FACULTAD DE INGENIERÍA Y CIENCIAS

UNIVERSIDAD ADOLFO IBÁÑEZ

A Stacked Generalization Ensemble Model for Help Desk Ticket Assignment

International Conference of the Chilean Computer Science Society (CCSS)

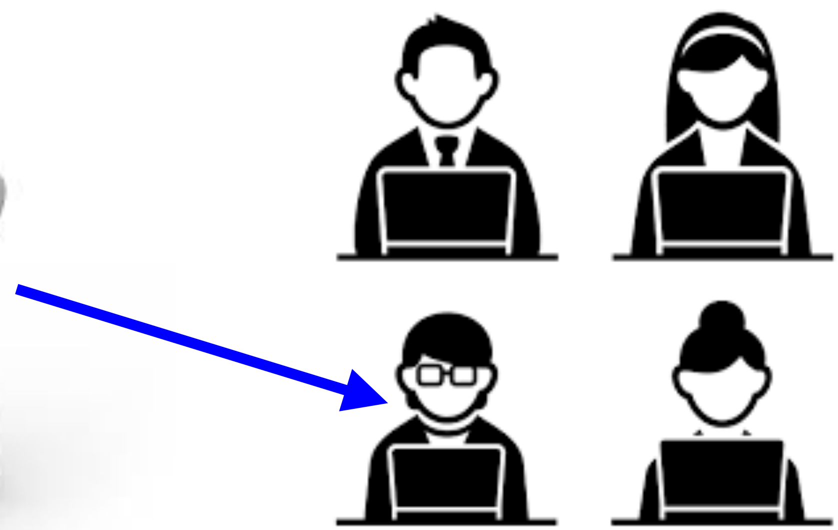
Sebastián Moreno
Wilfredo Yushimito
Sebastián Hughes (Magister en Ciencias de la Ingeniería, UAI)
Facultad de Ingeniería y Ciencias
Universidad Adolfo Ibáñez

Sebastián Moreno, Universidad Adolfo Ibáñez

- Ph.D. in computer Science, Purdue University, 2014.
- Associate professor (2022, at UAI since 2015).
- Head of Master of Science in Data Science (2022).
- Associate director postgraduate academic programs (2021)
  Magister en Ciencias de la Ingeniería
  Master of Science in Data Science
  Doctorado en Data Science
- More than 30 conference papers (KDD, ICDM, WWW) and WOS journals.
- Program committee in more than 40 conference including NeurIPS, KDD, WWW, ICDM, and SCCC.
- Research interest: Machine learning, neural networks, relational learning, statistical network analysis.

