2017-18 winter Wheel Spin

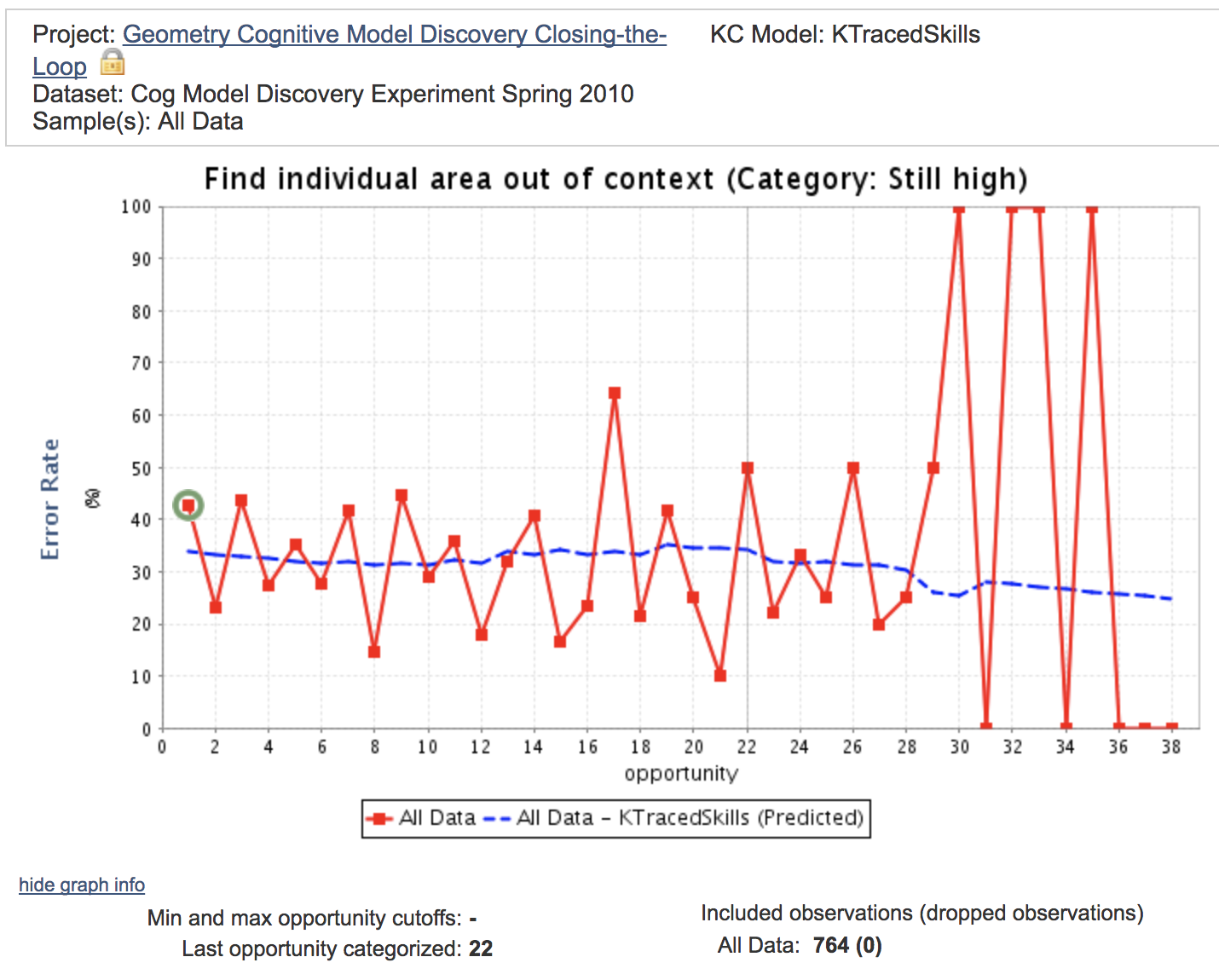
Winter Goal : write at least 2 papers. One upto the late of Jan. the other maybe after that.

Things to do.

* Summarizing what I did two months ago.
* Making hypothesis
* Making new dataset for testing hypothesis.
* 1) feature engineering on kc/ pd variable.
* 2) How fast? On 7th Opportunity or less. -how to test?

12/19

* Download ‘Cog Model Discovery Experiment Spring 2010’ dataset from datashop.(for KTracedSkill)
* Check the error rate and sort out the skills with good learning curve.
* Check each skill : number of opp, students, rate….etc.

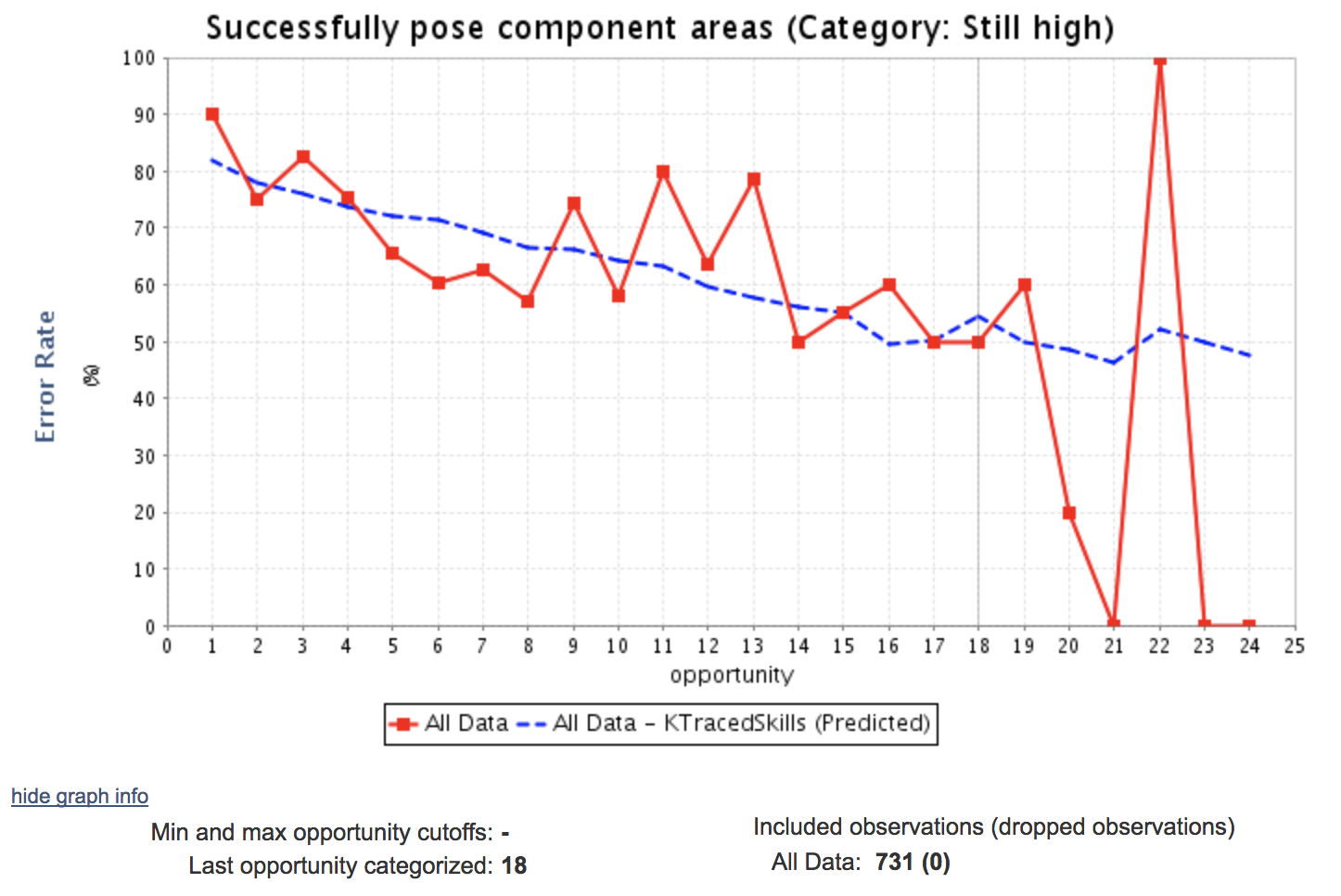


12/26

I should select several skills. Because I need to compare IDPT/ IDKC. There should be several skills…

-> ‘delete low and flat skills.’ file.

How to deal with successfully pose component areas…..it spreads 11 combined skills. -> deleted.



1/3/2018

**I wonder the way to deal with combined skills. Noboru suggested 1) separate those skills & opportunities with pearl script. 2) just regard the combined skill as the independent one. <-he recommended the second one.**

**Noboru also suggested topics**

1. **Compare combined skills effects b/w two ways mentioned above.**
2. **Comparing IDPT/ IDKC predictors on wheel spin.**

* **Write ‘introduction’ for the paper. - problem/ significance/ solution/ related research**
* **Predictive model**
* **Make a file ‘delete low and flat KCs’.**
* **Do some experiment.**

1. **See IDPT/ IDKC predictor on W/S in good skills.**
2. **See “ in still high skills.**
3. **See “ in no learning skills.**

**+**

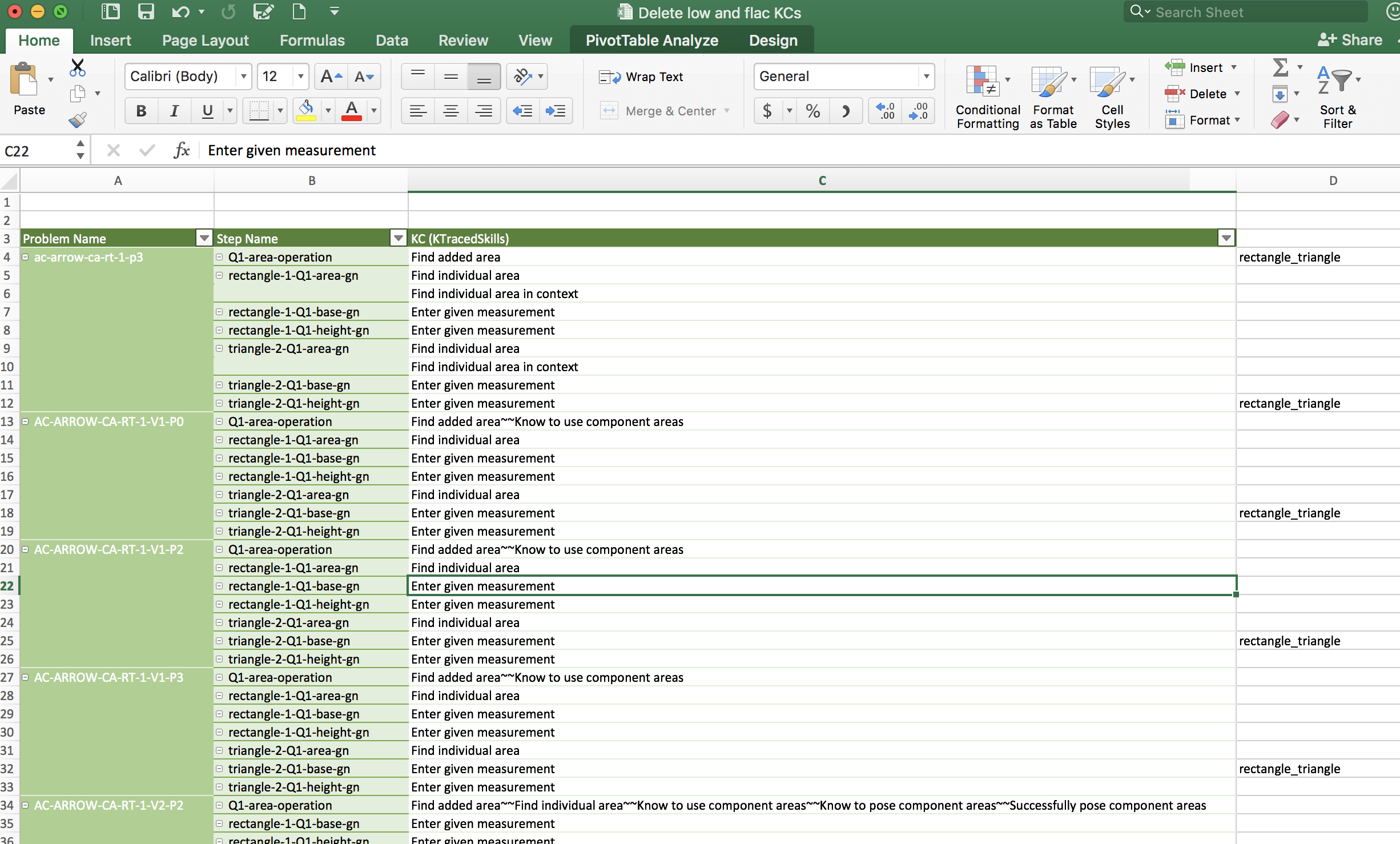
1. **Separating combined skill**
2. **Regard as one skill -> matsuda’s choice**

**1/15/2018**

**The question of the proposal is ‘is the cognitive tutor KC the best KC for detecting wheel-spin’. This paper doubt that the existing KC are too fine-grained. Maybe we need to use problem-type KC or curriculum based KC which comprehends several steps(KC).**

**-data preprocessing**

1. **Make problem type per each problem name in the raw data, based on step name.**

****

**2) Create IDKC/IDPT covariate. (calculate binary First Attempt/Max Opportunity per IDKC/IDPT)\_\_\_\_\_you need to re-assign the Opportunity for IDKC/ IDPT. This is because, first, max opportunity for the combined skill are shown as 0. For these combined skills, I needed to sort IDKC/ step end time/ and get opportunity for those skills. Matsuda said that regarding those skills as independent ones.**

* **Opp for IDPT : if(idpt1=idpt2, opp+1,1)**

**3) calculate the student/ PT/ KC/ IDKC/ IDPT**

**4) Dependent variable? About this, Matsuda said that maybe I could compare WS based on IDKC and that of IDPT...and try several methods for**

**& Matsuda really wants to analyze common answer in transaction data...maybe use SMART?**

**1.16**

**WS**

1. **=CONCATENATE(B3,C3,D3,E3,F3,G3,H3,I3,J3,K3,L3,M3,N3,O3,P3,Q3,R3,S3,T3,U3,V3,W3,X3,Y3,Z3,AA3,AB3,AC3,AD3,AE3,AF3,AG3,AH3,AI3,AJ3,AK3,AL3,AM3,AN3,AO3,AP3,AQ3,AR3,AS3,AT3,AU3,AV3,AW3,AX3,AY3,AZ3,BA3,BB3,BC3,BD3,BE3,BF3,BG3,BH3,BI3,BJ3,BK3,BL3,BM3,BN3,BO3,BP3,BQ3,BR3,BS3,BT3,BU3,BV3,BW3,BX3,BY3,BZ3,CA3,CB3,CC3,CD3,CE3,CF3,CG3,CH3,CI3,CJ3,CK3,CL3,CM3,CN3,CO3,CP3,CQ3,CR3,CS3,CT3,CU3,CV3,CW3,CX3,CY3,CZ3,DA3,DB3,DC3,DD3,DE3,DF3,DG3,DH3,DI3,DJ3)**

|  |  |  |
| --- | --- | --- |
|  | **IDKC WS** | **IDPT WS** |
| **5 covariate** | **79.93%**  **0.387 +/- 0.000** | **92.27% 0.242 +/- 0.000** |
| **ID covariate** | **76.11%**  **0.424 +/- 0.000** | **90.29% 0.287 +/- 0.000** |
| **KC covariate** | **78.61% 0.392 +/- 0.000** | **90.09% 0.298 +/- 0.000** |
| **PT covariate** | **76.11%**  **0.425 +/- 0.000** | **90.09% 0.287 +/- 0.000** |
| **IDKC covariate** | **80.68% 0.392 +/- 0.000** | **90.09% 0.293 +/- 0.000** |
| **IDPT covariate** | **76.32% 0.424 +/- 0.000** | **92.32%**  **0.242 +/- 0.000** |

**IDKC WS with 5 covariate variable**

**PerformanceVector:  
accuracy: 79.93%  
ConfusionMatrix:  
True: W M  
W: 1867 780  
M: 4935 20891  
root\_mean\_squared\_error: 0.387 +/- 0.000**

**Z**

**ID**

**PerformanceVector:  
accuracy: 76.11%  
ConfusionMatrix:  
True: W M  
W: 0 0  
M: 6802 21671  
root\_mean\_squared\_error: 0.424 +/- 0.000**

**KC**

**PerformanceVector:  
accuracy: 78.61%  
ConfusionMatrix:  
True: W M  
W: 1424 712  
M: 5378 20959  
root\_mean\_squared\_error: 0.392 +/- 0.000**

**PT**

**PerformanceVector:  
accuracy: 76.11%  
ConfusionMatrix:  
True: W M  
W: 0 0  
M: 6802 21671  
root\_mean\_squared\_error: 0.425 +/- 0.000**

**IDKC**

**PerformanceVector:  
accuracy: 80.68%  
ConfusionMatrix:  
True: W M  
W: 1621 320  
M: 5181 21351  
root\_mean\_squared\_error: 0.392 +/- 0.000**

**IDPT**

**PerformanceVector:  
accuracy: 76.32%  
ConfusionMatrix:  
True: W M  
W: 61 2  
M: 6741 21669  
root\_mean\_squared\_error: 0.424 +/- 0.000**

**IDPT WS with 5 covariate variables**

**PerformanceVector:  
accuracy: 92.27%  
ConfusionMatrix:  
True: M W  
M: 25156 1704  
W: 496 1117  
root\_mean\_squared\_error: 0.242 +/- 0.000**

**ID**

**PerformanceVector:  
accuracy: 90.29%  
ConfusionMatrix:  
True: M W  
M: 25619 2733  
W: 33 88  
root\_mean\_squared\_error: 0.287 +/- 0.000**

**KC**

**PerformanceVector:  
accuracy: 90.09%  
ConfusionMatrix:  
True: M W  
M: 25652 2821  
W: 0 0  
root\_mean\_squared\_error: 0.298 +/- 0.000**

**PT**

**PerformanceVector:  
accuracy: 90.09%  
ConfusionMatrix:  
True: M W  
M: 25652 2821  
W: 0 0  
root\_mean\_squared\_error: 0.287 +/- 0.000**

**IDKC**

**PerformanceVector:  
accuracy: 90.09%  
ConfusionMatrix:  
True: M W  
M: 25652 2821  
W: 0 0  
root\_mean\_squared\_error: 0.293 +/- 0.000**

**IDPT**

**PerformanceVector:  
accuracy: 92.32%  
ConfusionMatrix:  
True: M W  
M: 25167 1702  
W: 485 1119  
root\_mean\_squared\_error: 0.242 +/- 0.000**

**Call: glm(formula = dataKC$`IDKC WS` ~ ., family = binomial, data = dataKC)**

**Coefficients:**

**(Intercept) `ID covariate` `KC covariate`**

**6.7447 -2.6731 -6.3525**

**`PT covariate` `IDKC covariate` `IDPT covariate`**

**-0.4115 -1.1647 0.4239**

**Degrees of Freedom: 28472 Total (i.e. Null); 28467 Residual**

**Null Deviance: 31310**

**Residual Deviance: 26250 AIC: 26260**

**Coefficients:**

**(Intercept) dataKC$`ID covariate`**

**1.216 -3.004**

**Coefficients:**

**(Intercept) `KC covariate`**

**4.578 -7.423**

**Coefficients:**

**(Intercept) `PT covariate`**

**1.121 -2.877**

**Coefficients:**

**(Intercept) `IDKC covariate`**

**1.804 -3.885**

**Coefficients:**

**(Intercept) `IDPT covariate`**

**0.06668 -1.55632**

**-find how to get variance in logistic regression. (anova)**

**-balanced data**

**-smaller opportunity dataset.**

**-see student answer**

**-KLI paper or how people defined learning.**

**1.24**

**Make ‘delete blank’ excel data file - use all KC.**

**Adopt “I” (Indeterminate) concept : students who didn’t do 10 (or given) opportunities.**

**Can we predict W/S on 5 OPP or 6 OPP?**

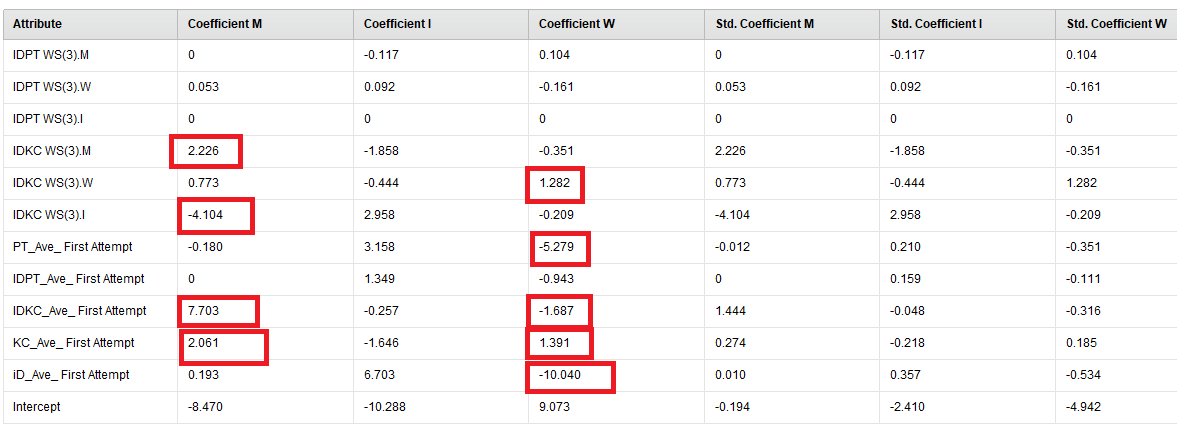
**-> for this, it might be better to use IDPT WS.**

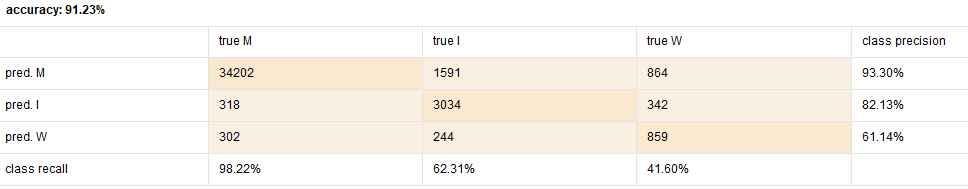
**=if(dm3<=1,”M”,if(do3<3,”I”,”W”))**

* **Prior number of correct responses by the student on this skill**
* **Response times on this skill. We first transform response times for each item into a Z score for that item (to account for some problems taking longer than others). We then took the geometric mean, γ \* prior\_average + (1- γ) \* new\_observation, with γ = 0.7. The geometric mean is a method of summarizing sequential data, but provides lesser weight to older observations, as prior observations are decayed by γ at each time step.**
* **How many times the student reached a bottom out hint on this skill.**
* **How often the student was rapidly guessing, computed across all skills, defined as submitting responses less than 2 seconds apart on successive items. We took the geometric mean in the same manner as for response time.**
* **How often the student gave a rapid response, computed across all skills, defined as responding in a time frame that suggests a reading rate of over 400 words per minute. We took the geometric mean of this feature.**
* **How often the student reached a bottom out hint on 3 consecutive problems, computed across all skills; a 1 indicates the student requested the answer on 3 consecutive problems. We took the geometric mean of this feature.**
* **The name of the current skill**
* **Matsuda want me to teach SimStudent again.**
* **Create flexible WS detector. Only use 3 (4,5,6,...) response sequence.**
* **Automatically calculate IDPT/ IDKC covariate as student practice the problem.**
* **Dependent variable can be the final mastery Opportunity.**

