head3 | procedure_code, diag_count_claim |
| procedure_head3 | procedure_code, diag_count_claim, proc_claim_line_diff |
| procedure_head3 | diag_count_claim, proc_claim_line_diff |
| procedure_head3 | diag_count_claim, procedure_primary_practitioner_id |
| procedure_head3 | proc_claim_line_diff, procedure_primary_practitioner_id |
| prescription_feats | drug_name |
| prescription_feats | drug_generic_name |
| prescription_feats | drug_name, prescription_practitioner_specialty_code |
| prescription_feats | bb_usc_code |
| prescription_feats | bb_usc_code, prescription_practitioner_specialty_code |
| prescription_feats | prescription_practitioner_id |

| table(s) | features subset |
|---|---|
| prescription_feats | payment_type |
| prescription_feats | drug_strength |
| prescription_feats | drug_id |
| prescription_feats | manufacturer |
| surgical_head | procedure_type_code |
| surgical_head | surgical_code |
| surgical_head | surgical_practitioner_id |
| surgical_head | surgical_primary_physician_role |
| pratictioner_head | all_pract_specialty_code |
| pratictioner_head | all_pract_state |
| pratictioner_head | all_pract_specialty_code, proc_claim_line_diff, diag_count_claim, all_pract_state |
| pratictioner_head | all_practitioner_id |
| pratictioner_head | all_practitioner_id, proc_claim_line_diff |
| diagnosis_pairs_unique | diagnosis_code1, diagnosis_code2 |
| diagnosis_prescription_link | diagnosis_code, drug_generic_name |
| diagnosis_surgical_link | diagnosis_code, surgical_code |
| procedure_surgical_link | procedure_code, surgical_code |
| diagnosis_procedure_surgical_link | diagnosis_code, procedure_code, surgical_code |
| diagnosis_procedure_link | diagnosis_code, procedure_code |
| diagnosis_procedure_link2 | diagnosis_code, procedure_code, plan_type |
| diagnosis_procedure_link4 | diagnosis_code, procedure_code, proc_claim_line_diff, diag_count_claim |

| table(s) | features subset |
|---|---|
| diagnosis_procedure_link4 | diagnosis_code, plan_type, proc_claim_line_diff, diag_count_claim |
| diagnosis_procedure_link4 | diagnosis_code, procedure_code, primary_physician_role, proc_claim_line_diff, diag_count_claim |
| diagnosis_procedure_link4 | diagnosis_code, plan_type, primary_physician_role, proc_claim_line_diff, diag_count_claim |
| patient_claim | claim_type |
| diagnosis_head2 | diagnosis_code, primary_physician_role, pat_claim_type |
| procedure_head3 | procedure_code, pat_claim_type |
| surgical_head3 | surgical_code, surgical_primary_physician_role, pat_claim_type |
| surgical_head3 | surgical_plan_type, surgical_primary_physician_role, pat_claim_type |

## 4.4 Extra features

- **act_start**: earlier date of activity for each patient
- **act_end**: latest date of activity for each patient
- **act_period**: difference between act_end and act_start
- **presc_day_supply**: average of presc_day_supply from prescription_feats by date

- **charge_amount**: sum of charge_amount from diagnosis_procedure_link2 by patient

# 5 Dmitry: feature description

## 5.1 Likelihood features

Likelihood features are calculated based on the subset of raw categorical features. Assuming that $k$ raw categorical features with values $x_1, \ldots, x_k$ have been chosen, for each unique combination of $x_1, \ldots, x_k$ we calculate the average of the target variable (is_screener) in case if the number of training example for the group is greater than 50. If the number of training examples for the unique combination is less than 50 we take the global average of is_screener as a value for the likelihood feature. This average is assigned to the likelihood feature for the dataset entry with given values of the chosen categorical features. To avoid overfitting we applied 3-fold cross validation procedure and assigned likelihoods evaluated on one fold to different folds. As a result of the described procedure we had a set of likelihoods for each patient_id. To combine them and get the only likelihood for each patient_id we used so called aggregation functions. Likelihood features can be splitted in six groups.

- Likelihood features, type 1.