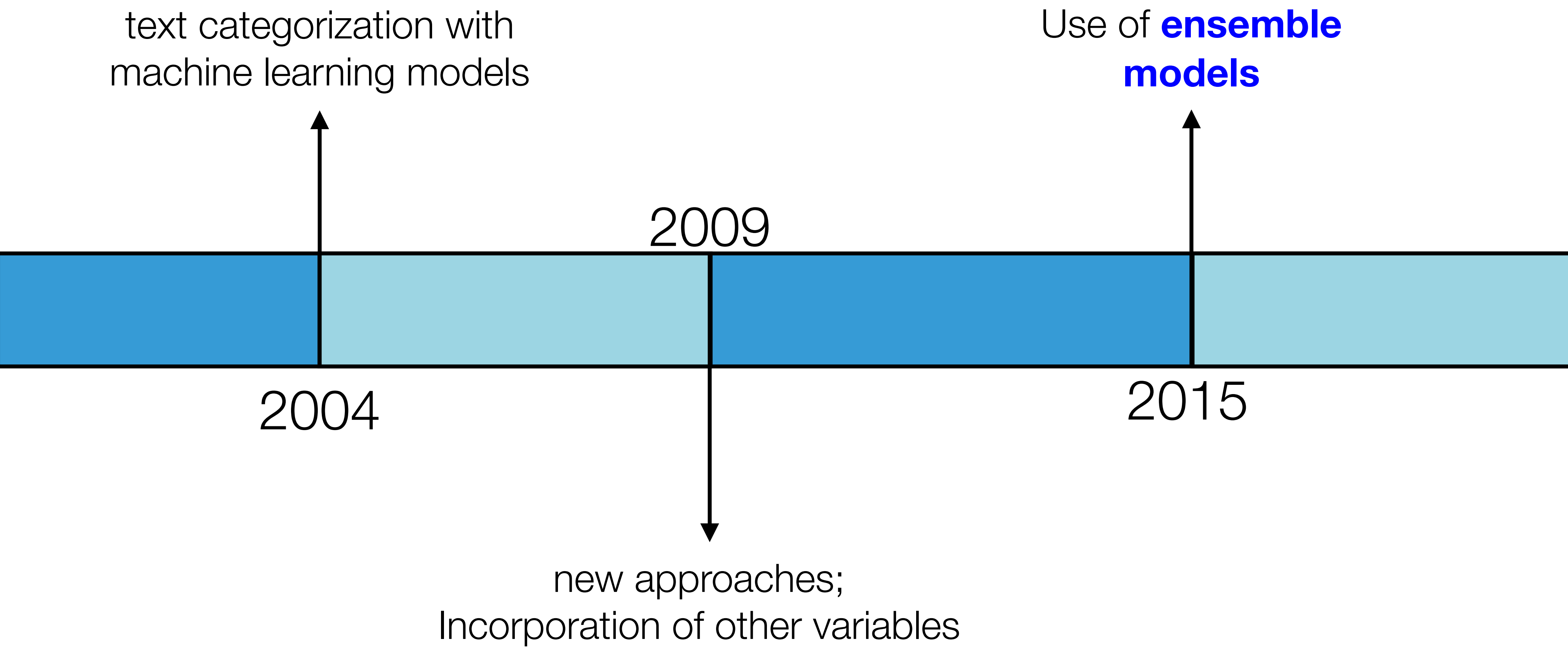
Problem definition

- Technological advances allow automating processes that previously required the direct intervention of a technician in case of an incident.
- In software projects, the bugs (errors in the code) must be analyzed and assigned to an expert for their resolution.
- Large software projects can have 50-60 bugs per day. If each bug is assigned manually in 5 minutes => 3.3 to 5.0 hours dedicated in the assignment task.
- **How can we assign the tickets automatically?**



State of the art

- Current work assumes that most of the data has several variables and text.