**1/31**

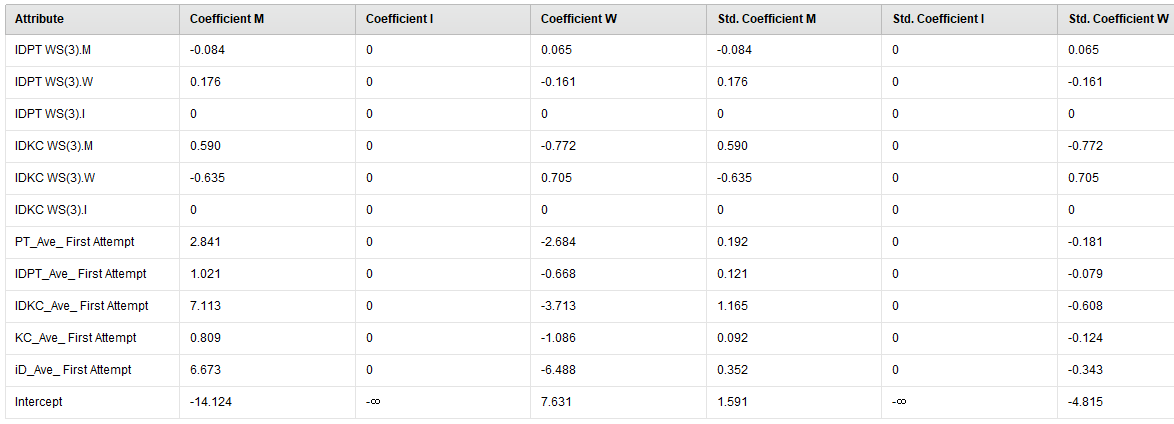
**IDKC WS(10) predict, using (3)**

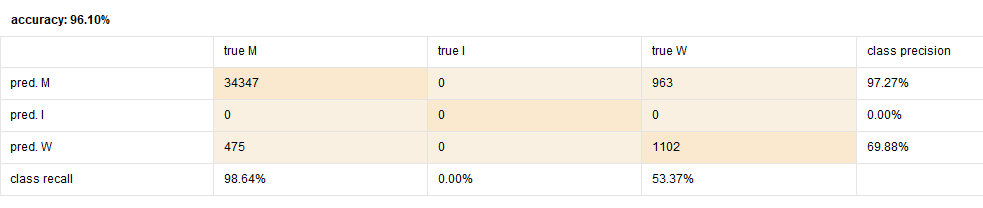
****

****

**root\_mean\_squared\_error: 0.265 +/- 0.000**

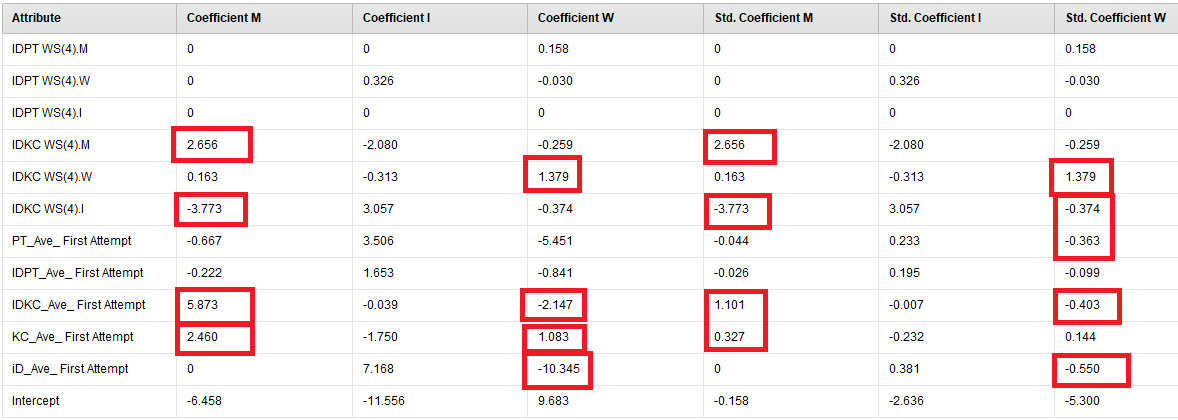
**Without I**

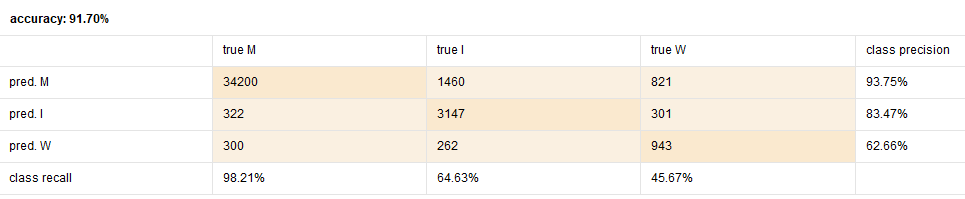
****

****

**root\_mean\_squared\_error: 0.162 +/- 0.000**

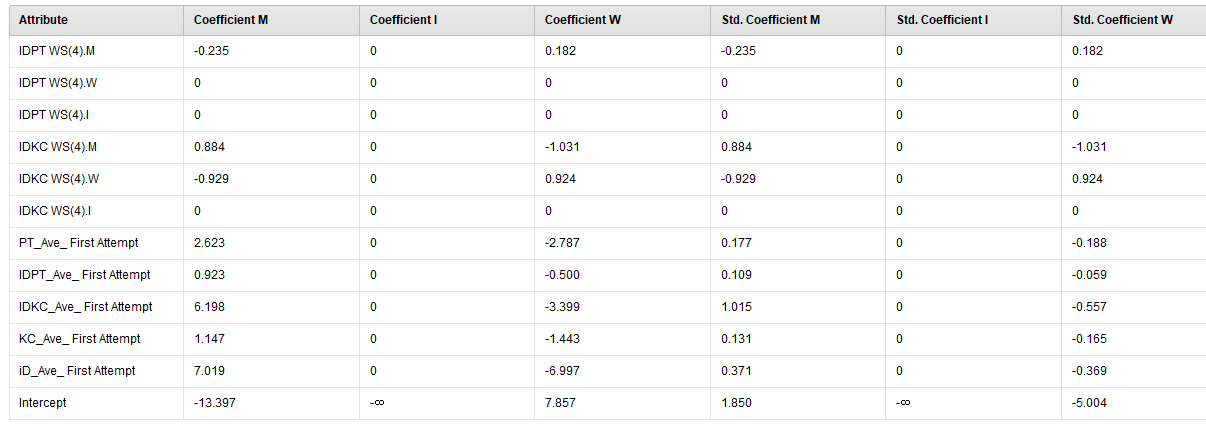
**IDKC WS(10) predict, using (4)**

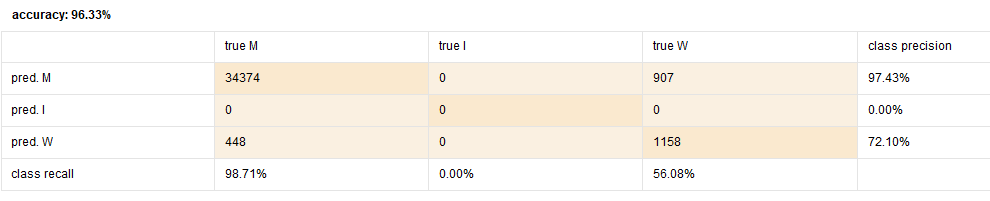
****

****

**root\_mean\_squared\_error: 0.258 +/- 0.000**

**Without I**

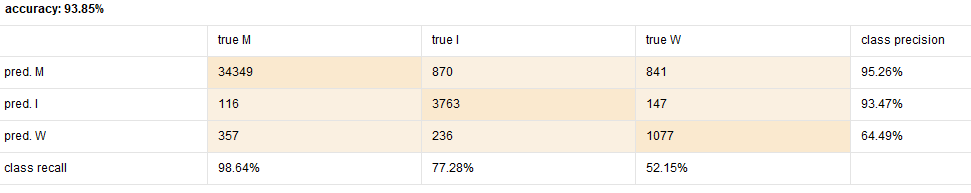
****

****

**root\_mean\_squared\_error: 0.159 +/- 0.000**

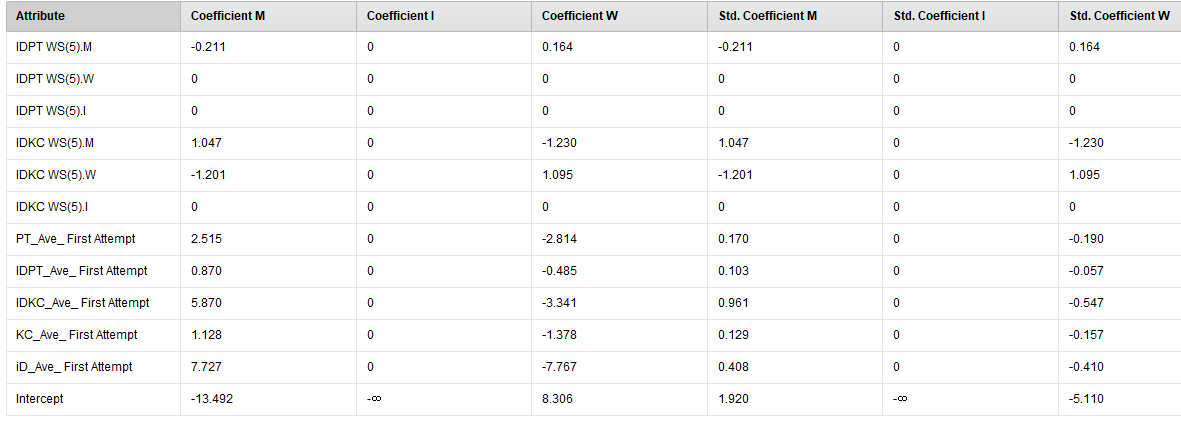
**IDKC WS(10) predict, using (5)**

****

****

**root\_mean\_squared\_error: 0.224 +/- 0.000**

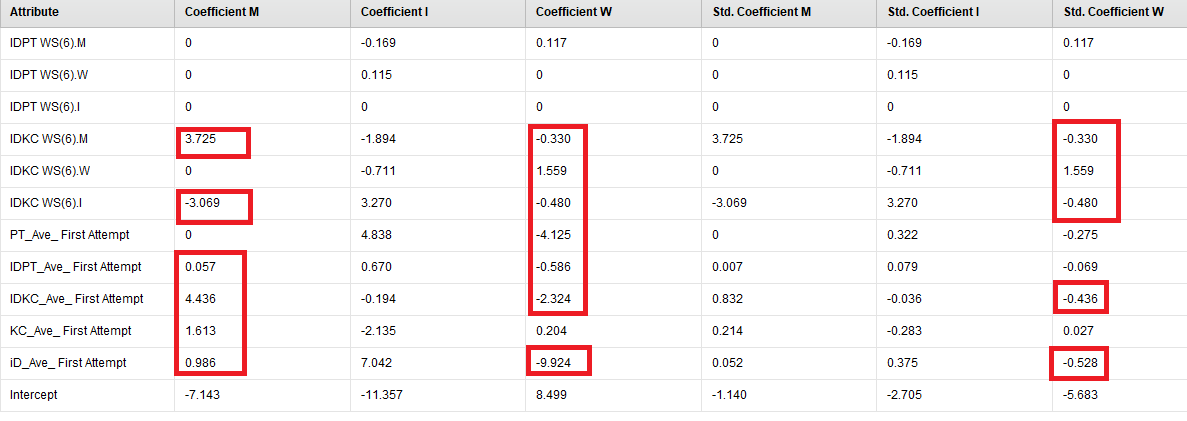
**Without I**

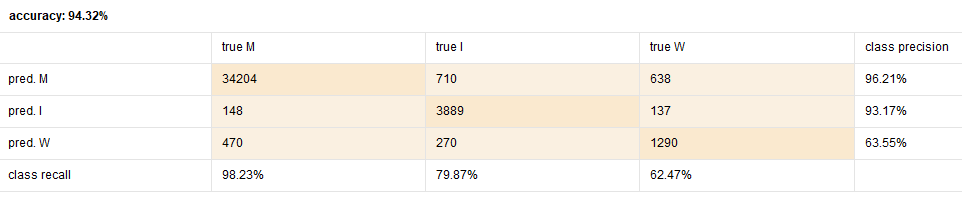
****

****

**root\_mean\_squared\_error: 0.154 +/- 0.000**

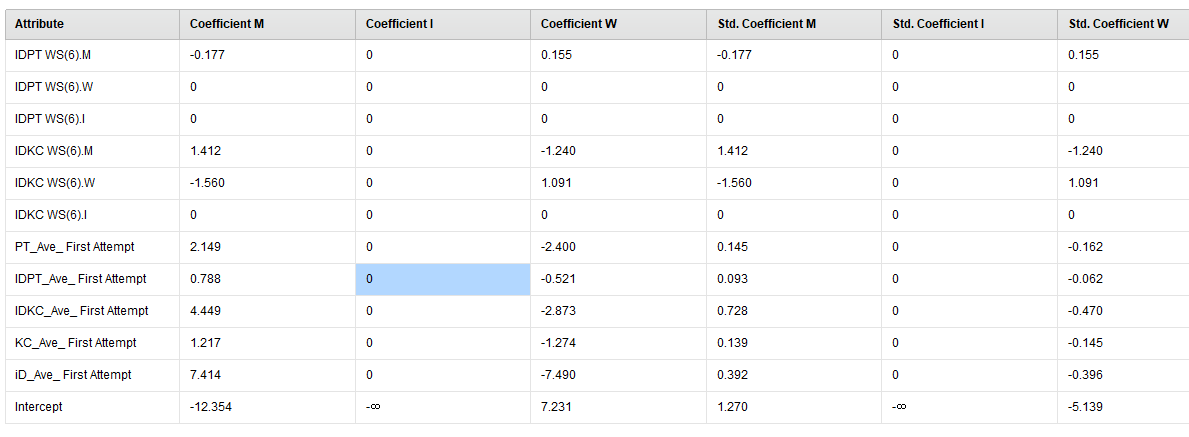
**IDKC WS(10) predict, using (6)**

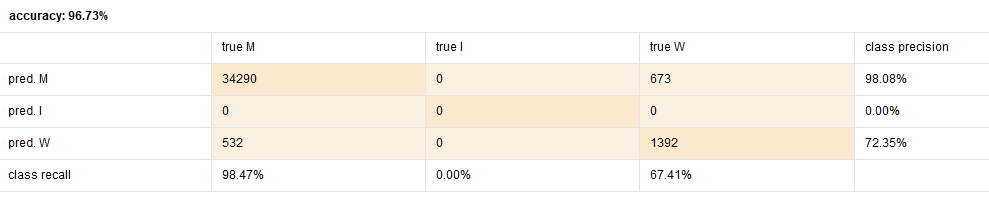
****

****

**root\_mean\_squared\_error: 0.208 +/- 0.000**

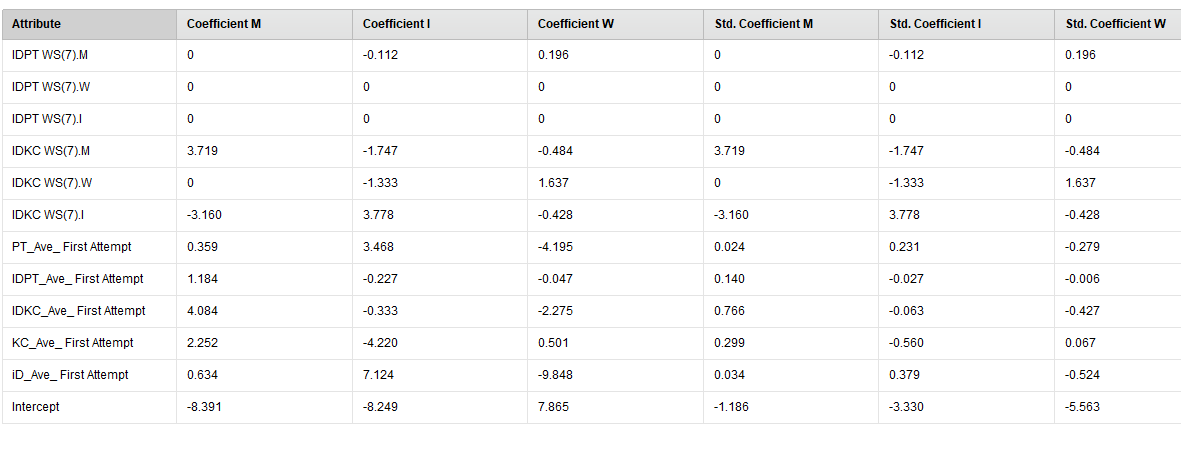
**Without I**

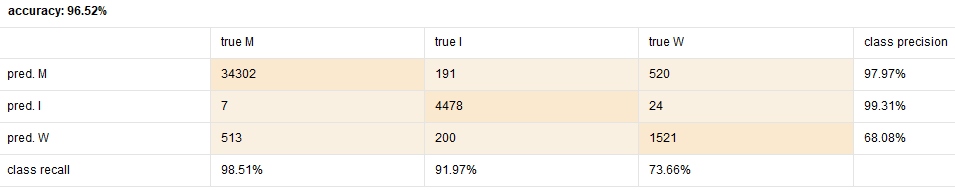
****

****

**root\_mean\_squared\_error: 0.145 +/- 0.000**

**IDKC WS(10) predict, using (7)**

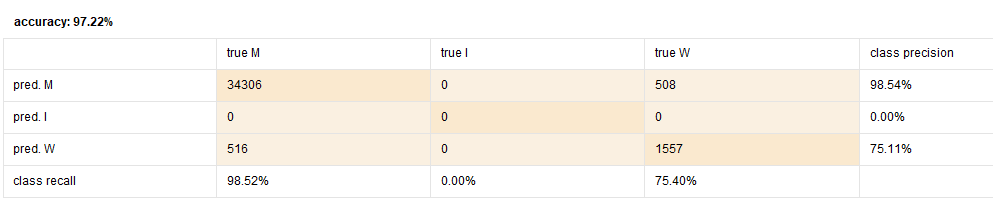
****

****

**root\_mean\_squared\_error: 0.164 +/- 0.000**

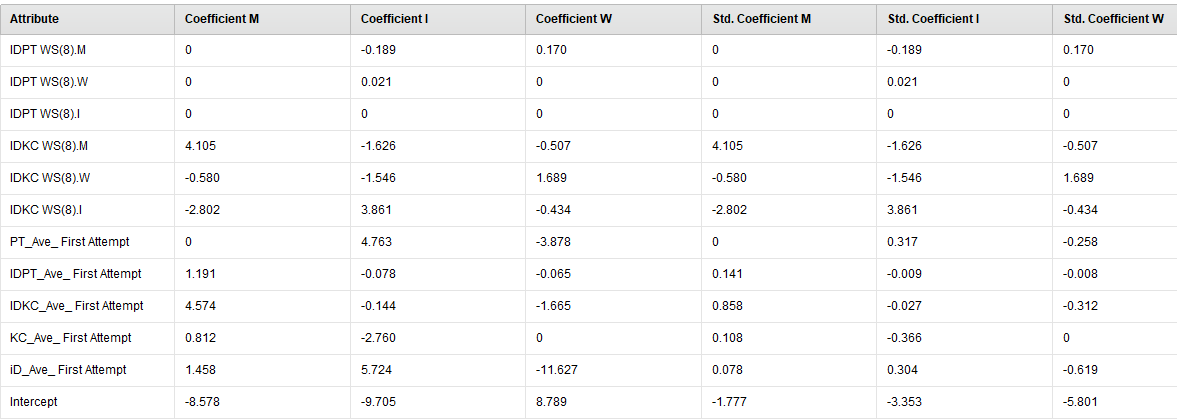
**Without I**

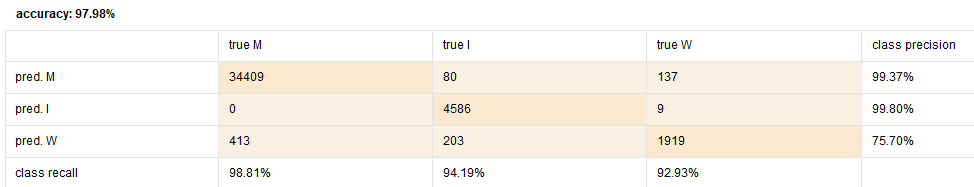
****

****

**root\_mean\_squared\_error: 0.135 +/- 0.000**

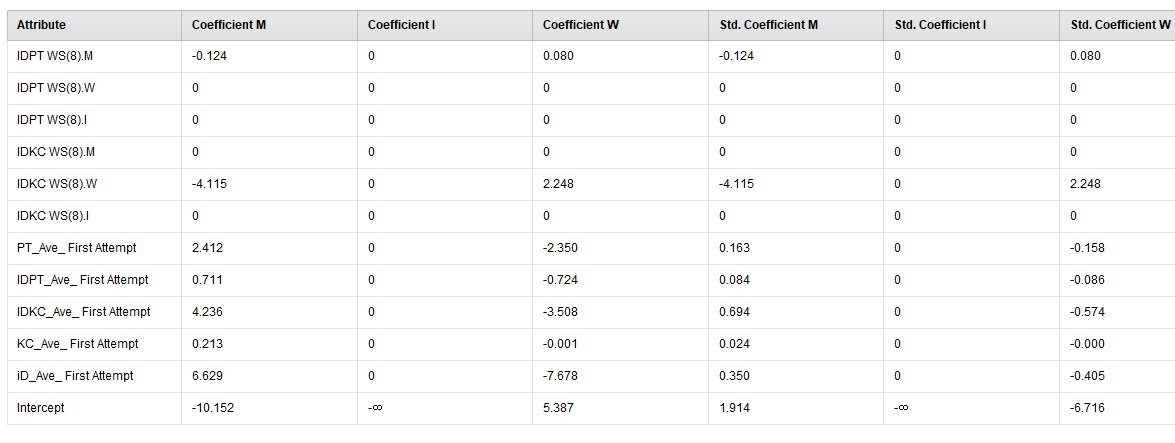
**IDKC WS(10) predict, using (8)**

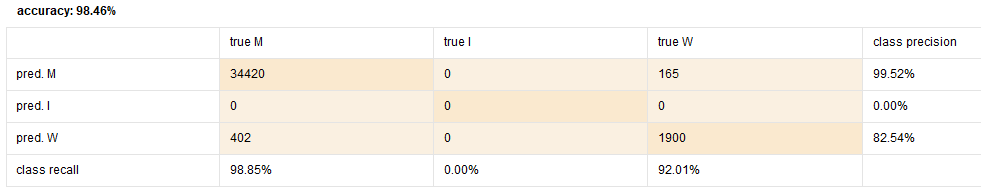
****

****

**root\_mean\_squared\_error: 0.130 +/- 0.000**

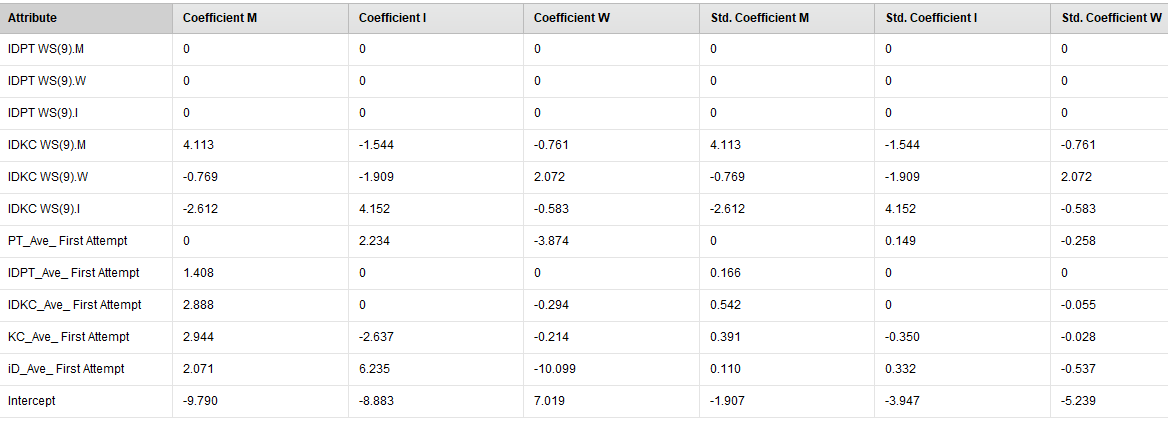
**Without I**

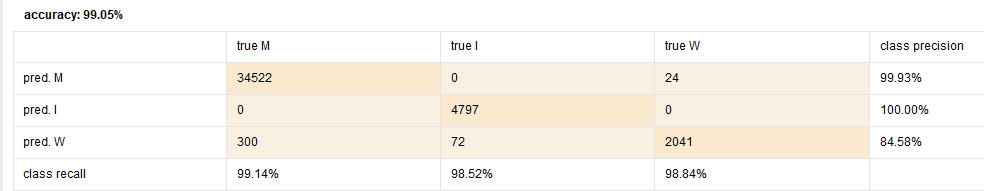
****

****

**root\_mean\_squared\_error: 0.103 +/- 0.000**

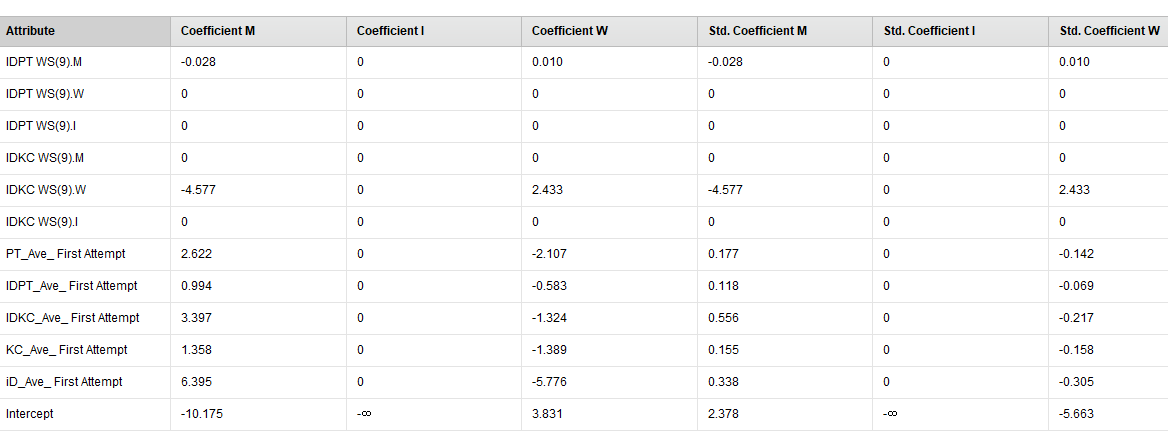
**IDKC WS(10) predict, using (9)**

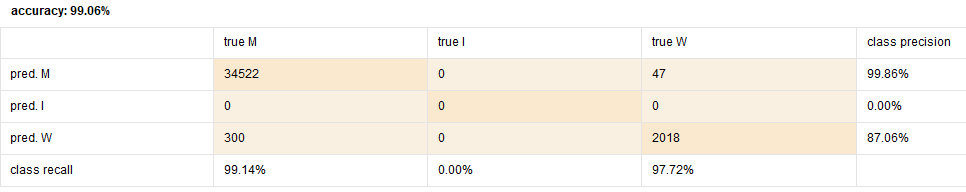
****

****

**root\_mean\_squared\_error: 0.091 +/- 0.000**

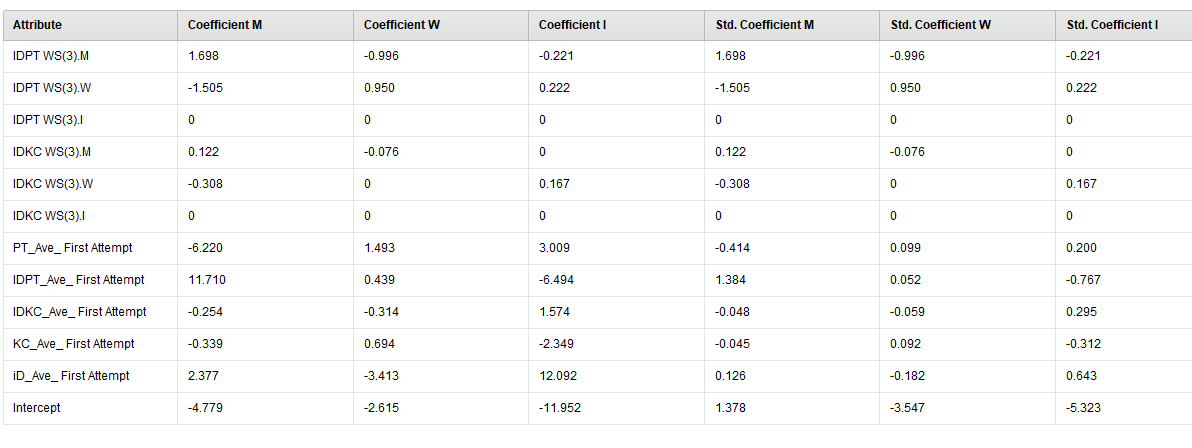
**Without I**

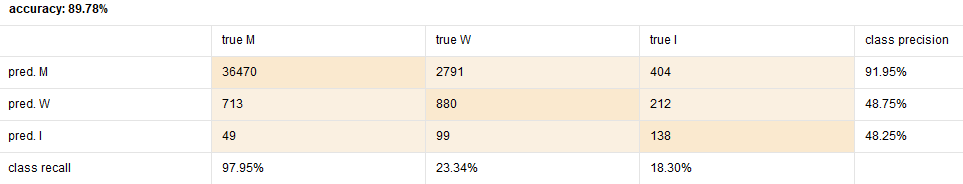
****

****

**root\_mean\_squared\_error: 0.084 +/- 0.000**

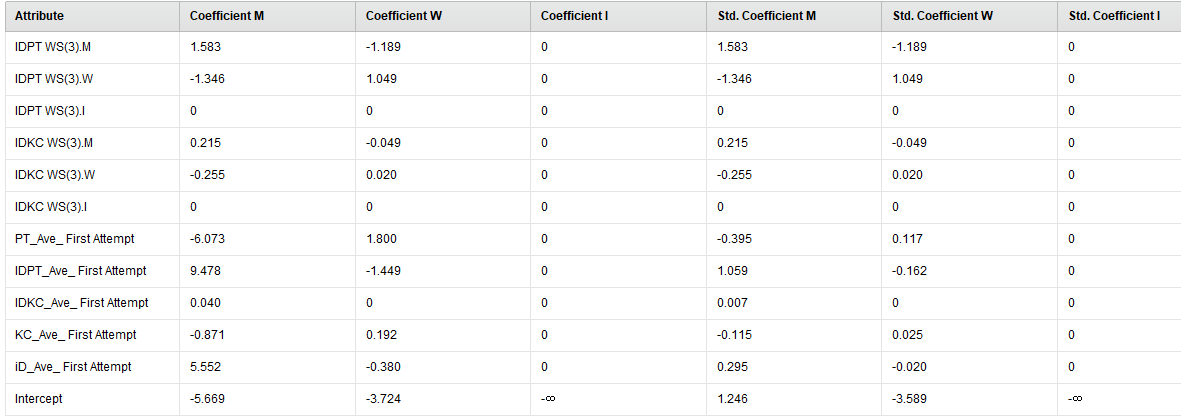
* **Predicting IDPT WS(10) with (3)**

****

****

**root\_mean\_squared\_error: 0.274 +/- 0.000**

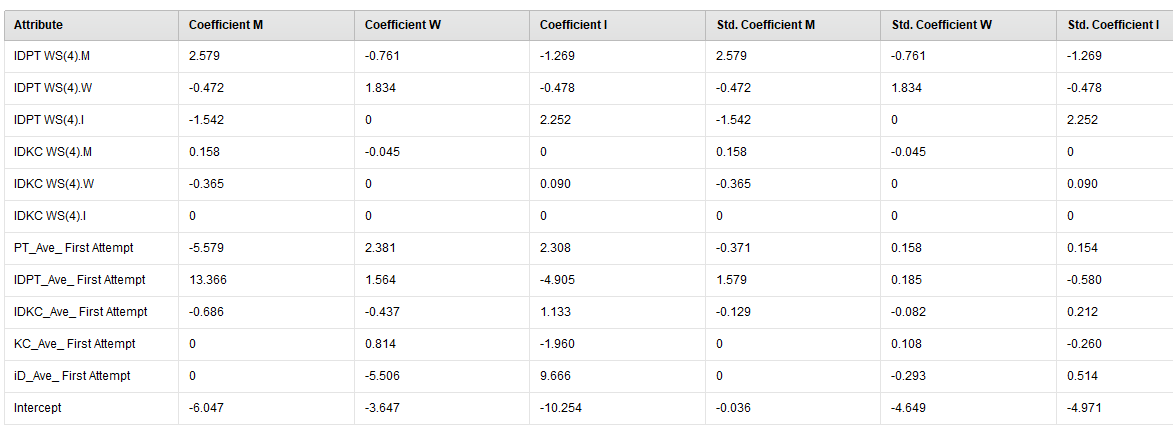
**Without I**

****

****

**root\_mean\_squared\_error: 0.242 +/- 0.000**

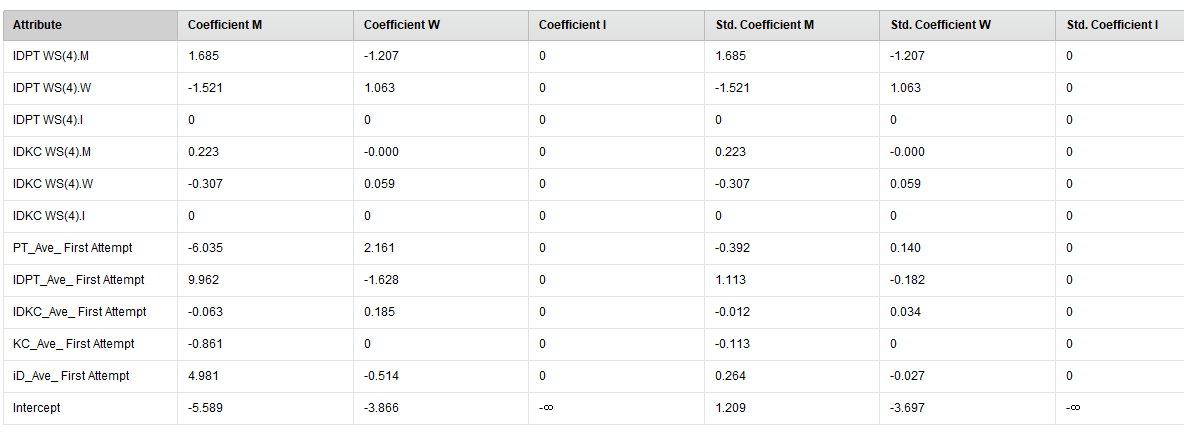
* **Predicting IDPT WS(10) with (4)**

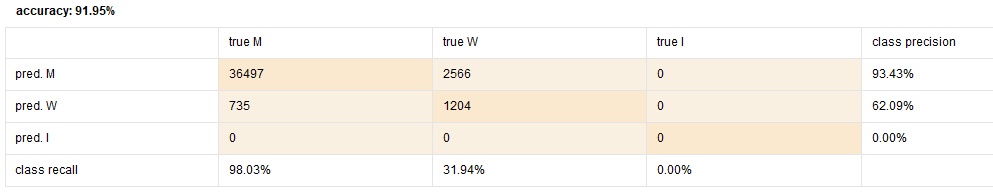
****

****

**root\_mean\_squared\_error: 0.265 +/- 0.000**

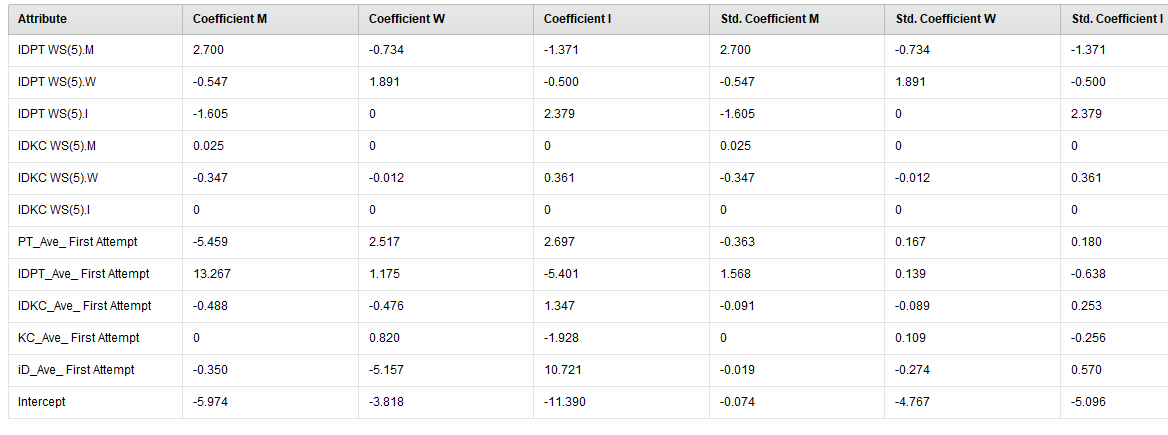
**Without I**

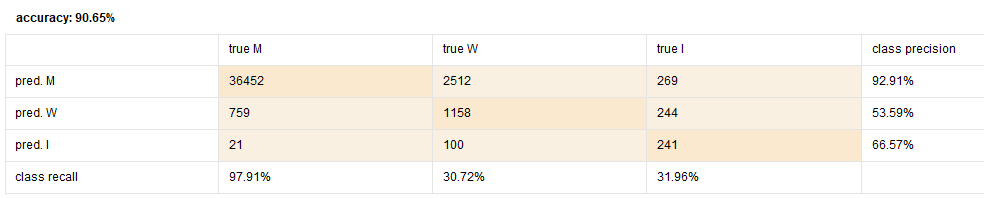
****

****

**root\_mean\_squared\_error: 0.236 +/- 0.000**

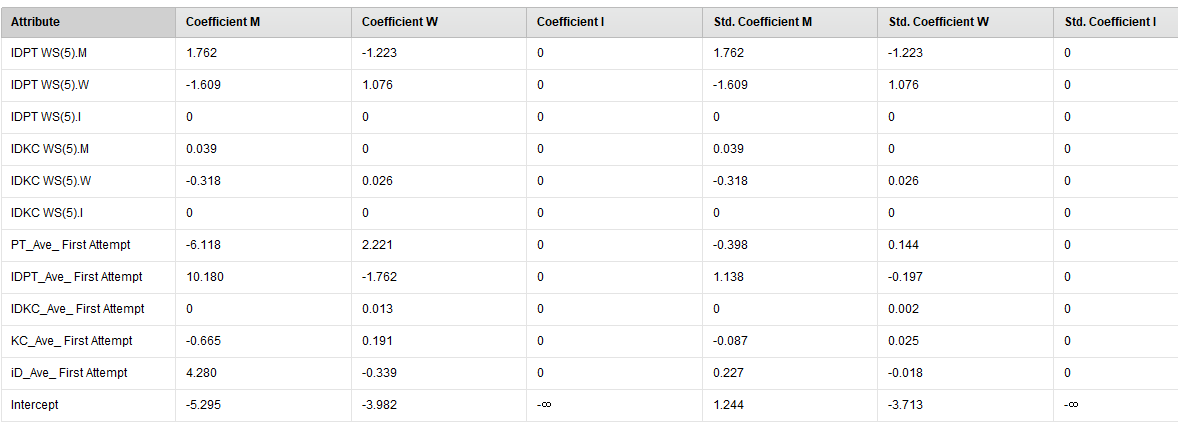
* **Predicting IDPT WS(10) with (5)**

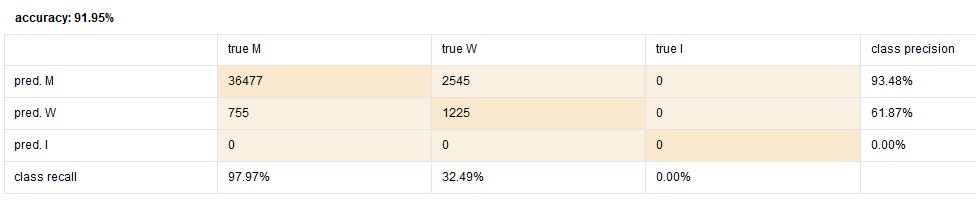
****

****

**root\_mean\_squared\_error: 0.262 +/- 0.000**

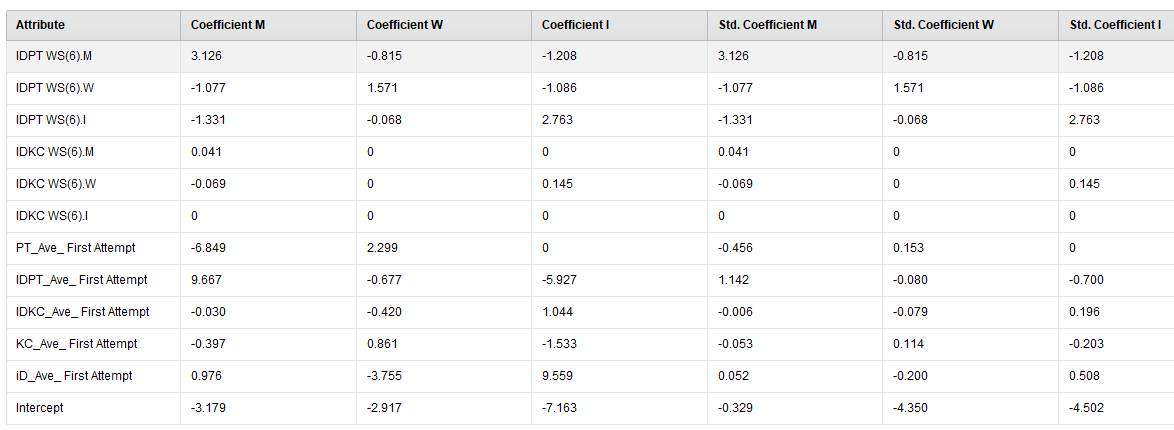
**Without I**

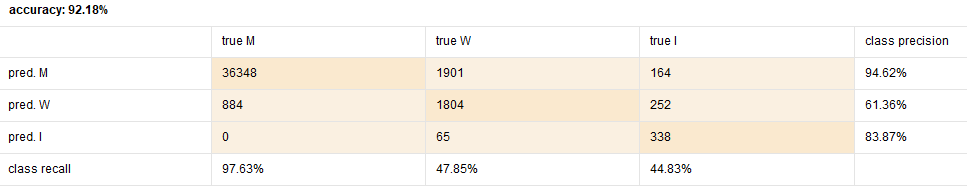
****

****

**root\_mean\_squared\_error: 0.234 +/- 0.000**

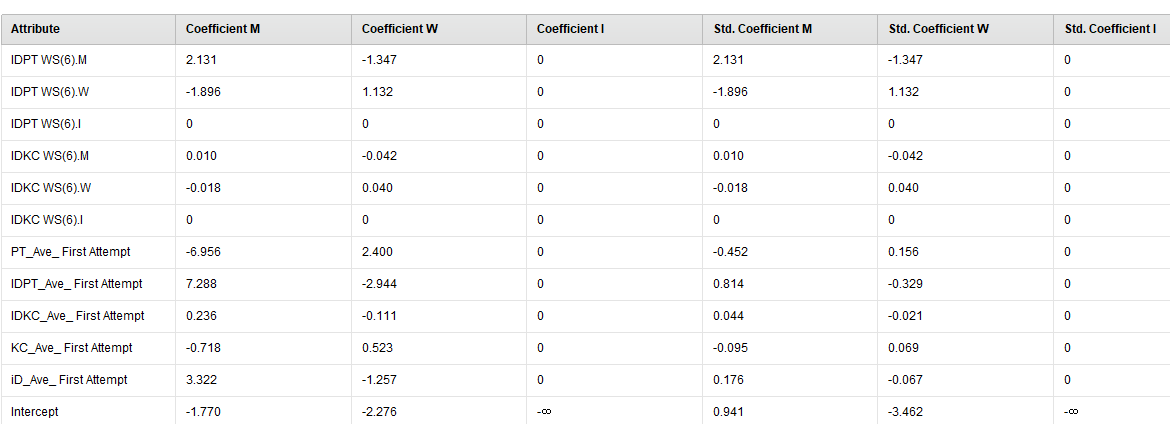