  Calculated without taking into consideration that some paitent_id had repeated values for the chosen subset of categorical features. Generic function used was max of average by each patient_id

**Table 1: likelihood features of type 1**

| table(s) | features subset | generic function |
|---|---|---|
| prescription_head | bb_usc_name | max |
| prescription_head | bb_usc_name, prescription_practitioner_id | max |
| prescription_head | payment_type | max |
| prescription_head | prescription_practitioner_id | max |
| diagnosis_head | diagnosis_code | max |
| diagnosis_head | diagnosis_practitioner_id | max |
| diagnosis_head | diagnosis_practitioner_id, diagnosis_code | max |
| diagnosis_head | diagnosis_code, diagnosis_practitioner_state | max |
| diagnosis_head | diagnosis_code, diagnosis_practitioner_state, diagnosis_practitioner_specialty_code | max |
| diagnosis_head | diagnosis_practitioner_specialty_code | max |
| diagnosis_head, patients | diagnosis_code, patient_age_group | max |
| diagnosis_head, patients | diagnosis_practitioner_id, patient_age_group | max |
| diagnosis_head | diagnosis_practitioner_specialty_code, diagnosis_code | max |
| diagnosis_head | diagnosis_code, primary_physician_role | max |
| diagnosis_head | diagnosis_code, diagnosis_practitioner_cbsa | max |
| diagnosis_head | diagnosis_code, diagnosis_practitioner_specialty_code, primary_physician_role | max |

**Table 1: likelihood features of type 1 (continuation)**

| table(s) | features subset | generic function |
|---|---|---|
| diagnosis_head | diagnosis_code, diagnosis_practitioner_cbsa, primary_physician_role | max |
| procedure_head | procedure_primary_practitioner_id | avg |
| procedure_head | procedure_code | max |
| procedure_head | procedure_code, place_of_service | max |
| procedure_head | procedure_rendering_practitioner_id | max |
| diagnosis_head, procedure_head | procedure_code, diagnosis_code | max |
| diagnosis_head, procedure_head | diagnosis_primary_practitioner_id, procedure_code, diagnosis_code | max |
| diagnosis_head, procedure_head | diagnosis_code, procedure_code, primary_physician_role | max |
| diagnosis_head, procedure_head | diagnosis_code, place_of_service | max |
| diagnosis_head, procedure_head | diagnosis_code, procedure_code, place_of_service | max |
| diagnosis_head, procedure_head | diagnosis_code, plan_type | max |

- Likelihood features, type 2.

  These features are calculated with taking into consideration that some paitent_id had repeated values for the chosen subset of categorical features. The main difference between type 1 and type 2 is that in type 1 the weight of target for patients with identical multiple records increases. Generic function used was max by each patient_id.

**Table 2: likelihood features of type 2**

| table(s) | features subset | generic function |
|---|---|---|
| diagnosis_head | diagnosis_code | max |
| diagnosis_head, patients | diagnosis_code, patient_age_group | max |
| diagnosis_head | diagnosis_code, primary_physician_role | max |
| diagnosis_head, procedure_head | procedure_code, diagnosis_code | max |

- Likelihood features, type 3.

  The same as type 1, but generic function used is second unique max value by each patient_id.

**Table 3: likelihood features of type 3**

| table(s) | features subset |
|---|---|
| diagnosis_head | diagnosis_code |
| diagnosis_head, procedure_head | procedure_code, diagnosis_code |

- Likelihood features, type 4.

  The same as type 1, but generic function used is an average 3 maximum likelihoods by each patient_id.

**Table 4: likelihood features of type 4**

| table(s) | features subset |
|---|---|
| diagnosis_head | diagnosis_code, primary_physician_role |
| diagnosis_head | diagnosis_practitioner_specialty_code, diagnosis_code |
| diagnosis_head | diagnosis_code |
| diagnosis_head, procedure_head | procedure_code, diagnosis_code |

- Likelihood features, type 5.

  The same as type 1, but generic function used is second max value (it could be the same as max value) by each patient_id.

**Table 5: likelihood features of type 5**

| table(s) | features subset |
|---|---|
| diagnosis_head | diagnosis_code |
| diagnosis_head, procedure_head | procedure_code, diagnosis_code |

- Likelihood features, type 6.