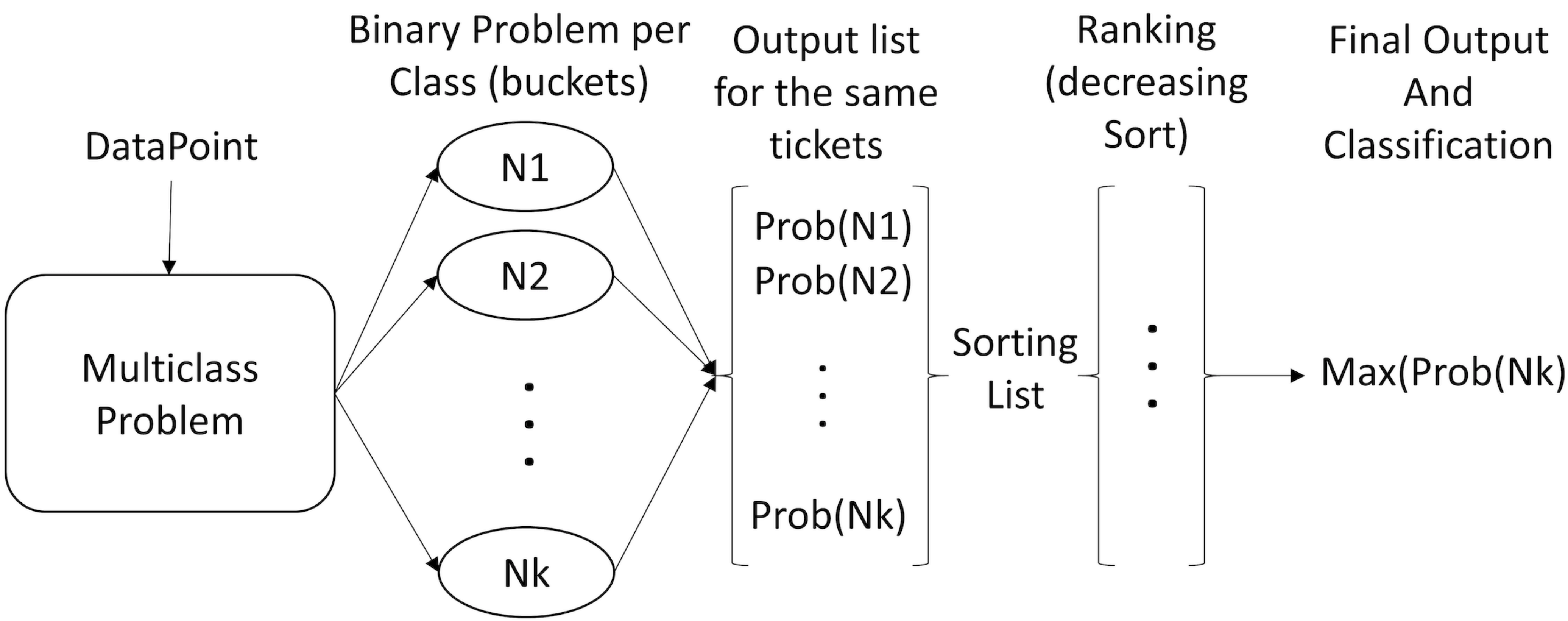


Dataset

- Originally, we had 3,726 tickets between 2016 and 2017:
  - 12 attributes
  - 10 programmers
  - No text data was available
- After cleaning and generation process, we had 1,051 tickets with 14 variables:
  - Origin: nominal (16 values => hospitals)
  - Module: nominal (14 values => category of the ticket)
  - Deadline: binary (1 for ticket with time limit)
  - Occupation: (numerical, 10 variables with the number of tickets that each programmer is currently working)
  - Class: nominal (10 values => programmers)

One Versus Rest (OVR) “ensemble model”

- The One Versus Rest (OVR) approach uses ensembles for k classes:
    - The data is repeated k times, and k models are learned (one per class).
    - Given a data point, k predictions are applied (one per model).
    - The final prediction corresponds to the class with highest probability.



One Versus Rest (OVR) “ensemble model”