* **Predicting IDPT WS(10) with (6)**

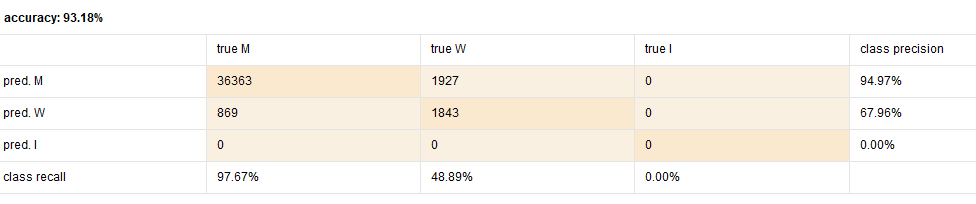
****

****

**root\_mean\_squared\_error: 0.239 +/- 0.000**

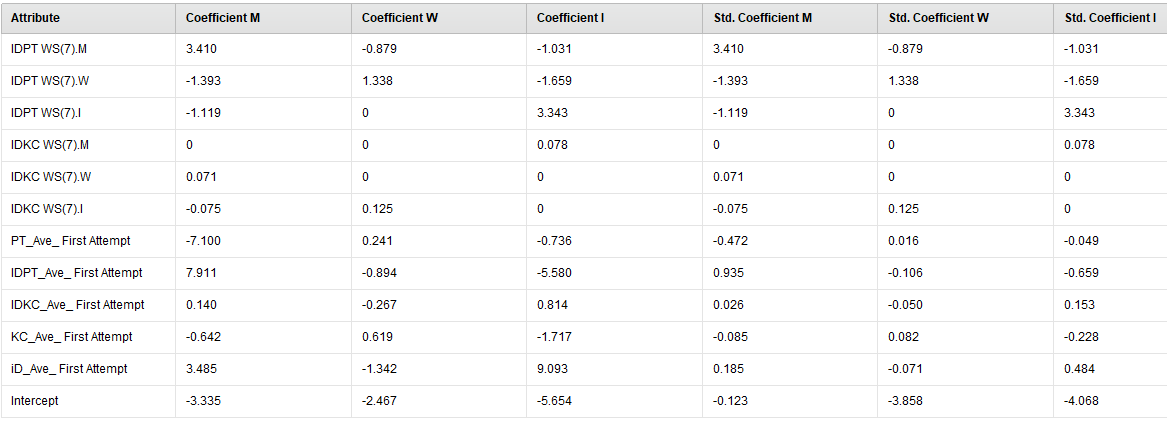
**Without I**

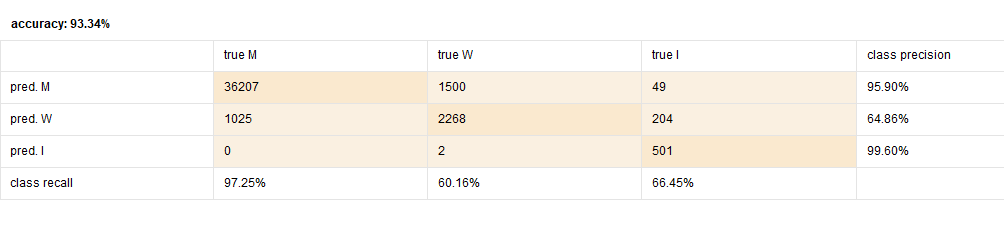
****

****

**root\_mean\_squared\_error: 0.214 +/- 0.000**

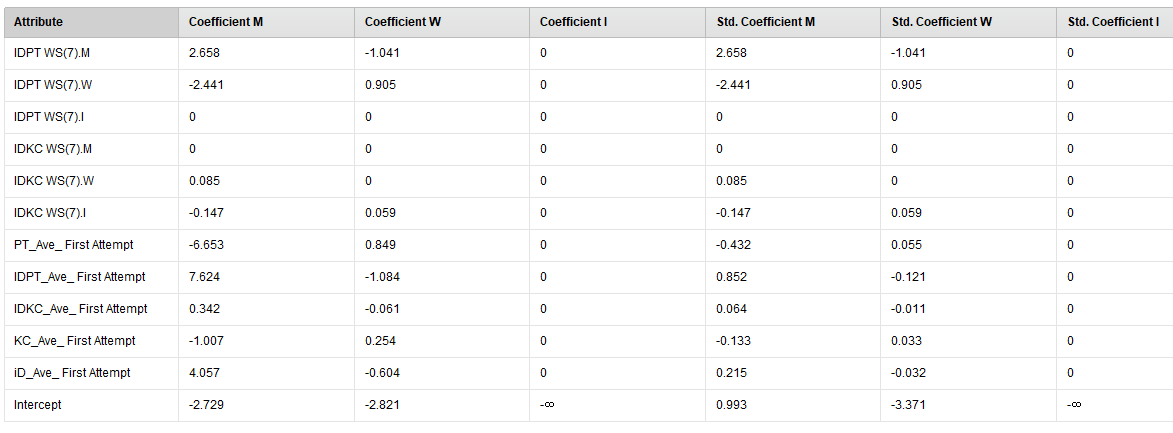
* **Predicting IDPT WS(10) with (7)**

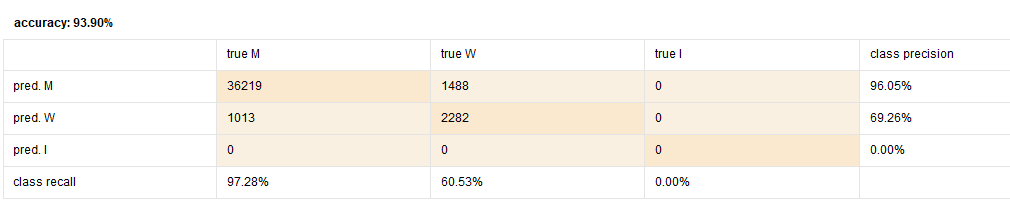
****

****

**root\_mean\_squared\_error: 0.220 +/- 0.000**

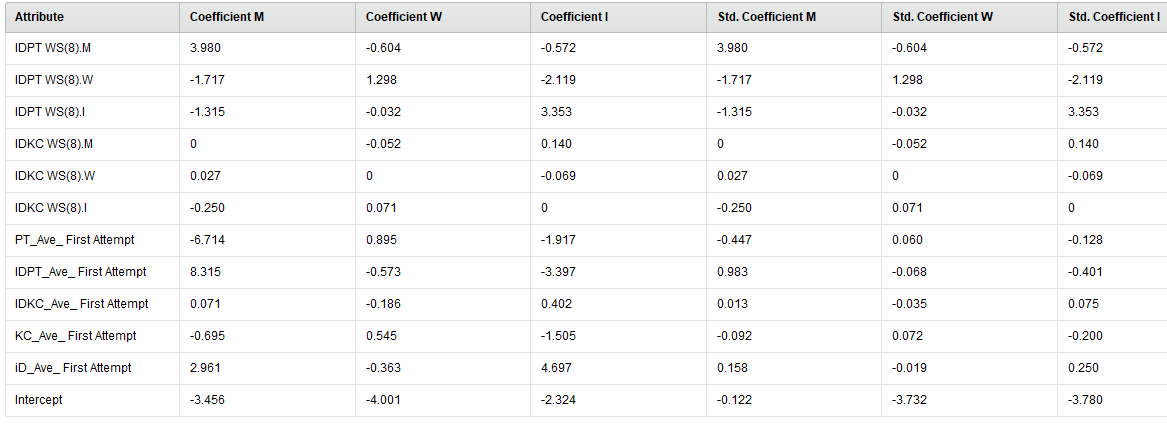
**Without I**

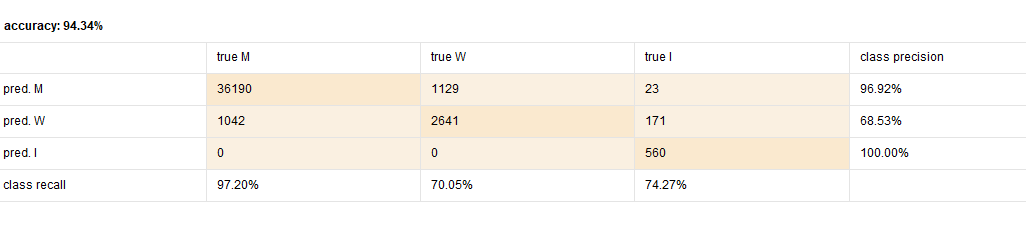
****

****

**root\_mean\_squared\_error: 0.203 +/- 0.000**

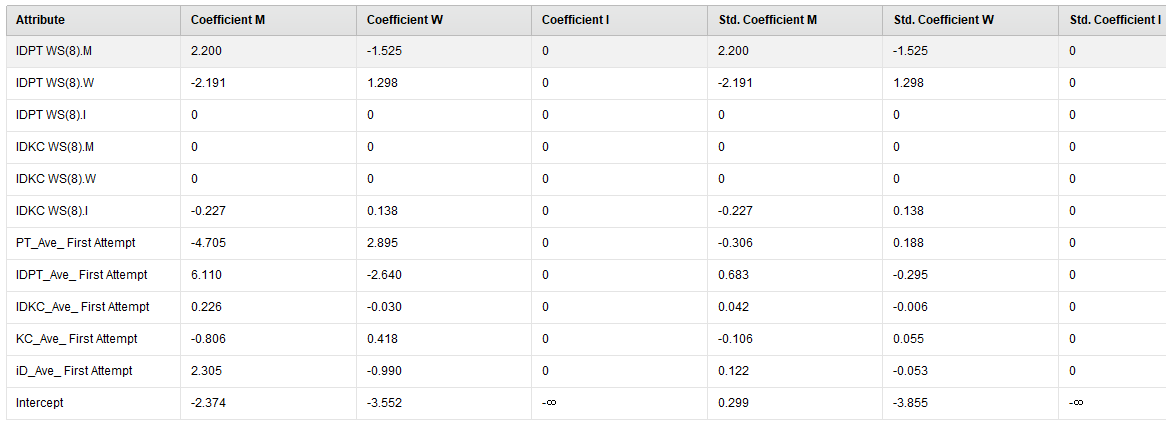
* **Predicting IDPT WS(10) with (8)**

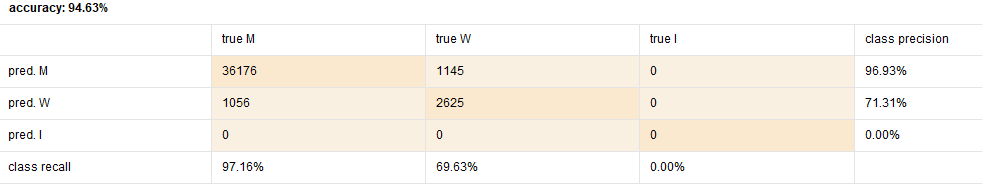
****

****

**root\_mean\_squared\_error: 0.201 +/- 0.000**

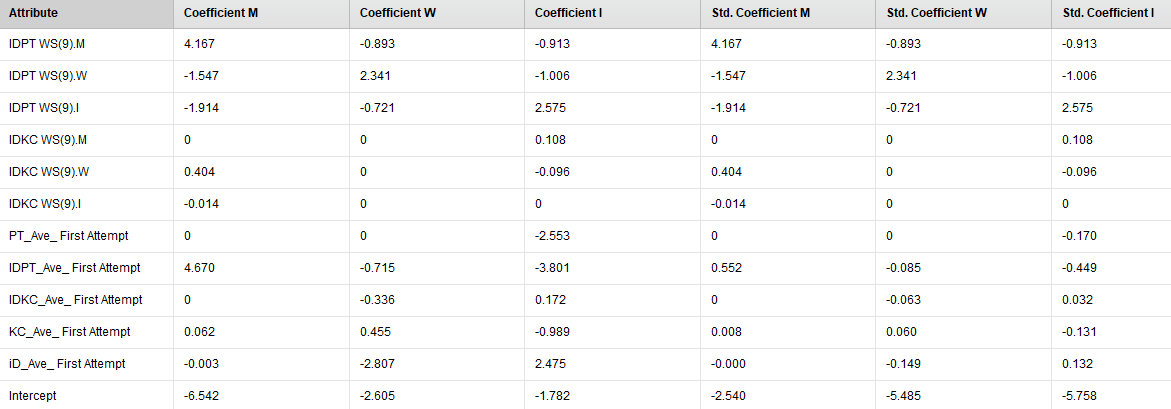
**Without I**

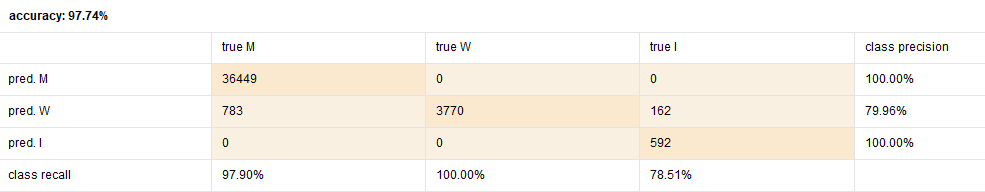
****

****

**root\_mean\_squared\_error: 0.187 +/- 0.000**

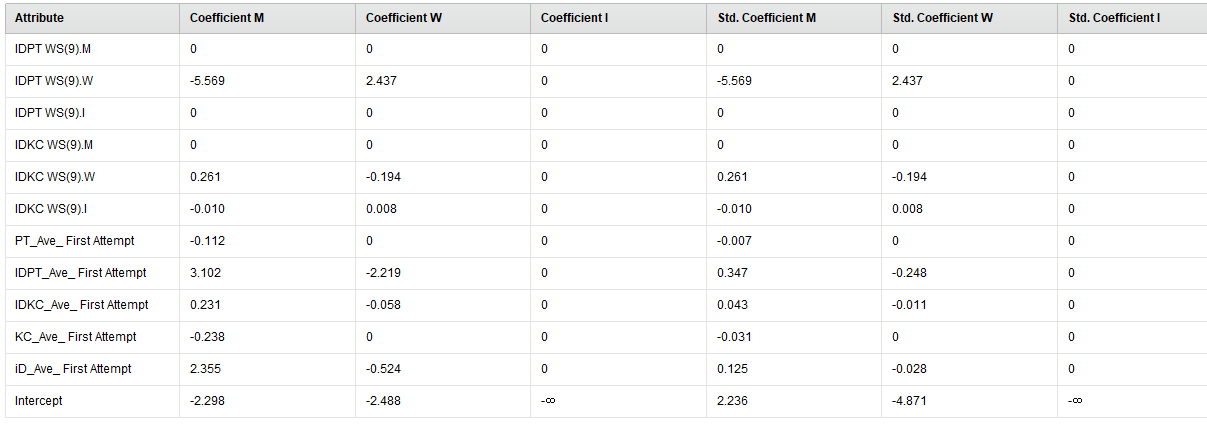
* **Predicting IDPT WS(10) with (9)**

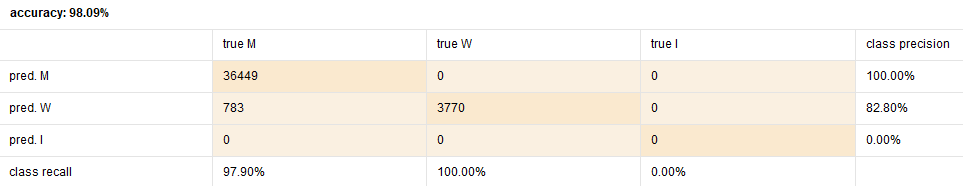
****

****

**root\_mean\_squared\_error: 0.138 +/- 0.000**

**Without I**

****

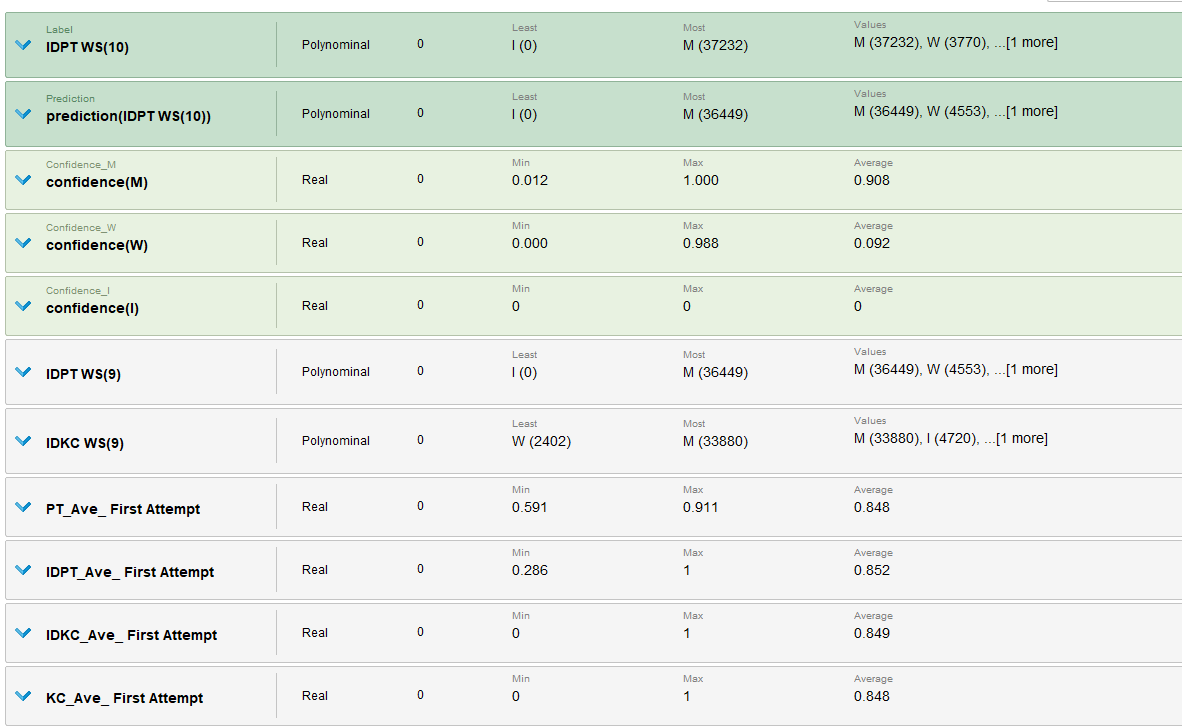
****

**root\_mean\_squared\_error: 0.122 +/- 0.000**

**2/1**

* **Kc ws/ pt ws**
* **Exclude indeterminate.**
* **Check output**
* **Common answer**
* **What’s new here…..?**

****

****

# column 선택해서 subset 만들기. 데이터 프레임이어야함.

pilot1<-subset(data, select=c("ID\_Ave\_FirstAttempt\_Opp1","PT\_Ave\_FirstAttempt\_Opp1","IDPT\_Ave\_First Attempt\_OPP1","IDKC\_Ave\_First Attempt\_Opp1","KC\_Ave\_FirstAttempt\_Opp1","IDPT WS\_10"))

#특정 열 추출.

pilot2<-pilot1[which(pilot1$`IDPT WS\_10`=='M'|pilot1$`IDPT WS\_10`=='W'),]

summary(pilot2)

#엑셀로 export.

library(xlsx)

install.packages("xlsx")

write.xlsx(pilot2, "C:/users/park/onedrive/seoyeondata/pilot2.xlsx")

model <- glm(pilot2$`IDPT WS\_10`~pilot2$ID\_Ave\_FirstAttempt\_Opp1+pilot2$PT\_Ave\_FirstAttempt\_Opp1+pilot2$`IDPT\_Ave\_First Attempt\_OPP1`+pilot2$`IDKC\_Ave\_First Attempt\_Opp1`+pilot2$KC\_Ave\_FirstAttempt\_Opp1,family=binomial)

summary(model)

**>> export the dataset : ‘delete idkc\_i’, ‘delete idpt\_i’, ‘delete kc\_i’ and ‘delete pt\_i’.**

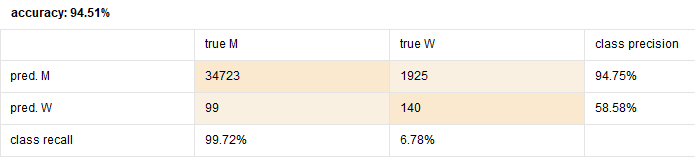
**2/4**

**Now, the main file is ‘Delete blank-Seoyeon’s Macbook pro-4’**

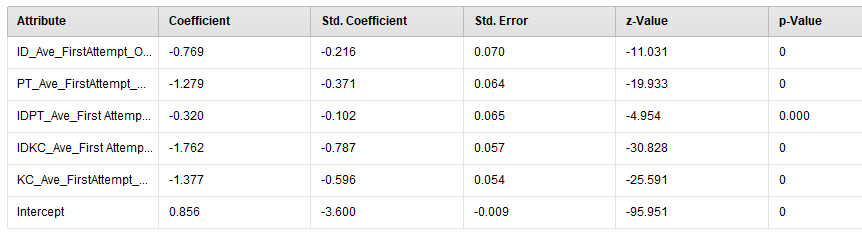
**Title: Can we detect wheel spin students with adaptive detectors?**

**IDKC WS(10)**

* **With all OPP1**

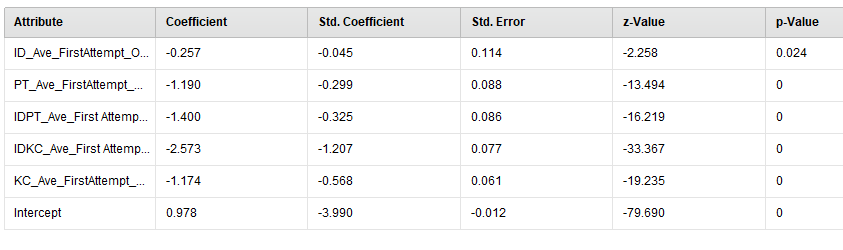
****

**PerformanceVector:  
accuracy: 94.51%  
ConfusionMatrix:  
True: M W  
M: 34723 1925  
W: 99 140  
AUC: 0.842 (positive class: W)  
precision: 58.58% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34723 1925  
W: 99 140  
recall: 6.78% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34723 1925  
W: 99 140**

****

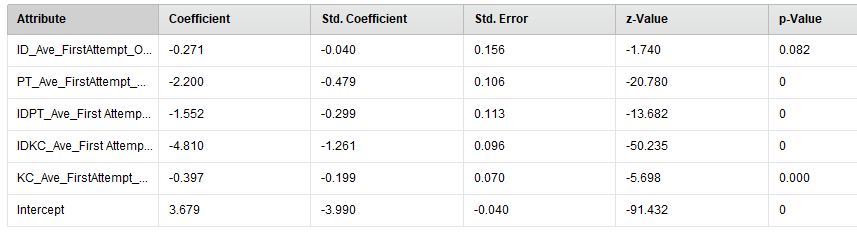
* **With all OPP2**

**PerformanceVector:  
accuracy: 94.42%  
ConfusionMatrix:  
True: M W  
M: 34722 1957  
W: 100 108  
AUC: 0.868 (positive class: W)  
precision: 51.92% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34722 1957  
W: 100 108  
recall: 5.23% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34722 1957  