  The same as type 1, but table is filtered before likelihood is calculated.

**Table 6: likelihood features of type 6**

| table(s) | features subset | filter column | filter value |
|---|---|---|---|
| diagnosis_head | diagnosis_code | primary_physician_role | ATG |

- Likelihood features, group type.

  Calculated based on each categorical in the chosen subset separately, after the maximum value of likelihoods by each entry is assigned to the group type likelihood. Generic function used was max by each patient_id.

**Table 7: likelihood features of group type**

| table(s) | features subset |
|---|---|
| diagnosis_head | diagnosis_code, diagnosis_practitioner_id |
| diagnosis_head | diagnosis_code, diagnosis_practitioner_id, diagnosis_code_prefix, diagnosis_practitioner_specialty_code |
| diagnosis_head, procedure_head | procedure_code, diagnosis_code |

Table 8 shows the fist step when we calculate likelihoods for the type 1, type 2 and group type. Table 9 shows the final step when the likelihoods by each patient_id are combined. The subset of raw categorical features for the tables contains diagnosis_code and diagnosis_practitioner_id.

**Table 8: first step for likelihood calculation**

| patient | screener | diagnosis | practitioner | 1,3,4,5 | 2 | group |
|---|---|---|---|---|---|---|
| 84654065 | 1 | 079.99 | 12694665 | 1.0 | 1.0 | 1.0 |
| 84654065 | 1 | 079.99 | 12694665 | 1.0 | 1.0 | 1.0 |
| 84654065 | 1 | 158.9 | 12695724 | 0.33 | 0.66 | 0.38 |
| 84654065 | 1 | 183.0 | 12669220 | 1.0 | 1.0 | 1.0 |
| 84884209 | 0 | 158.9 | 12695724 | 0.33 | 0.66 | 0.38 |
| 84884209 | 0 | 158.9 | 12695724 | 0.33 | 0.66 | 0.38 |
| 84884209 | 0 | 158.9 | 12695724 | 0.33 | 0.66 | 0.38 |
| 84884209 | 0 | 158.9 | 12695724 | 0.33 | 0.66 | 0.38 |
| 84884209 | 0 | 244.9 | 12695724 | 0.5 | 0.5 | 0.5 |
| 84945708 | 1 | 079.99 | 12694665 | 1.0 | 1.0 | 1.0 |
| 84945708 | 1 | 158.9 | 12695724 | 0.33 | 0.66 | 0.38 |
| 84945708 | 1 | 244.9 | 12695724 | 0.5 | 0.5 | 0.5 |

**Table 9: second step for likelihood calculation**

| patient | diagnosis | practitioner | 1 | 2 | 3 | 4 | 5 | group |
|---|---|---|---|---|---|---|---|---|
| 84654065 | 079.99 | 12694665 | 1.0 | 1.0 | 0.33 | 1.0 | 1.0 | 1.0 |
| 84884209 | 158.9 | 12695724 | 0.5 | 0.66 | 0.33 | 0.39 | 0.33 | 0.5 |
| 84945708 | 244.9 | 12695724 | 1.0 | 1.0 | 0.5 | 0.61 | 0.5 | 1.0 |

## 5.2 FTRL features

Another set of features have been generated using famous Google algorithm Follow The (Proximally) Regularized Leader (FTRL-Proximal) is the online algorithm presented in the [6, 5].

Assuming that our purpose is to minimise the loss function

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right) \tag{1}$$

where $\hat{y}_i$ is a prediction for the instance $i$, we transform the design matrix $X$ to the sparse binary design matrix such that one column corresponds to one level of categorical feature (due to the huge number of levels we have applied the hashing trick ([4, Chapter 3]) to transform values of categorical features to indices).

The prediction for sample $i$ is constructed as $\hat{y}_i = \sigma\left(w_i \cdot x_i\right)$, where $w_i$ is a weight vector of size $n$ on $i$-th iteration.