W: 100 108**

****

**\* with all OPP3**

**accuracy: 94.81%  
ConfusionMatrix:  
True: M W  
M: 34402 1493  
W: 420 572  
AUC: 0.909 (positive class: W)  
precision: 57.66% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34402 1493  
W: 420 572  
recall: 27.70% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34402 1493  
W: 420 572**

****

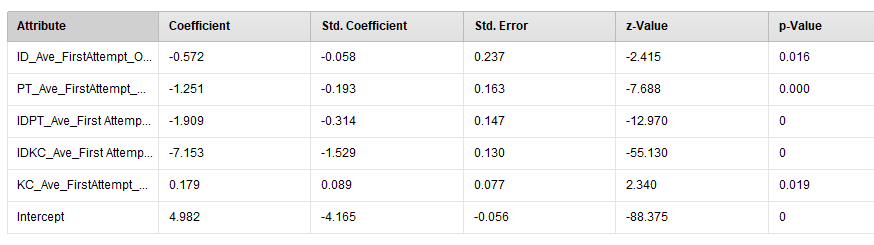
**\*with all OPP4**

**accuracy: 95.17%  
ConfusionMatrix:  
True: M W  
M: 34510 1471  
W: 312 594  
AUC: 0.917 (positive class: W)  
precision: 65.56% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34510 1471  
W: 312 594  
recall: 28.77% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34510 1471  
W: 312 594**

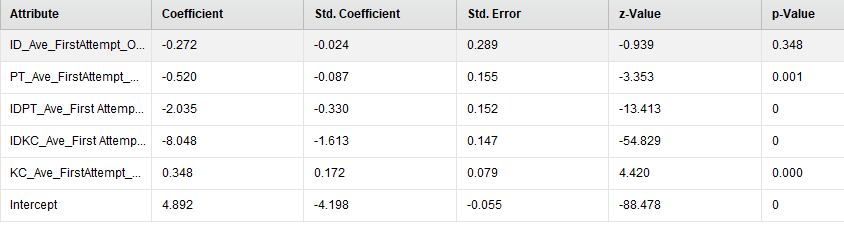
****

**\*with all OPP5**

**PerformanceVector:  
accuracy: 95.16%  
ConfusionMatrix:  
True: M W  
M: 34411 1373  
W: 411 692  
AUC: 0.927 (positive class: W)  
precision: 62.74% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34411 1373  
W: 411 692  
recall: 33.51% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34411 1373  
W: 411 692**

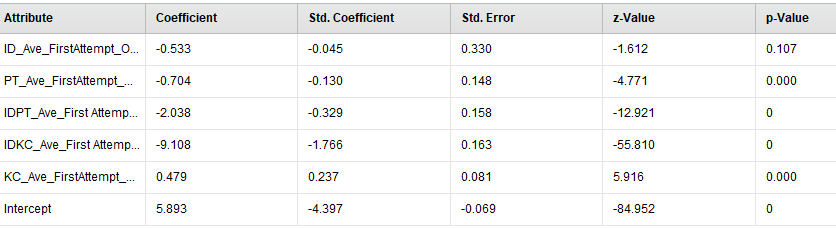
****

**\*with all OPP6**

****

**PerformanceVector:  
accuracy: 95.67%  
ConfusionMatrix:  
True: M W  
M: 34549 1323  
W: 273 742  
AUC: 0.926 (positive class: W)  
precision: 73.10% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34549 1323  
W: 273 742  
recall: 35.93% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34549 1323  
W: 273 742**

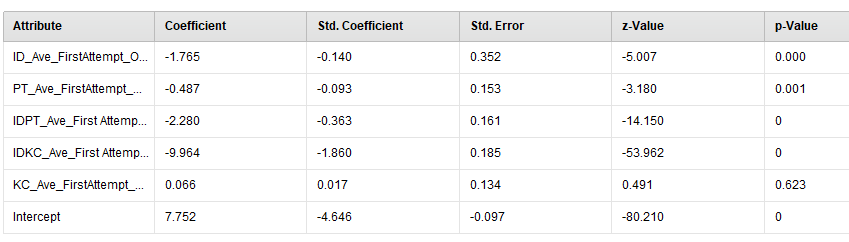
**\*WITH ALL OPP7**

****

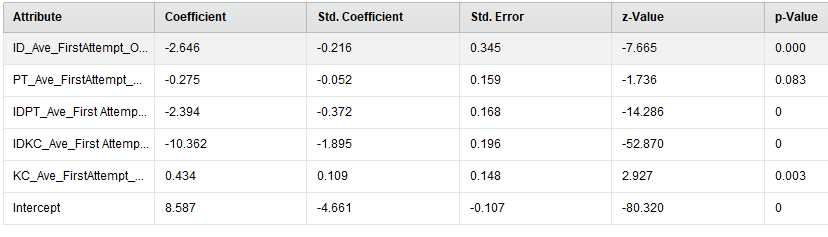
**PerformanceVector:  
accuracy: 95.91%  
ConfusionMatrix:  
True: M W  
M: 34474 1161  
W: 348 904  
AUC: 0.938 (positive class: W)  
precision: 72.20% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34474 1161  
W: 348 904  
recall: 43.78% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34474 1161  
W: 348 904**

**\*with all OPP8**

**PerformanceVector:  
accuracy: 95.99%  
ConfusionMatrix:  
True: M W  
M: 34516 1174  
W: 306 891  
AUC: 0.950 (positive class: W)  
precision: 74.44% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34516 1174  
W: 306 891  
recall: 43.15% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34516 1174  
W: 306 891**

****

**\*with all OPP9**

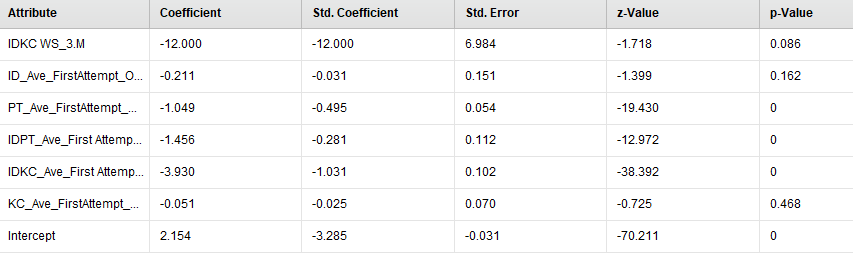
****

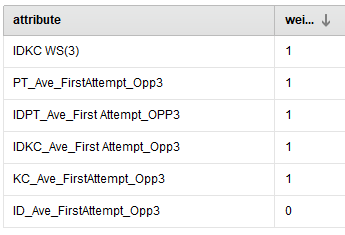
**PerformanceVector:  
accuracy: 96.15%  
ConfusionMatrix:  
True: M W  
M: 34461 1058  
W: 361 1007  
AUC: 0.952 (positive class: W)  
precision: 73.61% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34461 1058  
W: 361 1007  
recall: 48.77% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34461 1058  
W: 361 1007**

**------------------------------------------------------------------**

* **+idkc opp3**

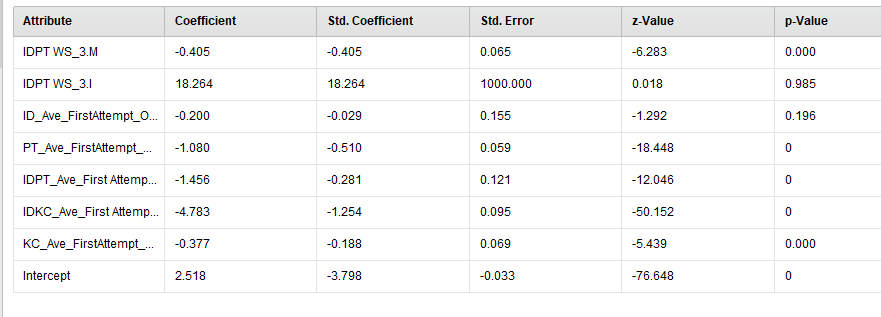
**PerformanceVector:  
accuracy: 94.85%  
ConfusionMatrix:  
True: M W  
M: 34545 1622  
W: 277 443  
AUC: 0.926 (positive class: W)  
precision: 61.53% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34545 1622  
W: 277 443  
recall: 21.45% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34545 1622  
W: 277 443**

****

****

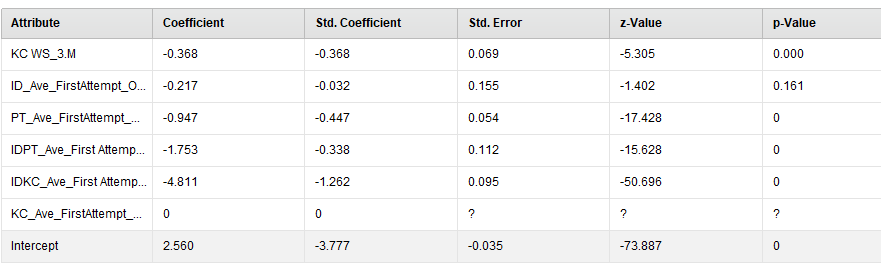
* **+idpt ws3**

**PerformanceVector:  
accuracy: 94.72%  
ConfusionMatrix:  
True: M W  
M: 34457 1583  
W: 365 482  
AUC: 0.907 (positive class: W)  
precision: 56.91% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34457 1583  
W: 365 482  
recall: 23.34% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34457 1583  
W: 365 482**

****

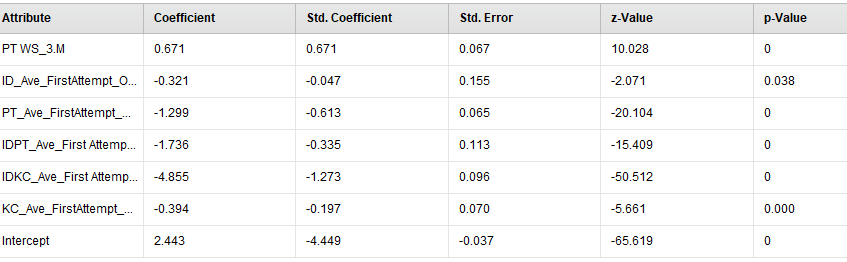
* **+kc ws 3**

**PerformanceVector:  
accuracy: 94.74%  
ConfusionMatrix:  
True: M W  
M: 34449 1569  
W: 373 496  
AUC: 0.908 (positive class: W)  
precision: 57.08% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34449 1569  
W: 373 496  
recall: 24.02% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34449 1569  
W: 373 496**

****

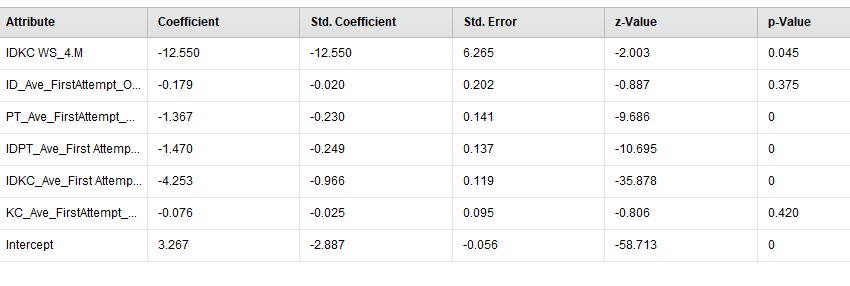
* **+pt opp3**

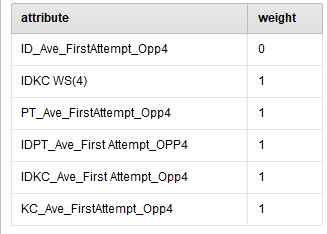
**PerformanceVector:  
accuracy: 94.84%  
ConfusionMatrix:  
True: M W  
M: 34467 1547  
W: 355 518  
AUC: 0.908 (positive class: W)  
precision: 59.34% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34467 1547  
W: 355 518  
recall: 25.08% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34467 1547  
W: 355 518**

****

* **+idkc opp4**

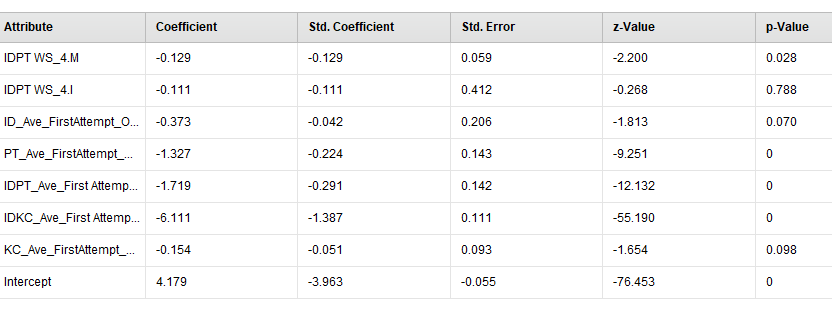
**PerformanceVector:  
accuracy: 95.19%  
ConfusionMatrix:  
True: M W  
M: 34542 1495  
W: 280 570  
AUC: 0.944 (positive class: W)  
precision: 67.06% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34542 1495  
W: 280 570  
recall: 27.60% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34542 1495  
W: 280 570**

****

****

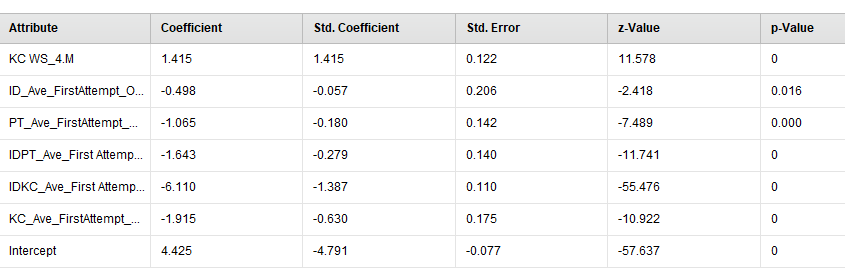
**+idpt opp4**

**PerformanceVector:  
accuracy: 95.14%  
ConfusionMatrix:  
True: M W  
M: 34471 1440  
W: 351 625  
AUC: 0.917 (positive class: W)  
precision: 64.04% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34471 1440  
W: 351 625  
recall: 30.27% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34471 1440  
W: 351 625**

****

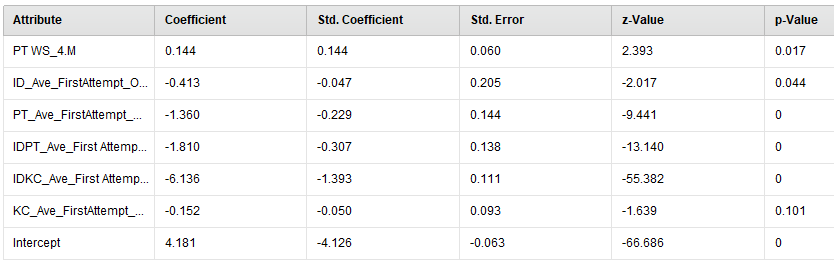
**+kc ws4**

**PerformanceVector:  
accuracy: 95.27%  
ConfusionMatrix:  
True: M W  
M: 34474 1397  
W: 348 668  
AUC: 0.922 (positive class: W)  
precision: 65.75% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34474 1397  
W: 348 668  
recall: 32.35% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34474 1397  
W: 348 668**

****

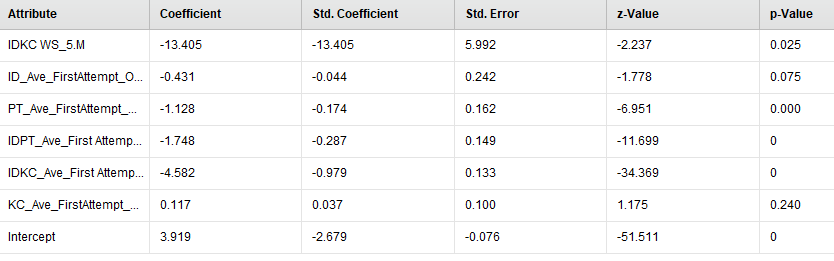
**+pt opp4**

**PerformanceVector:  
accuracy: 95.17%  
ConfusionMatrix:  
True: M W  
M: 34498 1456  
W: 324 609  
AUC: 0.918 (positive class: W)  
precision: 65.27% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34498 1456  
W: 324 609  
recall: 29.49% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34498 1456  
W: 324 609**

****

**+idkc opp5**

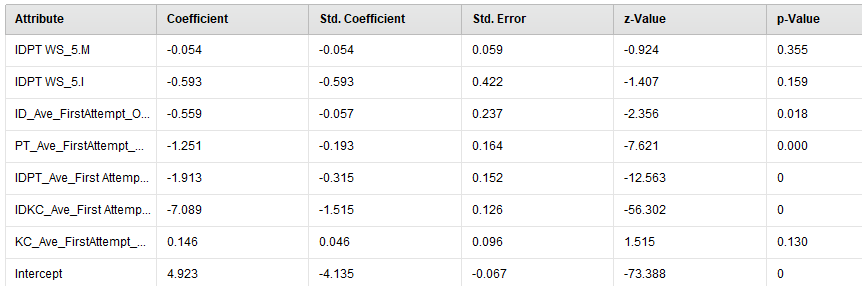
**PerformanceVector:  
accuracy: 95.41%  
ConfusionMatrix:  
True: M W  
M: 34447 1317  
W: 375 748  
AUC: 0.960 (positive class: W)  
precision: 66.61% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34447 1317  
W: 375 748  
recall: 36.22% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34447 1317  
W: 375 748**

****

****

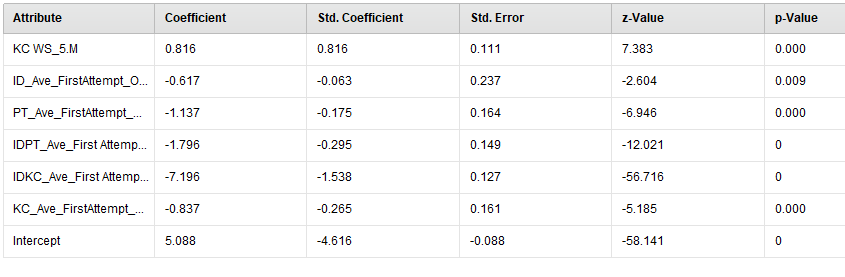
**+idpt opp5**

**PerformanceVector:  
accuracy: 95.17%  
ConfusionMatrix:  
True: M W  
M: 34419 1379  
W: 403 686  
AUC: 0.926 (positive class: W)  
precision: 62.99% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34419 1379  
W: 403 686  
recall: 33.22% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34419 1379  
W: 403 686**

****

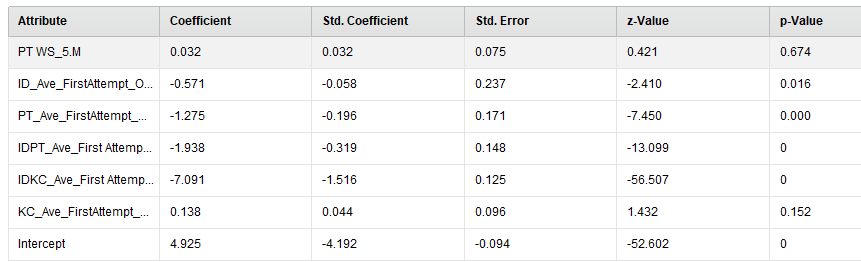
**+kc ws opp5**

**PerformanceVector:  
accuracy: 95.24%  
ConfusionMatrix:  
True: M W  
M: 34444 1376  
W: 378 689  
AUC: 0.927 (positive class: W)  
precision: 64.57% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34444 1376  
W: 378 689  
recall: 33.37% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34444 1376  
W: 378 689**

****

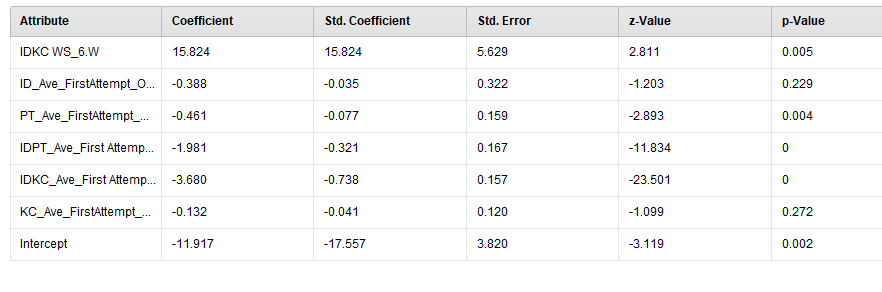
**+pt ws opp5**

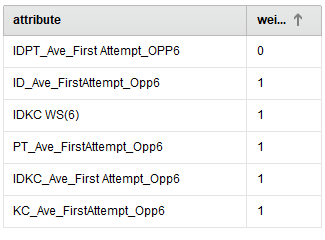
**PerformanceVector:  
accuracy: 95.17%  
ConfusionMatrix:  
True: M W  
M: 34417 1377  
W: 405 688  
AUC: 0.926 (positive class: W)  
precision: 62.95% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34417 1377  
W: 405 688  
recall: 33.32% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34417 1377  
W: 405 688**

****

**+idkc ws opp6**

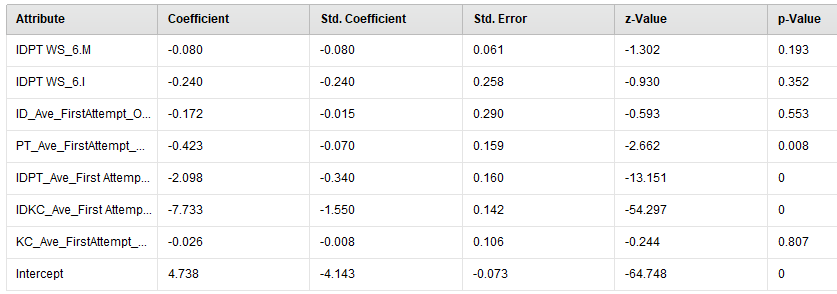
**PerformanceVector:  
accuracy: 96.28%  
ConfusionMatrix:  
True: M W  
M: 34385 937  
W: 437 1128  
AUC: 0.977 (positive class: W)  
precision: 72.08% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34385 937  
W: 437 1128  
recall: 54.62% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34385 937  
W: 437 1128**

****

****

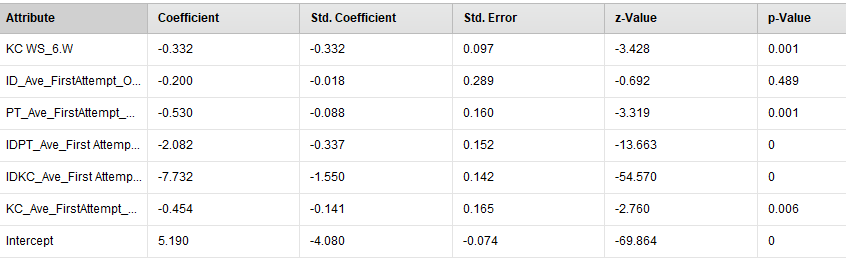
**+idpt ws opp6**

**PerformanceVector:  
accuracy: 95.71%  
ConfusionMatrix:  
True: M W  
M: 34550 1312  
W: 272 753  
AUC: 0.924 (positive class: W)  
precision: 73.46% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34550 1312  
W: 272 753  
recall: 36.46% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34550 1312  
W: 272 753**

****

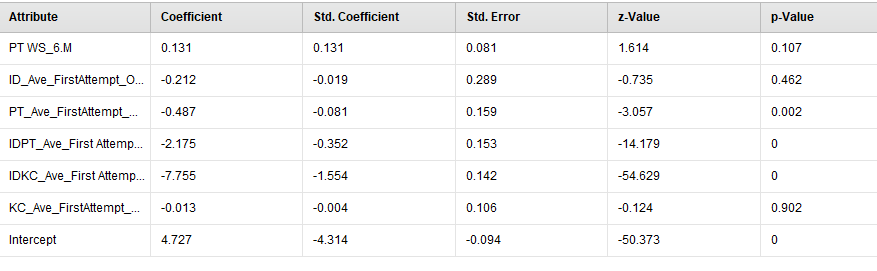
**+kc ws opp6**

**PerformanceVector:  
accuracy: 95.69%  
ConfusionMatrix:  
True: M W  
M: 34553 1322  
W: 269 743  
AUC: 0.924 (positive class: W)  
precision: 73.42% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34553 1322  
W: 269 743  
recall: 35.98% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34553 1322  
W: 269 743**

****

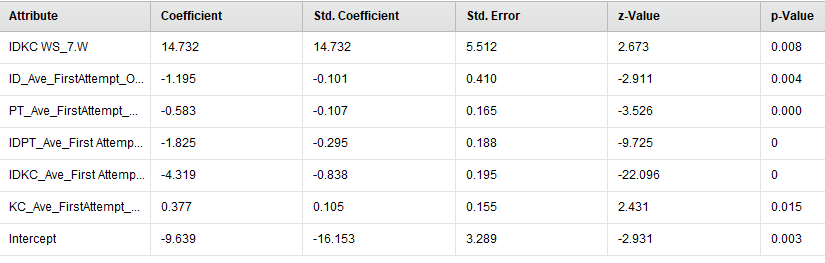
**+pt ws opp6**

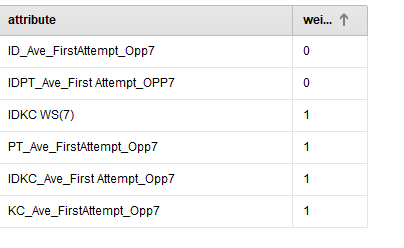
**PerformanceVector:  
accuracy: 95.64%  
ConfusionMatrix:  
True: M W  
M: 34533 1320  
W: 289 745  
AUC: 0.924 (positive class: W)  
precision: 72.05% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34533 1320  
W: 289 745  
recall: 36.08% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34533 1320  
W: 289 745**

****

**+idkc ws opp7**

**PerformanceVector:  
accuracy: 96.55%  
ConfusionMatrix:  
True: M W  
M: 34262 711  
W: 560 1354  
AUC: 0.986 (positive class: W)  
precision: 70.74% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34262 711  
W: 560 1354  
recall: 65.57% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34262 711  
W: 560 1354**

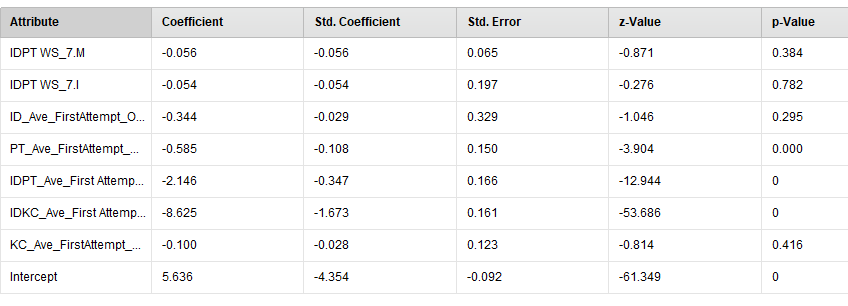
****

****

**PerformanceVector:(remove unused variable)  
accuracy: 96.94% +/- 0.26% (mikro: 96.94%)  
ConfusionMatrix:  
True: M W  
M: 34307 612  
W: 515 1453  
AUC: 0.986 +/- 0.002 (mikro: 0.986) (positive class: W)  
precision: 73.88% +/- 2.60% (mikro: 73.83%) (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34307 612  
W: 515 1453  
recall: 70.36% +/- 4.21% (mikro: 70.36%) (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34307 612  
W: 515 1453**

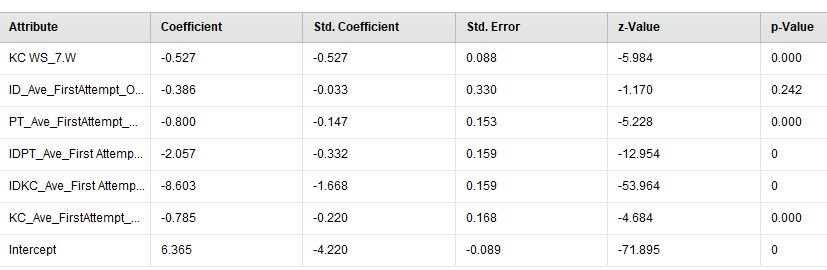
**+idpt ws opp7**

**PerformanceVector:  
accuracy: 95.88%  
ConfusionMatrix:  
True: M W  
M: 34456 1153  
W: 366 912  
AUC: 0.936 (positive class: W)  
precision: 71.36% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34456 1153  
W: 366 912  
recall: 44.16% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34456 1153  
W: 366 912**

****

**+kc opp7**

**PerformanceVector:  
accuracy: 95.94%  
ConfusionMatrix:  
True: M W  
M: 34454 1130  
W: 368 935  
AUC: 0.936 (positive class: W)  
precision: 71.76% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34454 1130  
W: 368 935  
recall: 45.28% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34454 1130  
W: 368 935**

****

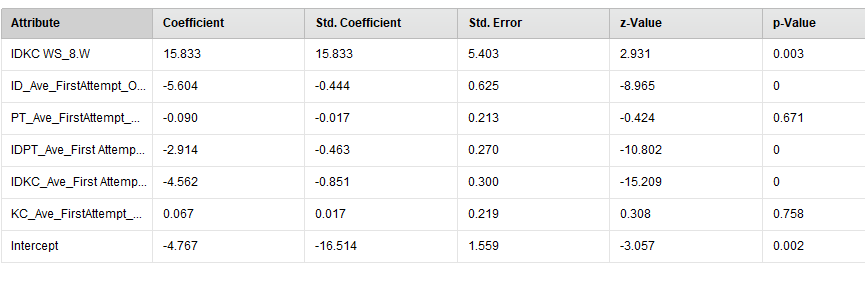
**+pt ws opp7**

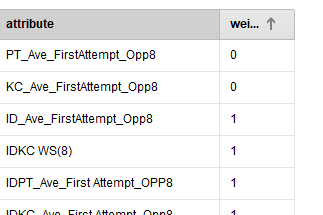
**PerformanceVector:  
accuracy: 96.00%  
ConfusionMatrix:  
True: M W  
M: 34500 1153  
W: 322 912  
AUC: 0.937 (positive class: W)  
precision: 73.91% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34500 1153  
W: 322 912  
recall: 44.16% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34500 1153  
W: 322 912**

****

**+idkc ws opp8**

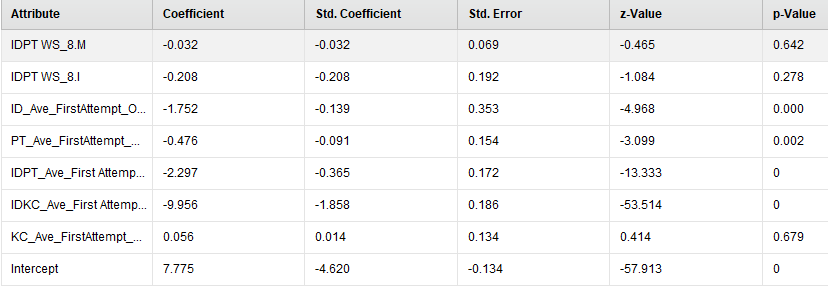
**PerformanceVector:  
accuracy: 98.20%  
ConfusionMatrix:  
True: M W  
M: 34289 130  
W: 533 1935  
AUC: 0.995 (positive class: W)  
precision: 78.40% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34289 130  
W: 533 1935  
recall: 93.70% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34289 130  
W: 533 1935**

****

****

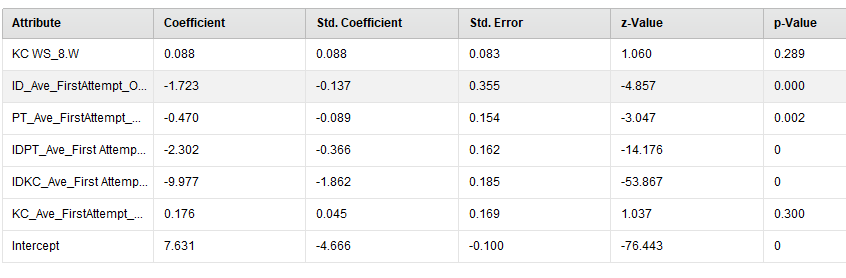
**+idpt ws opp8**

**PerformanceVector:  
accuracy: 96.01%  
ConfusionMatrix:  
True: M W  
M: 34513 1163  
W: 309 902  
AUC: 0.950 (positive class: W)  
precision: 74.48% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34513 1163  
W: 309 902  
recall: 43.68% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34513 1163  
W: 309 902**

****

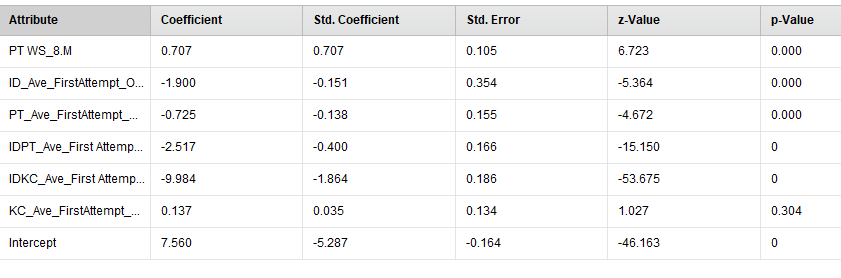
**+kc ws opp8**

**PerformanceVector:  
accuracy: 96.10%  
ConfusionMatrix:  
True: M W  
M: 34506 1124  
W: 316 941  
AUC: 0.950 (positive class: W)  
precision: 74.86% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34506 1124  
W: 316 941  
recall: 45.57% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34506 1124  
W: 316 941**