The algorithm updates the weights of the sample $i+1$ based on the previous samples $\{1,\ldots,i\}$ by finding

$$w_{i+1} = \arg\min_w \left( \sum_{r=1}^{i} g_r \cdot w + \frac{1}{2} \sum_{r=1}^{i} \tau_r ||w - w_r||_2^2 + \lambda_1 ||w||_1 \right) =$$
$$= \arg\min_w \left( w \cdot \sum_{r=1}^{i} (g_r - \tau_r w_r) + \frac{1}{2} ||w||_2^2 \sum_{r=1}^{i} \tau_r + \lambda_1 ||w||_1 + \text{const} \right)$$

and

$$\sum_{r=1}^{i} \tau_{rj} = \frac{\beta + \sqrt{\sum_{r=1}^{i} (g_{rj})^2}}{\alpha} + \lambda_2,\ j \in \{1,\ldots,N\},$$

where $\lambda_1$, $\lambda_2$ are regularization parameters, $\alpha$, $\beta$ are parameters of the learning rates schedule, $\tau_r = (\tau_{r1},\ldots,\tau_{rN})$ is a vector of learning rates for the step $r$, $g_r = \left(\frac{\partial L}{\partial w_{r1}},\ldots,\frac{\partial L}{\partial w_{rN}}\right)$ is a gradient vector of logarithmic loss $L$ in the form (1) for step $r$.

To generate FTRL features for the training set we applied cross-validation procedure with three folds. The FTRL features for test have been calculated based on the whole training set. The generic function max has been applied to get the only value by each patient_id. FTRL features can be splitted in 4 groups:

- FTRL features, type 1.

  Calculated by each entry (row) of the provided dataset. Generic function is max.

**Table 10: FTRL features of type 1**

| table(s) | features subset |
|---|---|
| diagnosis_head, patients | diagnosis_practitioner_id, diagnosis_code, patient_age_group, patient_state, ethinicity, household_income, education_level |
| diagnosis_head, procedure_head | diagnosis_code, procedure_code |

- FTRL features type 2.

  Calculated by each patient_id. No need for generic function.

**Table 11: FTRL features of type 2**

| table(s) | features subset |
|---|---|
| diagnosis_head, procedure_head | diagnosis_code, procedure_code |

- FTRL features type 3.

  Calculated by two consequent rows in the provided dataset for each patient_id. Generic function is max.

**Table 12: FTRL features of type 3**

| table(s) | features subset |
|---|---|
| diagnosis_head, patients | diagnosis_practitioner_id, diagnosis_code, patient_age_group, patient_state, ethinicity, household_income, education_level |

- FTRL features type 4.

  Calculated by each claim_id. Generic function is max.

**Table 13: FTRL features of type 4**

| table(s) | features subset |
|---|---|
| procedure_head2 | procedure_code procedure_primary_practitioner_id |
| procedure_head2 | procedure_code |
| diagnosis_head2 | diagnosis_practitioner_id, diagnosis_code, primary_physician_role |
| diagnosis_head2 | diagnosis_code, primary_physician_role |
| diagnosis_head2 | diganosis_code |

### 5.3 Count features

We have also created the following list of count features:

- patient _id_prescription_visit_count
- patient_id_diff_prescription_practitioner_count
- patient_id_diagnosis_visit_count
- patient_id_diff_diagnosis_practitioner_count
- patient_id_procedure_visit_count
- patient_id_diff_procedure_practitioner_count
- patient_id_surgical_visit_count
- patient_id_diff_surgical_practitioner_count
- patient_id_diff_diagnosis_code_count
- patient_id_diff_diagnosis_code_prefix_count
- procedure_charge_amount_max
- procedure_charge_amount_mean
- procedure_charge_amount_sum
- patient_id_diff_drug_count
- patient_id_diff_rx_number_count
- patient_id_diff_prescription_specialty_count

## 6 Models

The final model was homogeneous ensembling of 12 XGBoost [1] models with all features.

# 7 How to generate the predictions

## 7.1 Gilberto: file list and training

These scripts build 241 features for posterior use in the training process.