****

**+pt ws opp8**

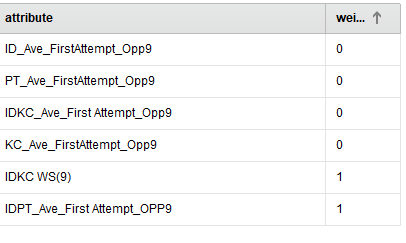
**PerformanceVector:  
accuracy: 96.15%  
ConfusionMatrix:  
True: M W  
M: 34509 1108  
W: 313 957  
AUC: 0.951 (positive class: W)  
precision: 75.35% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34509 1108  
W: 313 957  
recall: 46.34% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34509 1108  
W: 313 957**

****

**+idkc opp9**

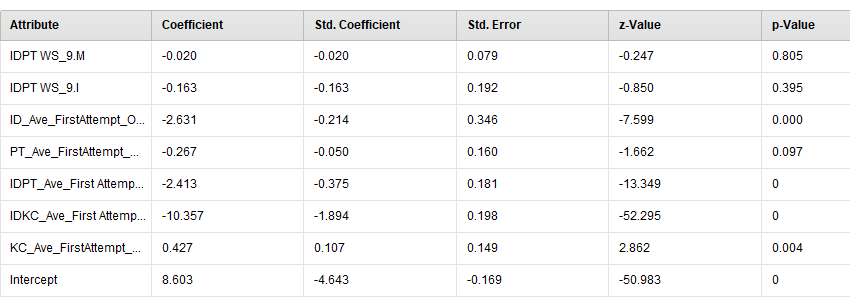
**PerformanceVector:  
accuracy: 98.96%  
ConfusionMatrix:  
True: M W  
M: 34440 0  
W: 382 2065  
AUC: 0.998 (positive class: W)  
precision: 84.39% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34440 0  
W: 382 2065  
recall: 100.00% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34440 0  
W: 382 2065**

****

****

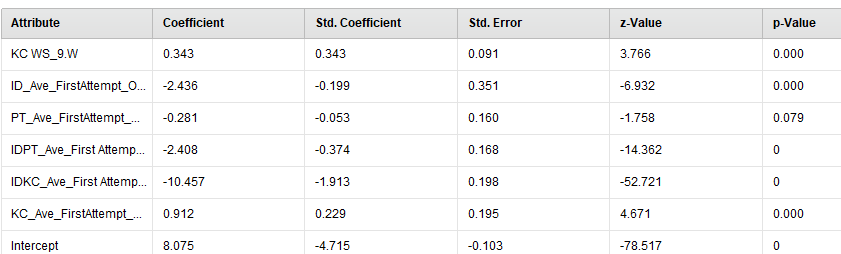
**+idpt opp9**

**PerformanceVector:  
accuracy: 96.15%  
ConfusionMatrix:  
True: M W  
M: 34462 1059  
W: 360 1006  
AUC: 0.951 (positive class: W)  
precision: 73.65% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34462 1059  
W: 360 1006  
recall: 48.72% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34462 1059  
W: 360 1006**

****

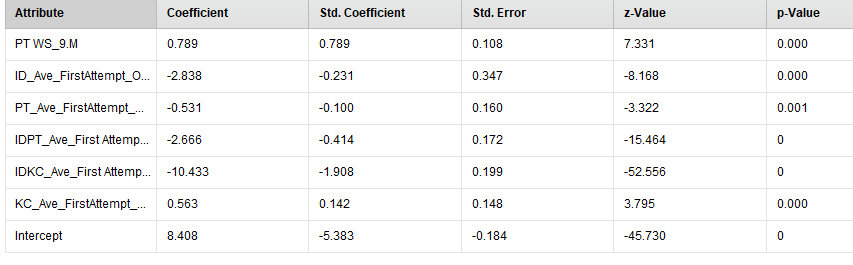
**+kc ws opp9**

**PerformanceVector:  
accuracy: 96.13%  
ConfusionMatrix:  
True: M W  
M: 34472 1079  
W: 350 986  
AUC: 0.952 (positive class: W)  
precision: 73.80% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34472 1079  
W: 350 986  
recall: 47.75% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34472 1079  
W: 350 986**

****

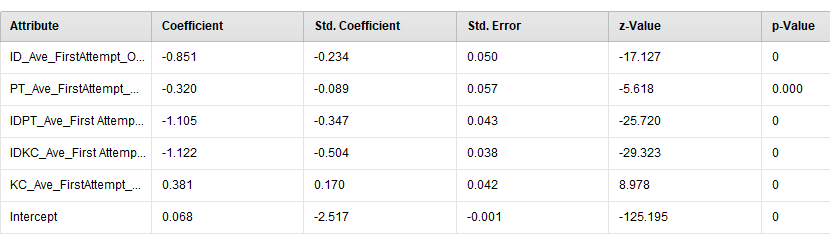
**+pt opp9**

**PerformanceVector:  
accuracy: 96.26%  
ConfusionMatrix:  
True: M W  
M: 34481 1038  
W: 341 1027  
AUC: 0.953 (positive class: W)  
precision: 75.07% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34481 1038  
W: 341 1027  
recall: 49.73% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34481 1038  
W: 341 1027**

****

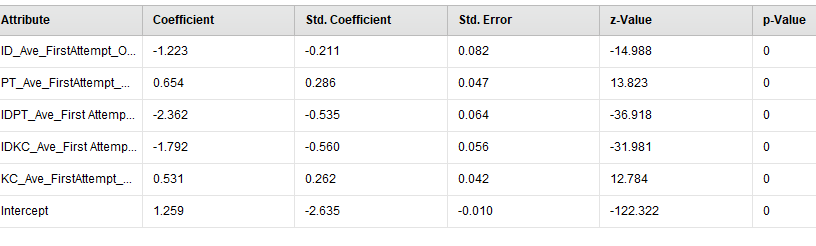
**----------------------------------------------------------------------**

* **IDPT WS(10)**
* **With all opp1**
* **PerformanceVector:  
  accuracy: 90.84%  
  ConfusionMatrix:  
  True: M W  
  M: 37071 3595  
  W: 161 175  
  AUC: 0.710 (positive class: W)  
  precision: 52.08% (positive class: W)  
  ConfusionMatrix:  
  True: M W  
  M: 37071 3595  
  W: 161 175  
  recall: 4.64% (positive class: W)  
  ConfusionMatrix:  
  True: M W  
  M: 37071 3595  
  W: 161 175**

****

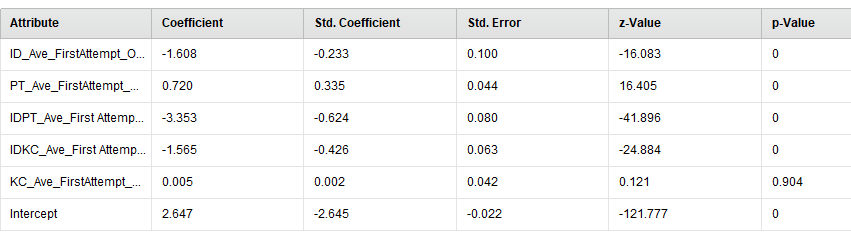
**+with all opp2**

**PerformanceVector:  
accuracy: 91.07%  
ConfusionMatrix:  
True: M W  
M: 36914 3345  
W: 318 425  
AUC: 0.756 (positive class: W)  
precision: 57.20% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36914 3345  
W: 318 425  
recall: 11.27% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36914 3345  
W: 318 425**

****

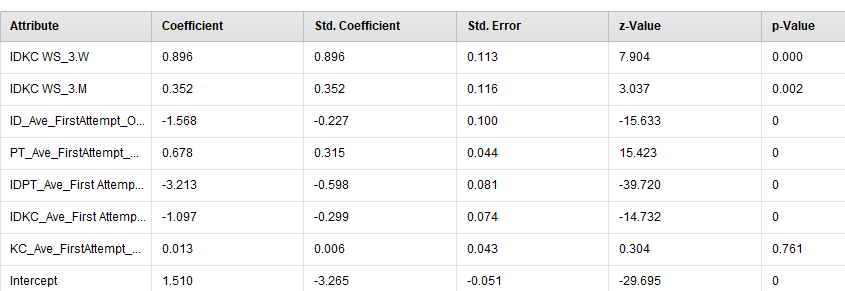
**+with all opp3**

**PerformanceVector:  
accuracy: 90.96%  
ConfusionMatrix:  
True: M W  
M: 36929 3404  
W: 303 366  
AUC: 0.761 (positive class: W)  
precision: 54.71% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36929 3404  
W: 303 366  
recall: 9.71% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36929 3404  
W: 303 366**

****

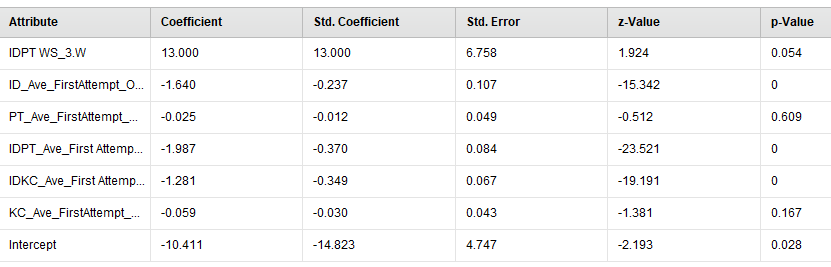
**+idkc opp3**

**PerformanceVector:  
accuracy: 90.98%  
ConfusionMatrix:  
True: M W  
M: 36984 3450  
W: 248 320  
AUC: 0.769 (positive class: W)  
precision: 56.34% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36984 3450  
W: 248 320  
recall: 8.49% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36984 3450  
W: 248 320**

****

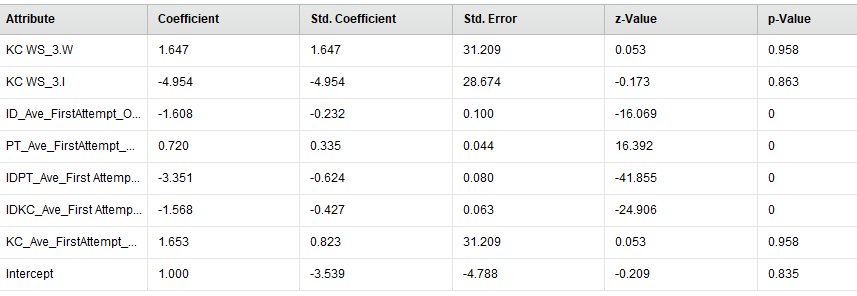
**+idpt opp3**

**PerformanceVector:  
accuracy: 91.13%  
ConfusionMatrix:  
True: M W  
M: 36974 3380  
W: 258 390  
AUC: 0.860 (positive class: W)  
precision: 60.19% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36974 3380  
W: 258 390  
recall: 10.34% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36974 3380  
W: 258 390**

****

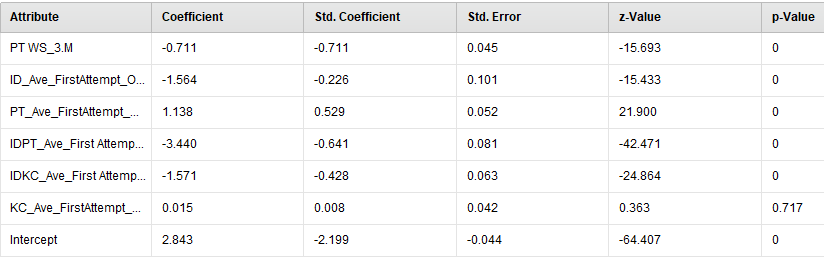
**+kc opp3**

**PerformanceVector:  
accuracy: 90.97%  
ConfusionMatrix:  
True: M W  
M: 36941 3413  
W: 291 357  
AUC: 0.762 (positive class: W)  
precision: 55.09% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36941 3413  
W: 291 357  
recall: 9.47% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36941 3413  
W: 291 357**

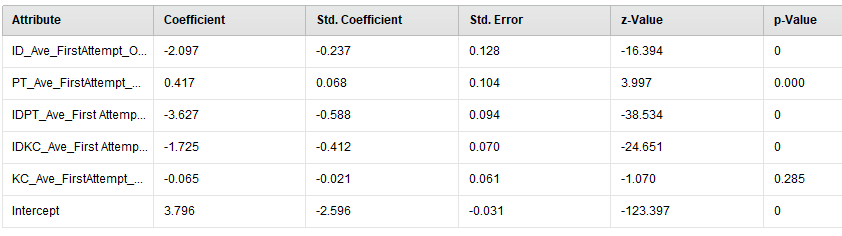
****

**+pt ws opp3**

**PerformanceVector:  
accuracy: 90.89%  
ConfusionMatrix:  
True: M W  
M: 36904 3408  
W: 328 362  
AUC: 0.771 (positive class: W)  
precision: 52.46% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36904 3408  
W: 328 362  
recall: 9.60% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36904 3408  
W: 328 362**

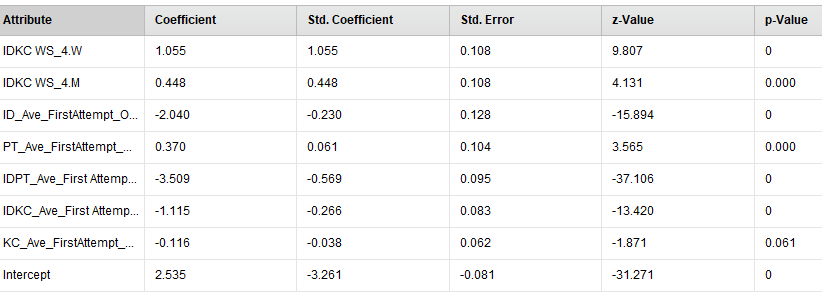
****

* **With all opp4**
* **PerformanceVector:  
  accuracy: 90.90%  
  ConfusionMatrix:  
  True: M W  
  M: 37024 3522  
  W: 208 248  
  AUC: 0.753 (positive class: W)  
  precision: 54.39% (positive class: W)  
  ConfusionMatrix:  
  True: M W  
  M: 37024 3522  
  W: 208 248  
  recall: 6.58% (positive class: W)  
  ConfusionMatrix:  
  True: M W  
  M: 37024 3522  
  W: 208 248**

****

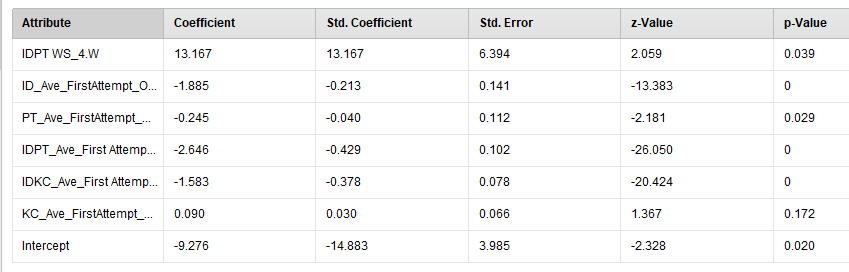
**+idkc opp4**

**PerformanceVector:  
accuracy: 90.87%  
ConfusionMatrix:  
True: M W  
M: 37035 3546  
W: 197 224  
AUC: 0.762 (positive class: W)  
precision: 53.21% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37035 3546  
W: 197 224  
recall: 5.94% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37035 3546  
W: 197 224**

****

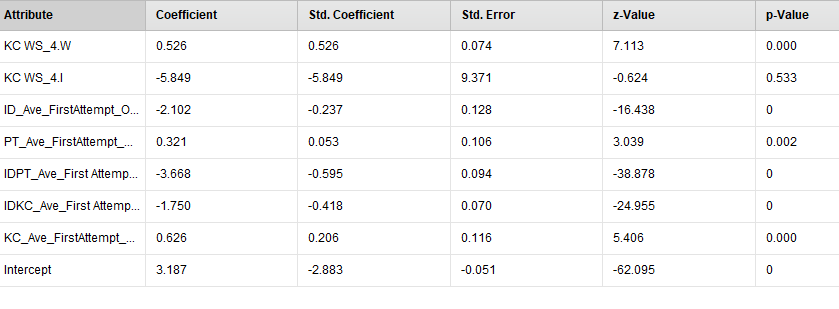
**+idpt opp4**

**PerformanceVector:  
accuracy: 91.12%  
ConfusionMatrix:  
True: M W  
M: 36908 3316  
W: 324 454  
AUC: 0.881 (positive class: W)  
precision: 58.35% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36908 3316  
W: 324 454  
recall: 12.04% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36908 3316  
W: 324 454**

****

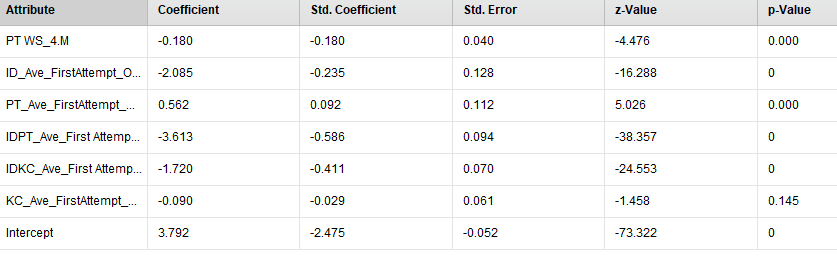
**+kc opp 4**

**PerformanceVector:  
accuracy: 90.89%  
ConfusionMatrix:  
True: M W  
M: 37012 3515  
W: 220 255  
AUC: 0.753 (positive class: W)  
precision: 53.68% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37012 3515  
W: 220 255  
recall: 6.76% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37012 3515  
W: 220 255**

****

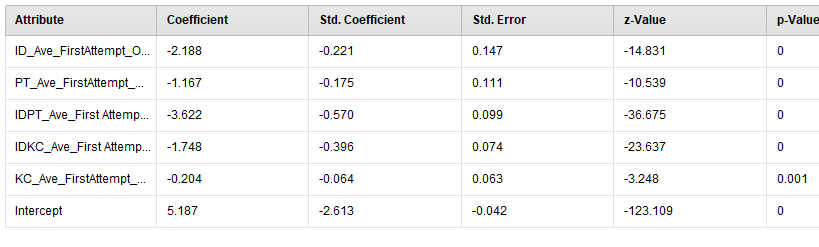
**+pt opp4**

**PerformanceVector:  
accuracy: 90.96%  
ConfusionMatrix:  
True: M W  
M: 37042 3516  
W: 190 254  
AUC: 0.753 (positive class: W)  
precision: 57.21% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37042 3516  
W: 190 254  
recall: 6.74% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37042 3516  
W: 190 254**

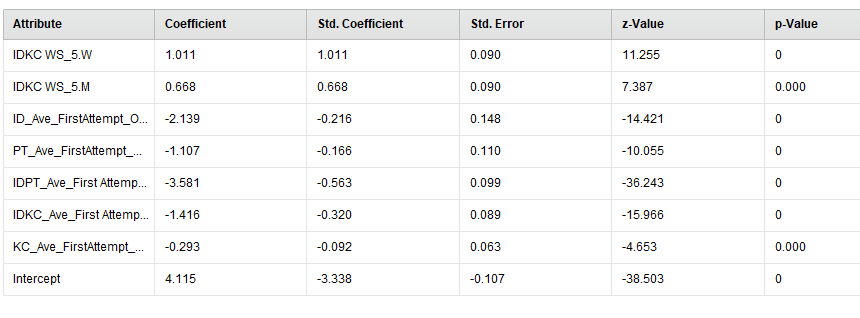
****

**+with all opp5**

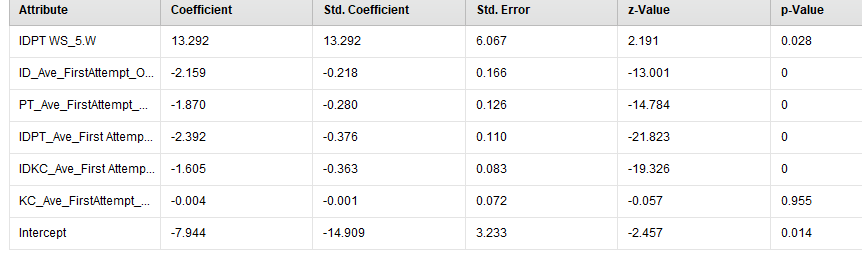
**PerformanceVector:  
accuracy: 90.82%  
ConfusionMatrix:  
True: M W  
M: 36962 3493  
W: 270 277  
AUC: 0.757 (positive class: W)  
precision: 50.64% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36962 3493  
W: 270 277  
recall: 7.35% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36962 3493  
W: 270 277**

****

**PerformanceVector:  
accuracy: 90.88%  
ConfusionMatrix:  
True: M W  
M: 36963 3472  
W: 269 298  
AUC: 0.765 (positive class: W)  
precision: 52.56% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36963 3472  
W: 269 298  
recall: 7.90% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36963 3472  
W: 269 298**

****

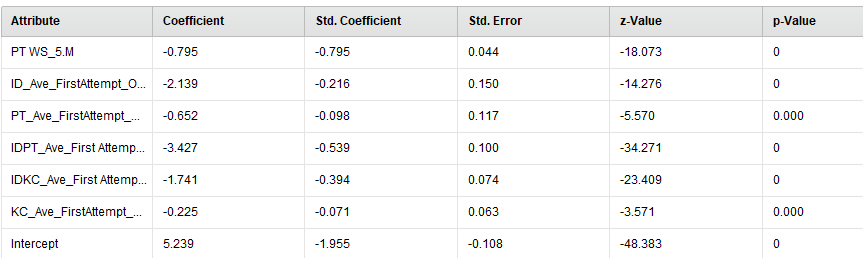
**PerformanceVector:  
accuracy: 91.45%  
ConfusionMatrix:  
True: M W  
M: 36849 3121  
W: 383 649  
AUC: 0.895 (positive class: W)  
precision: 62.89% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36849 3121  
W: 383 649  
recall: 17.21% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36849 3121  
W: 383 649**

****

**PerformanceVector:  
accuracy: 90.74%  
ConfusionMatrix:  
True: M W  
M: 36969 3533  
W: 263 237  
AUC: 0.759 (positive class: W)  
precision: 47.40% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36969 3533  
W: 263 237  
recall: 6.29% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36969 3533  
W: 263 237**

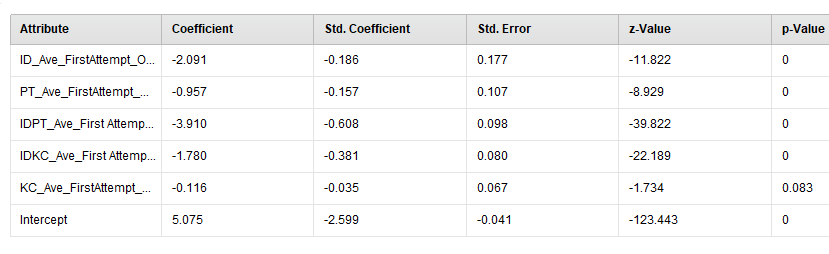
****

**PerformanceVector:  
accuracy: 90.82%  
ConfusionMatrix:  
True: M W  
M: 36929 3459  
W: 303 311  
AUC: 0.776 (positive class: W)  
precision: 50.65% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36929 3459  
W: 303 311  
recall: 8.25% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36929 3459  
W: 303 311**

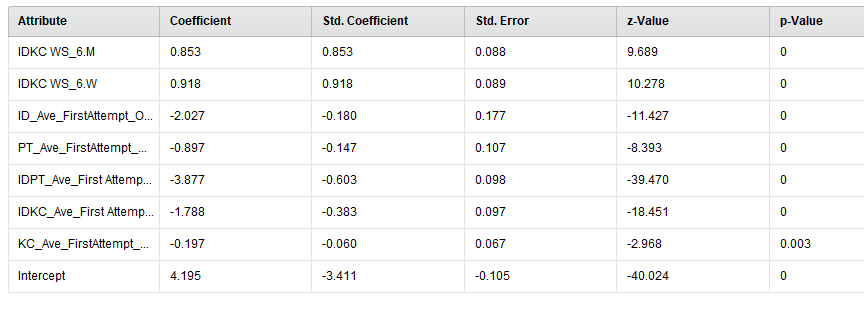
****

**+with all opp6**

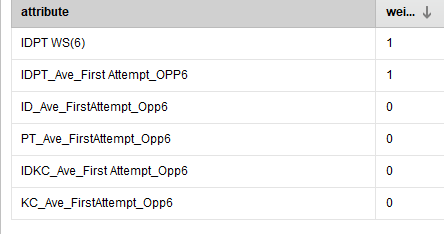
**PerformanceVector:  
accuracy: 90.85%  
ConfusionMatrix:  
True: M W  
M: 37043 3564  
W: 189 206  
AUC: 0.751 (positive class: W)  
precision: 52.15% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37043 3564  
W: 189 206  
recall: 5.46% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37043 3564  
W: 189 206**

****

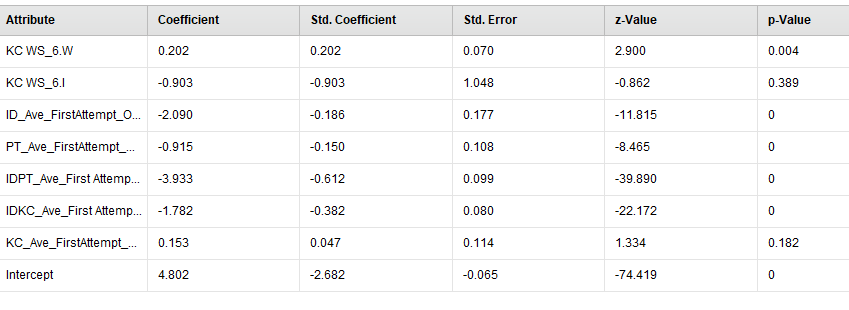
**PerformanceVector:  
accuracy: 90.91%  
ConfusionMatrix:  
True: M W  
M: 37044 3541  
W: 188 229  
AUC: 0.756 (positive class: W)  
precision: 54.92% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37044 3541  
W: 188 229  
recall: 6.07% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37044 3541  
W: 188 229**

****

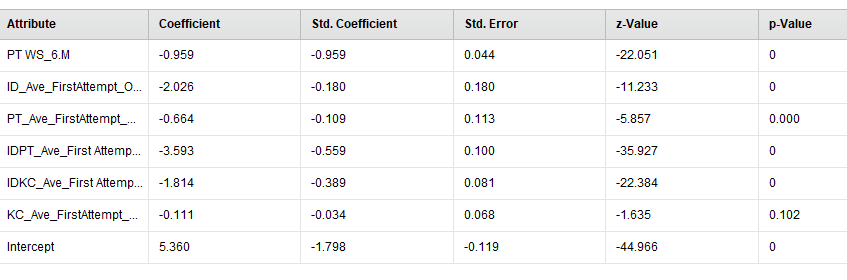
**PerformanceVector:  
accuracy: 92.07%  
ConfusionMatrix:  
True: M W  
M: 36465 2485  
W: 767 1285  
AUC: 0.943 (positive class: W)  
precision: 62.62% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36465 2485  
W: 767 1285  
recall: 34.08% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36465 2485  
W: 767 1285**

****

**PerformanceVector:  
accuracy: 90.89%  
ConfusionMatrix:  
True: M W  
M: 37057 3560  
W: 175 210  
AUC: 0.751 (positive class: W)  
precision: 54.55% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37057 3560  
W: 175 210  
recall: 5.57% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37057 3560  
W: 175 210**

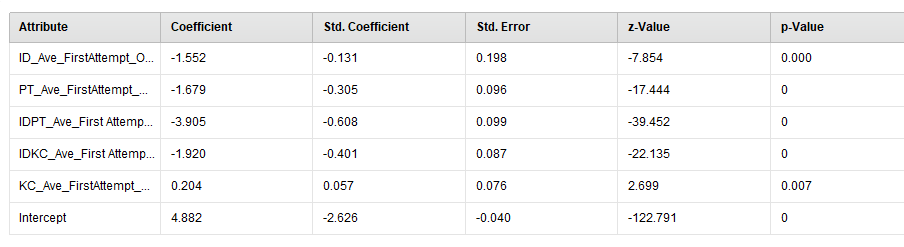
****

**PerformanceVector:  
accuracy: 90.92%  
ConfusionMatrix:  
True: M W  
M: 36952 3442  
W: 280 328  
AUC: 0.773 (positive class: W)  
precision: 53.95% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36952 3442  
W: 280 328  
recall: 8.70% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36952 3442  
W: 280 328**

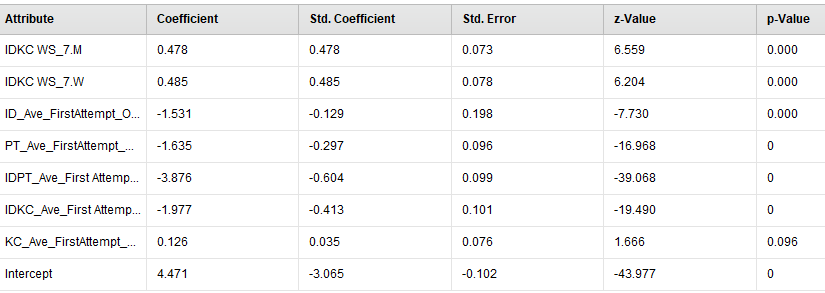
****

**+with all opp7**

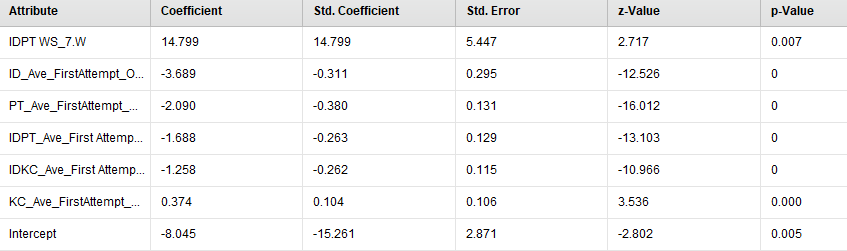
**PerformanceVector:  
accuracy: 91.04%  
ConfusionMatrix:  
True: M W  
M: 37023 3463  
W: 209 307  
AUC: 0.761 (positive class: W)  
precision: 59.50% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37023 3463  
W: 209 307  
recall: 8.14% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37023 3463  
W: 209 307**

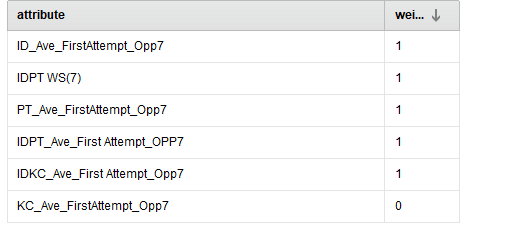
****

**PerformanceVector:  
accuracy: 91.03%  
ConfusionMatrix:  
True: M W  
M: 37014 3458  
W: 218 312  
AUC: 0.762 (positive class: W)  
precision: 58.87% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37014 3458  
W: 218 312  
recall: 8.28% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37014 3458  
W: 218 312**

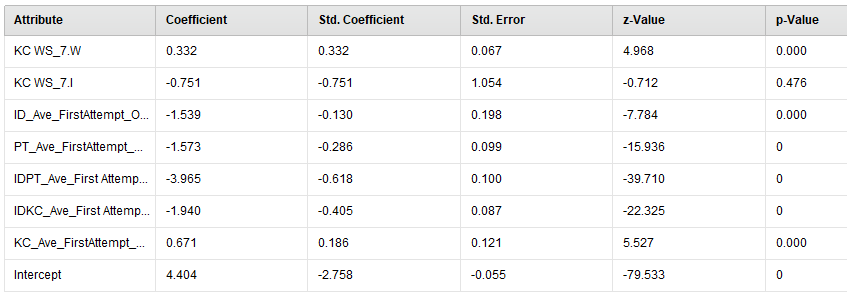
****

**PerformanceVector:  
accuracy: 93.11%  
ConfusionMatrix:  
True: M W  
M: 36241 1836  
W: 991 1934  
AUC: 0.963 (positive class: W)  
precision: 66.12% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36241 1836  
W: 991 1934  
recall: 51.30% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36241 1836  
W: 991 1934**

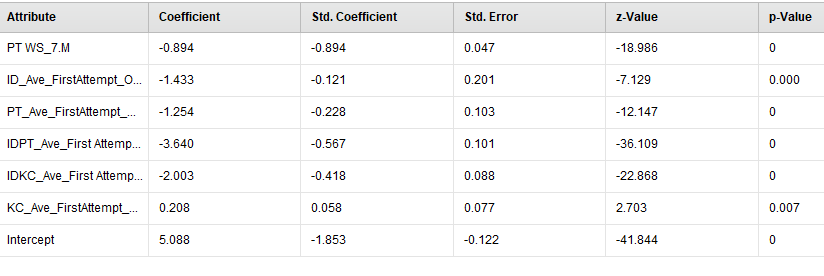
****

****

**PerformanceVector:  
accuracy: 91.11%  
ConfusionMatrix:  
True: M W  
M: 37025 3439  
W: 207 331  
AUC: 0.760 (positive class: W)  
precision: 61.52% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37025 3439  
W: 207 331  
recall: 8.78% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 37025 3439  
W: 207 331**

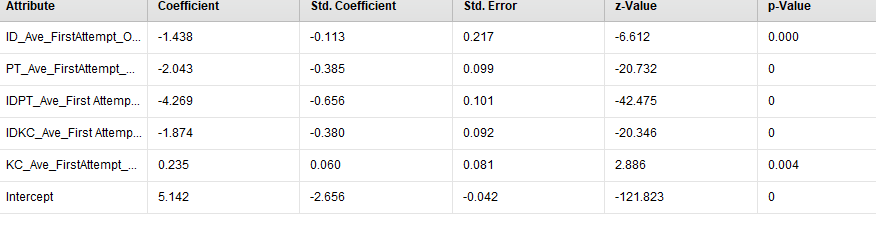
****

**PerformanceVector:  
accuracy: 91.05%  
ConfusionMatrix:  
True: M W  
M: 36904 3341  
W: 328 429  
AUC: 0.775 (positive class: W)  
precision: 56.67% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36904 3341  
W: 328 429  
recall: 11.38% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36904 3341  
W: 328 429**

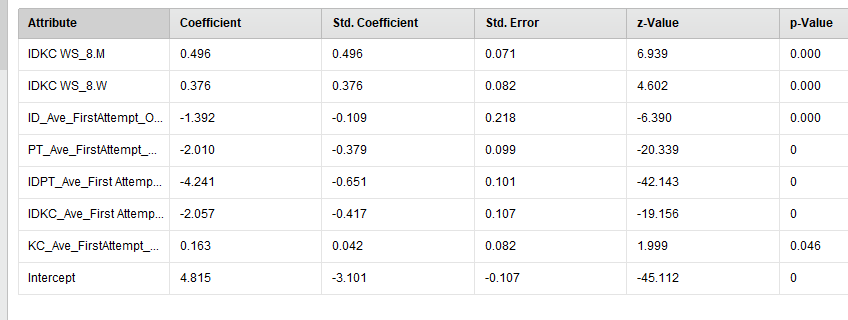
****

**+with all opp8**

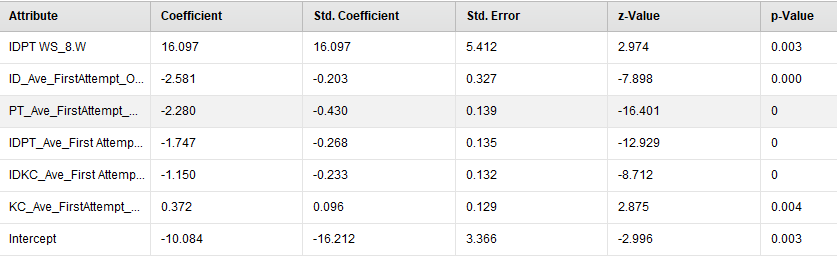
**PerformanceVector:  
accuracy: 91.12%  
ConfusionMatrix:  
True: M W  
M: 36995 3404  
W: 237 366  
AUC: 0.767 (positive class: W)  
precision: 60.70% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36995 3404  
W: 237 366  
recall: 9.71% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36995 3404  
W: 237 366**

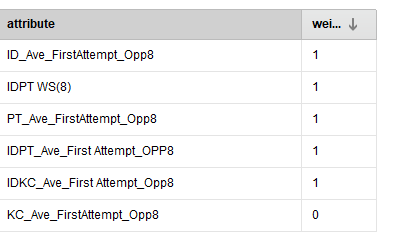
****

**PerformanceVector:  
accuracy: 91.14%  
ConfusionMatrix:  
True: M W  
M: 36998 3399  
W: 234 371  
AUC: 0.769 (positive class: W)  
precision: 61.32% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36998 3399  
W: 234 371  
recall: 9.84% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36998 3399  
W: 234 371**

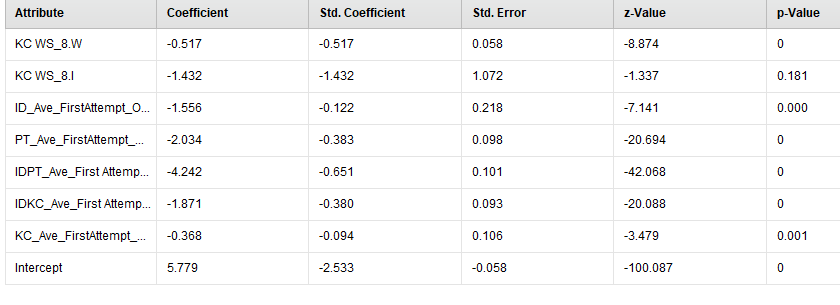
****

**PerformanceVector:  
accuracy: 94.30%  
ConfusionMatrix:  
True: M W  
M: 35865 969  
W: 1367 2801  
AUC: 0.974 (positive class: W)  
precision: 67.20% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 35865 969  
W: 1367 2801  
recall: 74.30% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 35865 969  
W: 1367 2801**

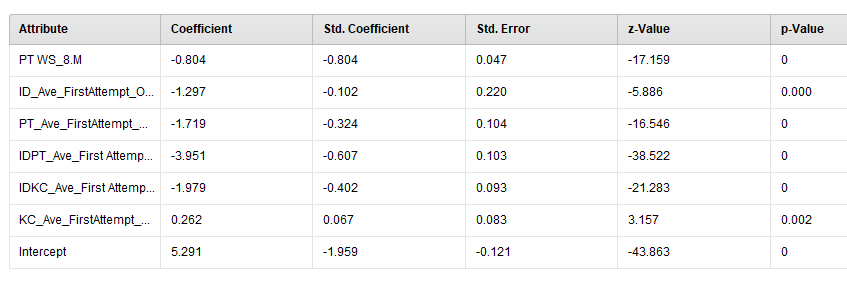
****

****

**PerformanceVector:  
accuracy: 91.18%  
ConfusionMatrix:  
True: M W  
M: 36998 3381  
W: 234 389  
AUC: 0.769 (positive class: W)  
precision: 62.44% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36998 3381  
W: 234 389  
recall: 10.32% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36998 3381  
W: 234 389**

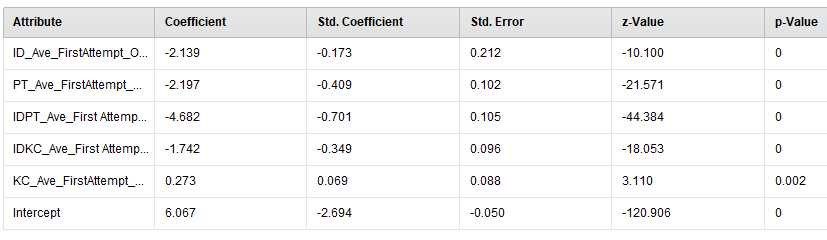
****

**PerformanceVector:  
accuracy: 91.27%  
ConfusionMatrix:  
True: M W  
M: 36964 3312  
W: 268 458  
AUC: 0.777 (positive class: W)  
precision: 63.09% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36964 3312  
W: 268 458  
recall: 12.15% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36964 3312  
W: 268 458**

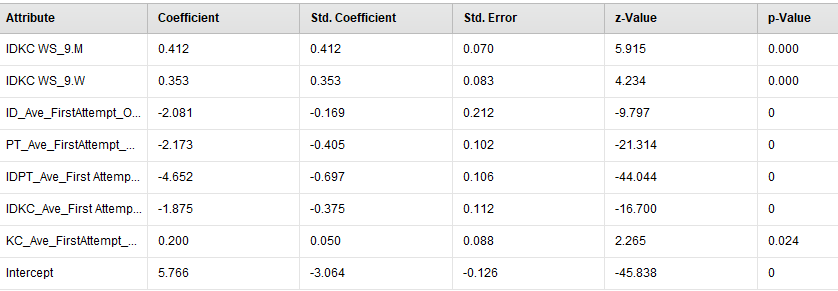
****

**+with all opp9**

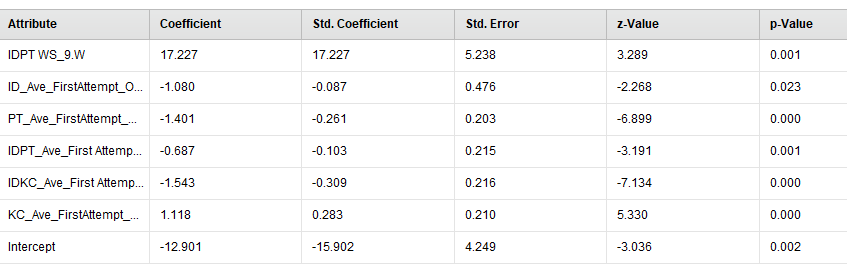
**PerformanceVector:  
accuracy: 91.12%  
ConfusionMatrix:  
True: M W  
M: 36906 3313  
W: 326 457  
AUC: 0.783 (positive class: W)  
precision: 58.37% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36906 3313  
W: 326 457  
recall: 12.12% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36906 3313  
W: 326 457**

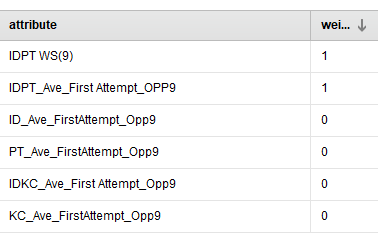
****

**PerformanceVector:  
accuracy: 91.05%  
ConfusionMatrix:  
True: M W  
M: 36886 3323  
W: 346 447  
AUC: 0.784 (positive class: W)  
precision: 56.37% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36886 3323  
W: 346 447  
recall: 11.86% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36886 3323  
W: 346 447**

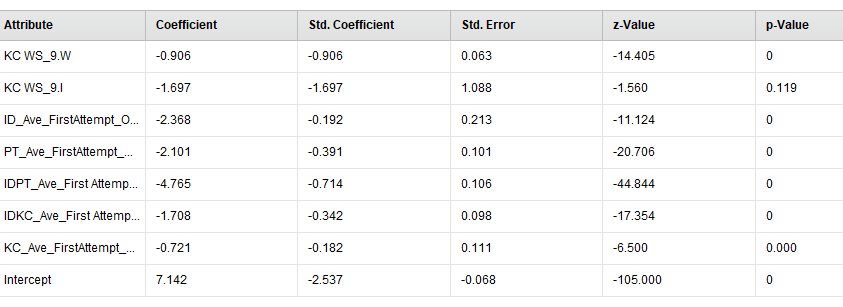
****

**PerformanceVector:  
accuracy: 98.09%  
ConfusionMatrix:  
True: M W  
M: 36449 2  
W: 783 3768  
AUC: 0.992 (positive class: W)  
precision: 82.79% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36449 2  
W: 783 3768  
recall: 99.95% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36449 2  
W: 783 3768**

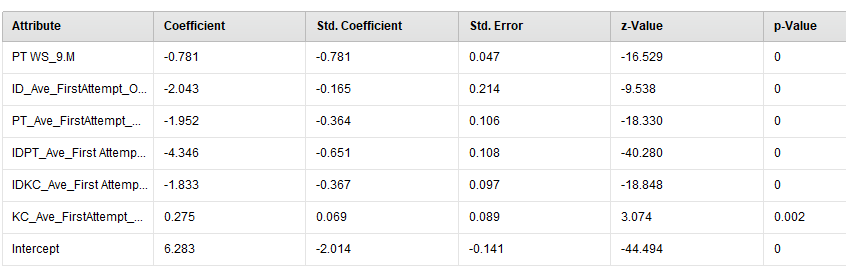
****

****

**PerformanceVector:  
accuracy: 91.18%  
ConfusionMatrix:  
True: M W  
M: 36889 3272  
W: 343 498  
AUC: 0.786 (positive class: W)  
precision: 59.22% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36889 3272  
W: 343 498  
recall: 13.21% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36889 3272  
W: 343 498**

****

**PerformanceVector:  
accuracy: 91.30%  
ConfusionMatrix:  
True: M W  
M: 36870 3204  
W: 362 566  
AUC: 0.794 (positive class: W)  
precision: 60.99% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36870 3204  
W: 362 566  
recall: 15.01% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 36870 3204  
W: 362 566**

****

**------------------------------------------------**

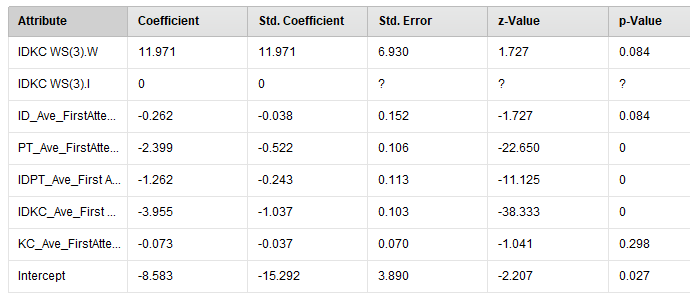
**Feb 5.**

**Using ‘remove unused variables’ and run’ idkc ws 10 - idkc ws 3,4,5,6,7,8,9’ again.**

**Idkc 10**

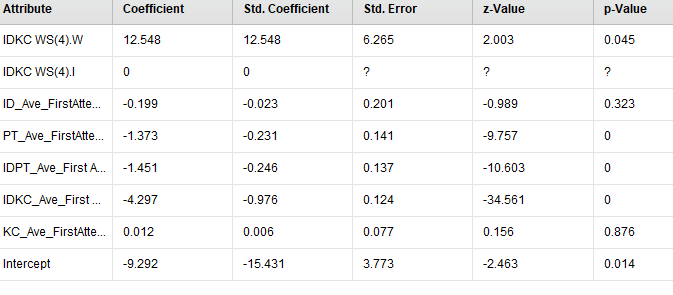
* **5 covariate + idkc ws opp3**

**PerformanceVector:  
accuracy: 94.91%  
ConfusionMatrix:  
True: M W  
M: 34491 1546  
W: 331 519  
AUC: 0.927 (positive class: W)  
precision: 61.06% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34491 1546  
W: 331 519  
recall: 25.13% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34491 1546  
W: 331 519**

****

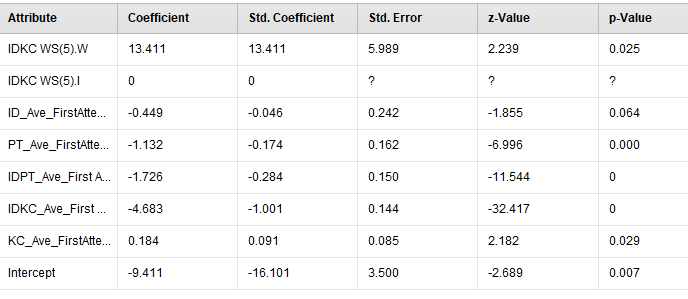
* **5 covariate + idkc ws opp4**

**PerformanceVector:  
accuracy: 95.19%  
ConfusionMatrix:  
True: M W  
M: 34542 1495  
W: 280 570  
AUC: 0.944 (positive class: W)  
precision: 67.06% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34542 1495  
W: 280 570  
recall: 27.60% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34542 1495  
W: 280 570**

****

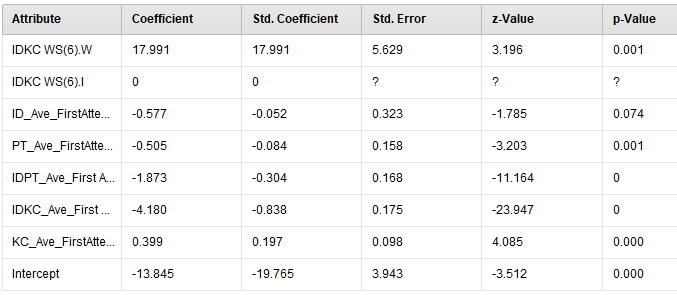
* **5 covariate + idkc ws opp5**

**PerformanceVector:  
accuracy: 95.38%  
ConfusionMatrix:  
True: M W  
M: 34459 1340  
W: 363 725  
AUC: 0.960 (positive class: W)  
precision: 66.64% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34459 1340  
W: 363 725  
recall: 35.11% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34459 1340  
W: 363 725**

****

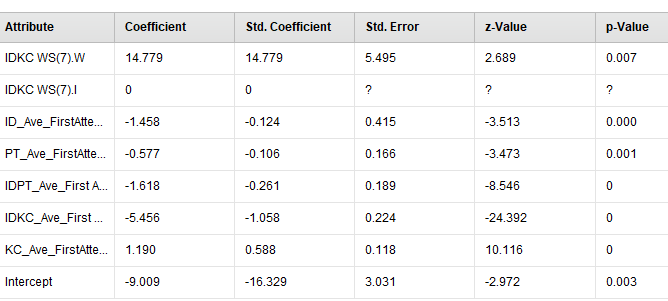
* **5 covariate + idkc ws opp6**

**PerformanceVector:  
accuracy: 96.20%  
ConfusionMatrix:  
True: M W  
M: 34386 965  
W: 436 1100  
AUC: 0.977 (positive class: W)  
precision: 71.61% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34386 965  
W: 436 1100  
recall: 53.27% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34386 965  
W: 436 1100**