1. Make sure all input files are in the folder `data/input/`
2. Run the script `run.all.r` in the folder `genentech-R-giba` using R language
3. Uutput files containing all features are placed in `data/team/giba/`

**Dependencies:**
**R:** data.table, pROC, bit64
**Vowpal Wabbit** [2]

## 7.2 Kohei: file list and training

| Task | File name |
|---|---|
| **Build features** | gen_feat.py |
| **Train model** | kohei_v52.py |

Note that Kohei's model requires huge amount of computational resources. It requires 160 GB RAM and expected runtime is about 30 hours with 36 cores of CPUs. Amazon EC2 m4.10xlarge instance is suitable for computing this.

1. Make sure all input files are in the folder `data/input/`
2. Open command line and change path to `genentech-py-kohei`
3. Run the following list of commands:
   - `python gen_feat.py`
   - `python gen_feat2.py`
   - `python kohei_v52.py --validate --fold1`
   - `python kohei_v52.py --validate --fold2`
   - `python kohei_v52.py --validate --fold3`

- `python kohei_v52.py --test`
- `python kohei_v52.py --merge`

**Dependencies:**
**Anaconda 2.4.1:** Python 2.7.11, bloscpack 0.10.0, blosc 1.2.5, numpy 1.10.1, scipy 0.16.0, sklearn 0.16.0, xgboost 0.4 For reference, Dockerfile for setting up the environment is also available on the `genentech-py-kohei/base` directory.

## 7.3 Dmitry: file list and training

| Task | File name |
|---|---|
| **Copy files from Amazon S3 to Amazon Redshift** | copy_from_s3_to_redshift.py |
| **Remove excluded patients** | generate_train_test_without_excluded_patients.py |
| **Generate CV table** | generate_cv_table.py |
| **Clean diagnosis description** | clean_diagnosis_code_table.py |
| **Generate additional tables** | generate_feature_tables.py |
| **Build features** | build_features.py |
| **Helper functions** | utils.py |
| **Build FTRL features** | ftrl.py, ftrl2.py, ftrl3.py, ftrl4.py |
| **Credentials for Amazon** | data/input/credentials.csv |

1. Upload all data files to Amazon S3 bucket
2. Fill the file `credentials.csv` in the folder `data/input/` with your credentials for Amazon S3 and Amazon Redshift
3. Open command line and change path to `genentech-py-dmitry`
4. Run the script `run.sh`
5. The features will be saved in the `data/team/dmitry/train_feats_dmitry_fold_x.csv` files

**Dependencies:**
The model can be run on Linux Ubuntu 12.04 or Mac OS.
**Python:** psycopg2, pandas, numpy, sklearn [3], xgboost, ml_metrics, scipy, os, glob, itertools, nltk, random, sys, getopt, csv, re

**Other:** pypy, HDF5 (https://www.hdfgroup.org/HDF5/)
The Python version used 2.7.3.

## 7.4 Lucas: file list and training

| Task | File name |
|---|---|
| **Utility functions** | base_util.py |
| **Builds features and loads features from other teammates** | s00_data_build.py |
| **Run the xgb model to get the final predictions** | s01_xgb_02.py |

1. Run all code from other team mates
2. Open command line and change path to `genentech-py-lucas`
3. Run s00_data_build.py with command `python3 s00_data_build.py`
4. Run s01_xgb_02.py with command `python3 s01_xgb_02.py`

**Dependencies:**
**Python:** numpy, pandas, scipy, scikit-learn [3], ml_metrics, cython, psycopg2, h5py, tables, xgboost (installed from https://github.com/dmlc/xgboost.git)
The Python version used 3.0.

# References


[1] https://github.com/dmlc/xgboost

[2] https://github.com/JohnLangford/vowpal_wabbit/wiki

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay *Scikit-learn: Machine Learning in Python.* 1991, Journal of Machine Learning Research, 12, pp. 2825-2830.

[4] A. Rajaraman and J. D. Ullman *Mining of massive datasets.* 2011, Cambridge University Press.

[5] H.B. McMahan, G. Hold, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos and J. Kubica *Ad Click Prediction: a View from the Trenches.* August, 2013, KDD, Chicago, Illinois, USA.

[6] H. B. McMahan *Follow-the-Regularized-Leader and mirror descent: Equivalence theorems and L1 regularization.* 2011, 14th International Conference on Artificial Intelligence and Statistics (AISTATS), 15.

[7] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html