****

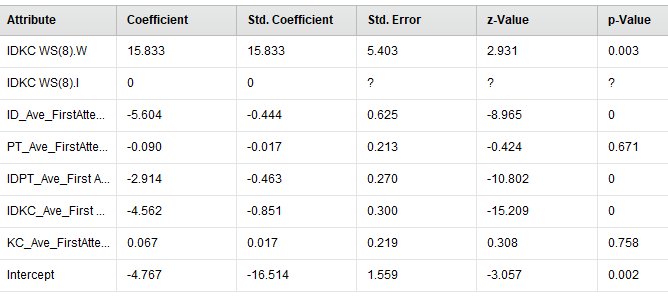
* **5 covariate + idkc ws opp7**

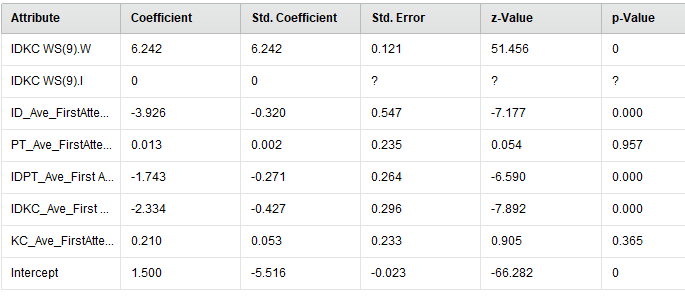
**PerformanceVector:  
accuracy: 96.95%  
ConfusionMatrix:  
True: M W  
M: 34318 620  
W: 504 1445  
AUC: 0.987 (positive class: W)  
precision: 74.14% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34318 620  
W: 504 1445  
recall: 69.98% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34318 620  
W: 504 1445**

****

* **5 covariate + idkc ws opp8**

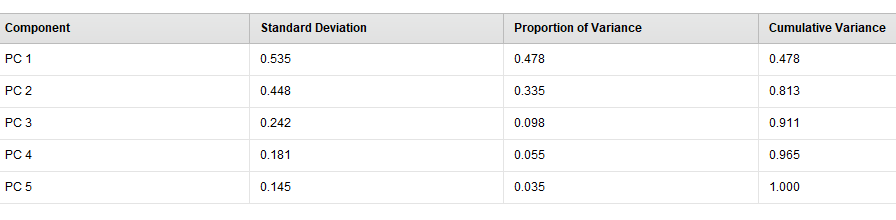
**PerformanceVector:  
accuracy: 98.20%  
ConfusionMatrix:  
True: M W  
M: 34289 130  
W: 533 1935  
AUC: 0.995 (positive class: W)  
precision: 78.40% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34289 130  
W: 533 1935  
recall: 93.70% (positive class: W)  
ConfusionMatrix:  
True: M W  
M: 34289 130  
W: 533 1935**

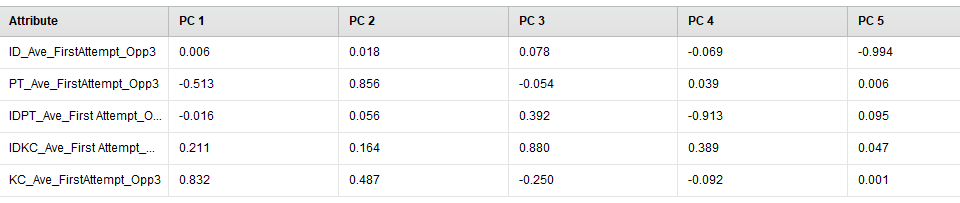
****

* **5 covariate + idkc ws opp9**
* **PerformanceVector:  
  accuracy: 98.96%  
  ConfusionMatrix:  
  True: M W  
  M: 34440 0  
  W: 382 2065  
  AUC: 0.998 (positive class: W)  
  precision: 84.39% (positive class: W)  
  ConfusionMatrix:  
  True: M W  
  M: 34440 0  
  W: 382 2065  
  recall: 100.00% (positive class: W)  
  ConfusionMatrix:  
  True: M W  
  M: 34440 0  
  W: 382 2065**
* ****

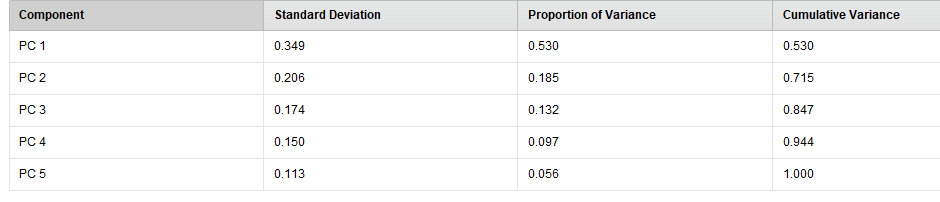
**PCA**

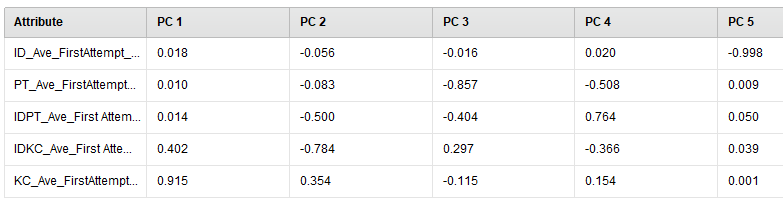
**3**

****

****

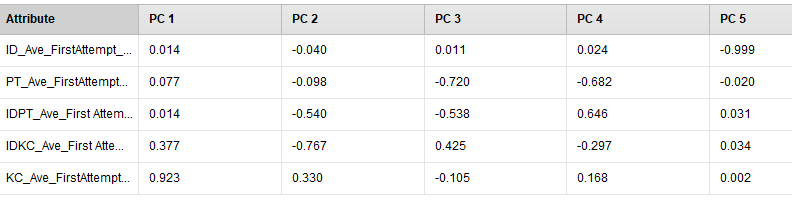
**4**

****

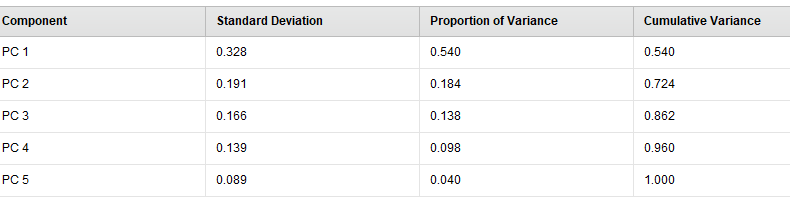
****

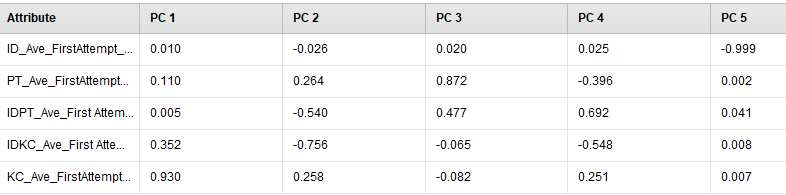
**5**

****

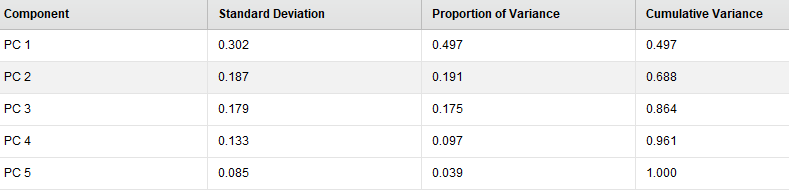
****

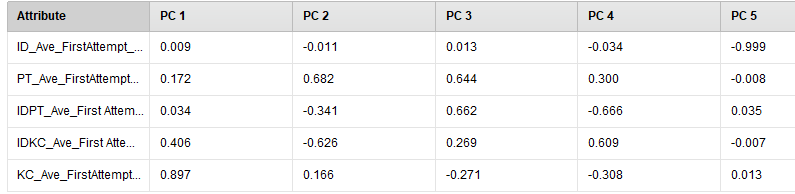
**6**

****

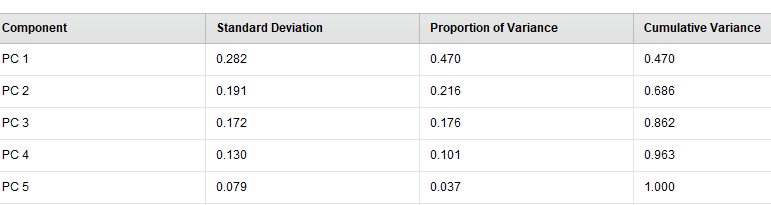
****

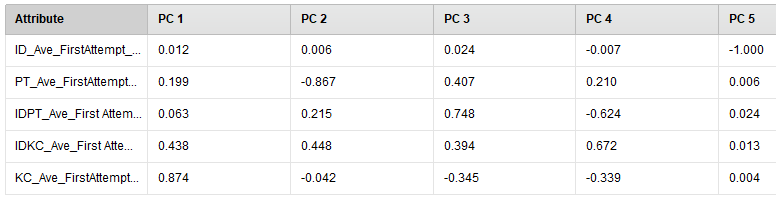
**7**

****

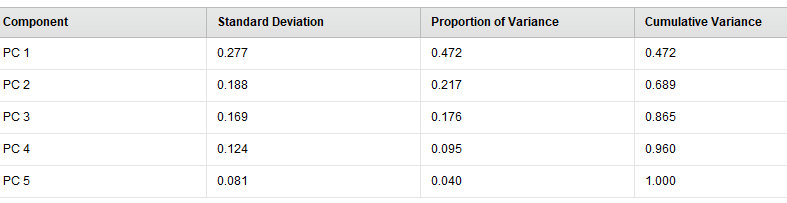
****

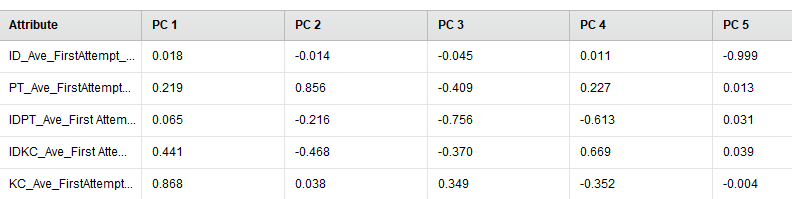
**8**

****

****

**9**

****

****

Vif

Call:

glm(formula = delete\_idkc\_I$`IDKC WS\_10` ~ delete\_idkc\_I$`iD\_Ave\_ First Attempt` +

delete\_idkc\_I$`KC\_Ave\_ First Attempt` + delete\_idkc\_I$`IDKC\_Ave\_ First Attempt` +

delete\_idkc\_I$`IDPT\_Ave\_ First Attempt` + delete\_idkc\_I$`PT\_Ave\_ First Attempt`,

family = "binomial")

Deviance Residuals:

Min 1Q Median 3Q Max

-2.409e-06 -2.409e-06 -2.409e-06 -2.409e-06 -2.409e-06

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.657e+01 4.203e+04 -0.001 0.999

delete\_idkc\_I$`iD\_Ave\_ First Attempt` -2.568e-12 4.217e+04 0.000 1.000

delete\_idkc\_I$`KC\_Ave\_ First Attempt` -8.229e-13 2.553e+04 0.000 1.000

delete\_idkc\_I$`IDKC\_Ave\_ First Attempt` 2.292e-13 1.853e+04 0.000 1.000

delete\_idkc\_I$`IDPT\_Ave\_ First Attempt` -2.006e-12 2.333e+04 0.000 1.000

delete\_idkc\_I$`PT\_Ave\_ First Attempt` 2.630e-12 3.699e+04 0.000 1.000

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 0.00e+00 on 36886 degrees of freedom

Residual deviance: 2.14e-07 on 36881 degrees of freedom

AIC: 12

Number of Fisher Scoring iterations: 25

> vif(result)

delete\_idkc\_I$`iD\_Ave\_ First Attempt` delete\_idkc\_I$`KC\_Ave\_ First Attempt`

1.442145 2.465694

delete\_idkc\_I$`IDKC\_Ave\_ First Attempt` delete\_idkc\_I$`IDPT\_Ave\_ First Attempt`

2.676736 2.214297

delete\_idkc\_I$`PT\_Ave\_ First Attempt`

1.809601

**\**

**Gradient Boosted trees.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Gradient boosted trees.** | **.** | **opp3** | **opp4** | **opp5** | **opp6** | **opp7** | **opp8** | **opp9** |
|  | **accuracy** | **94.03** | **94.75** | **95.3** | **96.78** | **97.68** | **98.84** | **99.48** |
|  | **auc** | **0.936** | **0.961** | **0.973** | **0.986** | **0.993** | **0.998** | **1** |
|  | **precision** | **47.05** | **52.59** | **56.25** | **69.73** | **77.98** | **85.14** | **94.66** |
|  | **recall** | **53.22** | **63.83** | **72.15** | **74.96** | **81.65** | **96.03** | **96.22** |

**OPP6**

**Model Metrics Type: Binomial  
 Description: N/A  
 model id: rm-h2o-model-gradient\_boosted\_trees-281895  
 frame id: rm-h2o-frame-gradient\_boosted\_trees-147211  
 MSE: 0.022498379  
 R^2: 0.57427996  
 AUC: 0.9857248  
 logloss: 0.07599165  
 CM: Confusion Matrix (vertical: actual; across: predicted):  
 M W Error Rate  
 M 34150 672 0.0193 = 672 / 34,822  
 W 513 1552 0.2484 = 513 / 2,065  
Totals 34663 2224 0.0321 = 1,185 / 36,887  
Gains/Lift Table (Avg response rate: 5.60 %):  
 Group Cumulative Data Fraction Lower Threshold Lift Cumulative Lift Response Rate Cumulative Response Rate Capture Rate Cumulative Capture Rate Gain Cumulative Gain  
 1 0.01003063 0.763765 17.428450 17.428450 0.975676 0.975676 0.174818 0.174818 1642.844971 1642.844971  
 2 0.02095589 0.696917 15.824006 16.591981 0.885856 0.928849 0.172881 0.347700 1482.400639 1559.198055  
 3 0.03014612 0.641736 14.121745 15.838914 0.790560 0.886691 0.129782 0.477482 1312.174534 1483.891424  
 4 0.04004121 0.433775 9.885799 14.367765 0.553425 0.804333 0.097821 0.575303 888.579920 1336.776530  
 5 0.05072248 0.344232 9.384851 13.318450 0.525381 0.745591 0.100242 0.675545 838.485146 1231.845047  
 6 0.10071299 0.203647 5.066337 9.222381 0.283623 0.516285 0.253269 0.928814 406.633674 822.238109  
 7 1.00000000 0.007557 0.079159 1.000000 0.004431 0.055982 0.071186 1.000000 -92.084124 0.000000  
Variable Importances:  
 Variable Relative Importance Scaled Importance Percentage  
 IDKC WS\_6 3160.225098 1.000000 0.634715  
IDKC\_Ave\_First Attempt\_Opp6 1073.151489 0.339581 0.215537  
IDPT\_Ave\_First Attempt\_OPP6 260.104431 0.082306 0.052241  
 KC\_Ave\_FirstAttempt\_Opp6 250.729050 0.079339 0.050358  
 PT\_Ave\_FirstAttempt\_Opp6 161.178391 0.051002 0.032372  
 ID\_Ave\_FirstAttempt\_Opp6 73.577866 0.023282 0.014778  
Model Summary:  
 Number of Trees Model Size in Bytes Min. Depth Max. Depth Mean Depth Min. Leaves Max. Leaves Mean Leaves  
 20 5017 5 5 5.00000 13 17 16.35000  
Scoring History:  
 Timestamp Duration Number of Trees Training MSE Training LogLoss Training AUC Training Lift Training Classification Error  
 2018-02-23 08:23:38 0.015 sec 0 0.05285 0.21577 0.50000 1.00000 0.94402  
 2018-02-23 08:23:38 0.253 sec 1 0.04406 0.16595 0.98073 16.36452 0.03855  
 2018-02-23 08:23:38 0.384 sec 2 0.03992 0.14876 0.98171 16.43991 0.03578  
 2018-02-23 08:23:38 0.466 sec 3 0.03675 0.13659 0.98223 16.87579 0.03633  
 2018-02-23 08:23:38 0.539 sec 4 0.03414 0.12685 0.98280 16.90845 0.03603  
 2018-02-23 08:23:39 0.656 sec 5 0.03206 0.11903 0.98329 16.95548 0.03592  
 2018-02-23 08:23:39 0.725 sec 6 0.03039 0.11264 0.98350 16.97680 0.03573  
 2018-02-23 08:23:39 0.779 sec 7 0.02906 0.10735 0.98360 16.96745 0.03573  
 2018-02-23 08:23:39 0.829 sec 8 0.02789 0.10259 0.98401 16.97215 0.03573  
 2018-02-23 08:23:39 0.886 sec 9 0.02696 0.09858 0.98424 16.84911 0.03505  
 2018-02-23 08:23:39 0.942 sec 10 0.02614 0.09506 0.98444 16.94691 0.03364  
 2018-02-23 08:23:39 1.008 sec 11 0.02554 0.09216 0.98445 16.83420 0.03435  
 2018-02-23 08:23:39 1.073 sec 12 0.02496 0.08943 0.98467 16.99159 0.03424  
 2018-02-23 08:23:39 1.145 sec 13 0.02449 0.08707 0.98481 16.85456 0.03364  
 2018-02-23 08:23:39 1.202 sec 14 0.02405 0.08490 0.98492 16.85184 0.03367  
 2018-02-23 08:23:39 1.263 sec 15 0.02373 0.08309 0.98507 17.42962 0.03367  
 2018-02-23 08:23:39 1.325 sec 16 0.02340 0.08131 0.98516 17.43079 0.03275  
 2018-02-23 08:23:39 1.386 sec 17 0.02316 0.07986 0.98522 17.52501 0.03275  
 2018-02-23 08:23:39 1.453 sec 18 0.02288 0.07838 0.98542 17.52772 0.03218  
 2018-02-23 08:23:39 1.526 sec 19 0.02264 0.07699 0.98567 17.62221 0.03213  
 2018-02-23 08:23:40 1.599 sec 20 0.02250 0.07599 0.98572 17.42845 0.03213**

### **그래디언트 부스팅 회귀 트리**

**그래디언트 부스팅 회귀 트리는 여러 개의 결정 트리를 묶어 강력한 모델을 만드는 또 다른 앙상블 방법입니다. 이름이 회귀지만 이 모델은 회귀와 분류 모두에 사용할 수 있습니다.** [**5**](https://tensorflow.blog/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D/2-3-6-%EA%B2%B0%EC%A0%95-%ED%8A%B8%EB%A6%AC%EC%9D%98-%EC%95%99%EC%83%81%EB%B8%94/#5) **랜덤 포레스트와는 달리 그래디언트 부스팅은 이전 트리의 오차를 보완하는 방식으로 순차적으로 트리를 만듭니다. 기본적으로 그래디언트 부스팅 회귀 트리에는 무작위성이 없습니다. 대신 강력한 사전 가지치기가 사용됩니다. 그래디언트 부스팅 트리는 보통 하나에서 다섯 정도의 깊지 않은 트리를 사용하므로 메모리를 적게 사용하고 예측도 빠릅니다. 그래디언트 부스팅의 근본 아이디어는 이런 얕은 트리 같은 간단한 모델(약한 학습기weak learner라고도 합니다)을 많이 연결하는 것입니다. 각각의 트리는 데이터의 일부에 대해서만 예측을 잘 수행할 수 있어서 트리가 많이 추가될수록 성능이 좋아집니다. [6](https://tensorflow.blog/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D/2-3-6-%EA%B2%B0%EC%A0%95-%ED%8A%B8%EB%A6%AC%EC%9D%98-%EC%95%99%EC%83%81%EB%B8%94/#6)**

**그래디언트 부스팅 트리는 머신러닝 경연 대회에서 우승을 많이 차지하였고 업계에서도 널리 사용합니다. 랜덤 포레스트보다는 매개변수 설정에 조금 더 민감하지만 잘 조정하면 더 높은 정확도를 제공해줍니다.**

**앙상블 방식에 있는 사전 가지치기나 트리 개수 외에도 그래디언트 부스팅에서 중요한 매개변수는 이전 트리의 오차를 얼마나 강하게 보정할 것인지를 제어하는 learning\_rate입니다. 학습률이 크면 트리는 보정을 강하게 하기 때문에 복잡한 모델을 만듭니다. n\_estimators 값을 키우면 앙상블에 트리가 더 많이 추가되어 모델의 복잡도가 커지고 훈련 세트에서의 실수를 바로잡을 기회가 더 많아집니다.**

**아래는 유방암 데이터셋을 이용해 GradientBoostingClassifier를 사용한 예입니다. 기본값인 깊이가 3인 트리 100개와 학습률 0.1을 사용하였습니다.**

**Feb 27.**

**In the middle of writing EDM paper, Noboru pointed out my features are kind of controversial.**

1. **M/W at each opportunity : if students already mastered on their 3rd opportunity, why do we need to train the data with this feature? The output should be ‘M’. However, noboru said it can be possible if there’s no negative example of detecting wheel-spinning of this feature (already mastery but prediction is W, so M->W).**

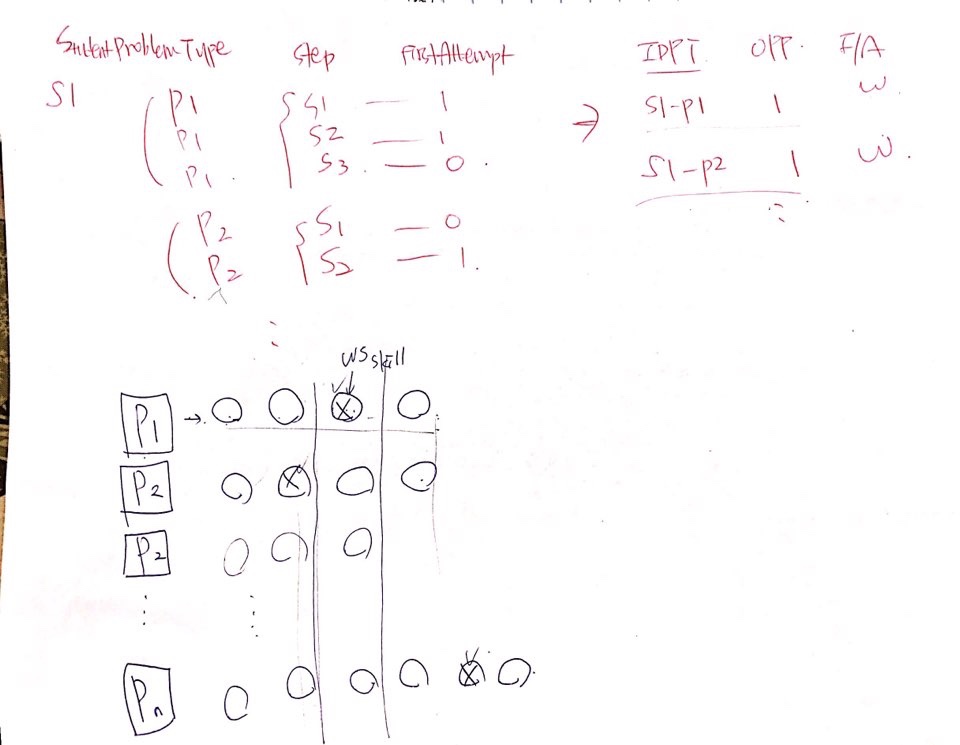
**I need to check this : there are 4 M->W out of 36,887 in GBT model on 3rd opportunity.**

**None in logistic regression.**

**Regardless of this fact, this can be a feature. Because it’s not cheating. We give the information about clear color on that opportunity (M or W) to predict final W. Some of W can be M in the final 10th opportunity, and others remain W in the 10th opportunity.**

1. **IDPT average correct first attempt : noboru liked the concept of ‘Problem type’ . However, my coding is not correct because my IDPT has the same number of opportunity as the sum of constituent step(or kc) opportunity. The crown jewelry (핵심 그림?) ..**

****

****

**IDPT up to each opportunity can be understood as general math competency of each student. Because, regardless of whether the problem type includes a wheel-spinning skill, we used IDPT (P1~Pn) average correct first attempt as one of our features.**

**3) when you look at the summary file, comparing the model using 5 average first attempt vs the model using 5 average first attempt + M/W at each opportunity, recall rate has huge difference.**

**4) maybe I can calculate suedo -R^**

**5) do I need to report comparing 5 average first attempt vs 5+M/W?**