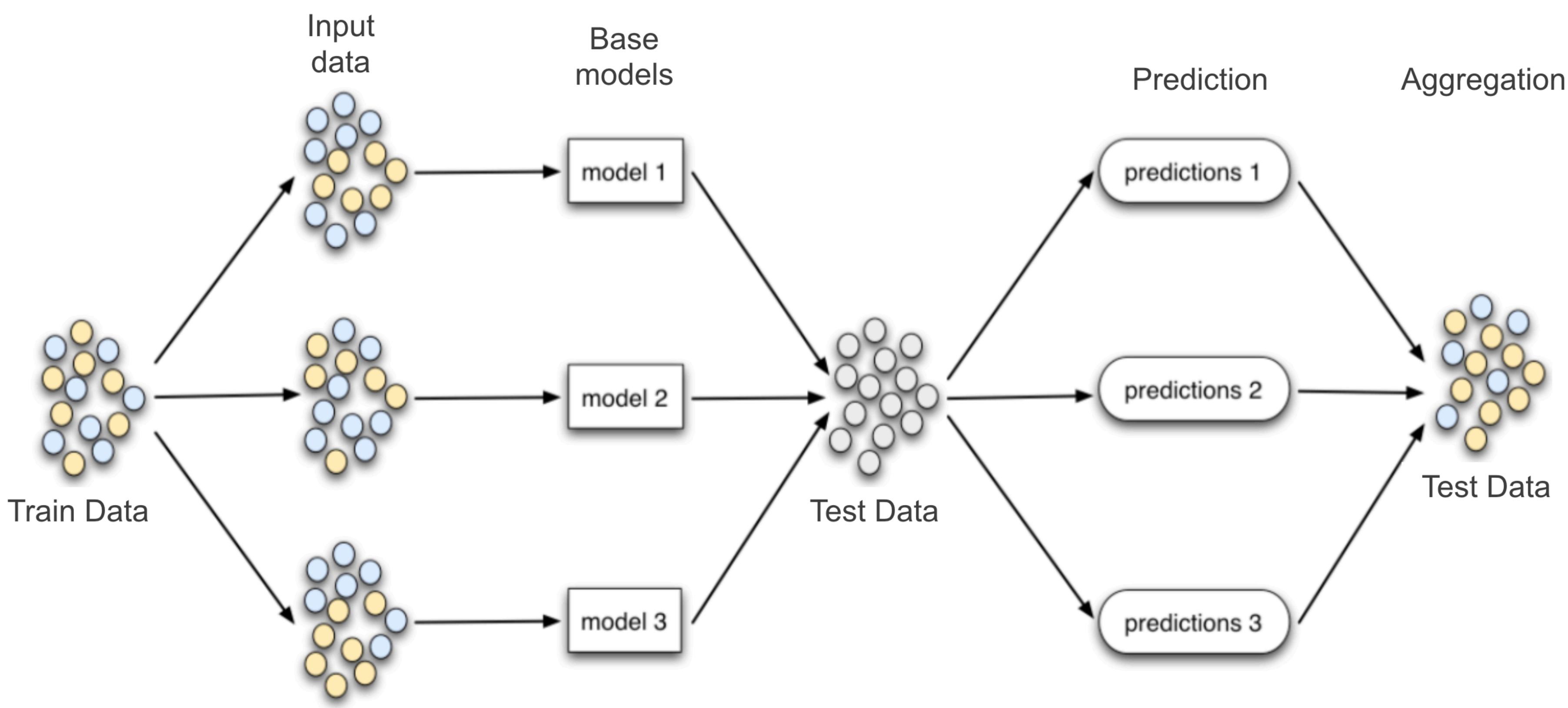
- The One Versus Rest (OVR) approach uses ensembles for k classes:
  - The data is repeated k times, and k models are learned (one per class).
  - Given a data point, k predictions are applied (one per model).
  - The final prediction corresponds to the class with highest probability.
- Example, probabilities for a datapoint, from a dataset with 5 classes.

	class 1	class 2	class 3	class 4	class 5
Ensemble	0.93	0.96	0.45	0.04	0.30

- **Class 2 is selected, the class with highest probability.**

Ensemble models for multiclass prediction

- Typical approach for multiclass problems using ensembles for k classes:
  - The data is repeated m times, and m multiclass-models are learned.
  - Given a data point, m*k predictions are applied (k predictions per model).
  - The final prediction corresponds to an aggregation of the values.



Ensemble models for multiclass prediction

- Example, probabilities using 4 models, from a dataset with 5 classes.

	class 1	class 2	class 3	class 4	class 5
Multiclass 1	0.31	0.32	0.25	0.02	0.10
Multiclass 2	0.44	0.45	0.01	0.02	0.08
Multiclass 3	0.40	0.09	0.05	0.38	0.08
Multiclass 4	0.01	0.03	0.02	0.02	0.92

What class should we predict?

Class 1 is high in 3 models, and wins in one.

Class 2 wins in 2 models.

Classes 3 and 4 should not be selected.

Class 5 has the highest probability of all.

Ensemble models for multiclass prediction

- Ensemble of multiclass models could produce a high diversity in their answer.

	class 1	class 2	class 3	class 4	class 5
Multiclass 1	0.31	0.32	0.25	0.02	0.10
Multiclass 2	0.44	0.45	0.01	0.02	0.08
Multiclass 3	0.40	0.09	0.05	0.38	0.08
Multiclass 4	0.01	0.03	0.02	0.02	0.92

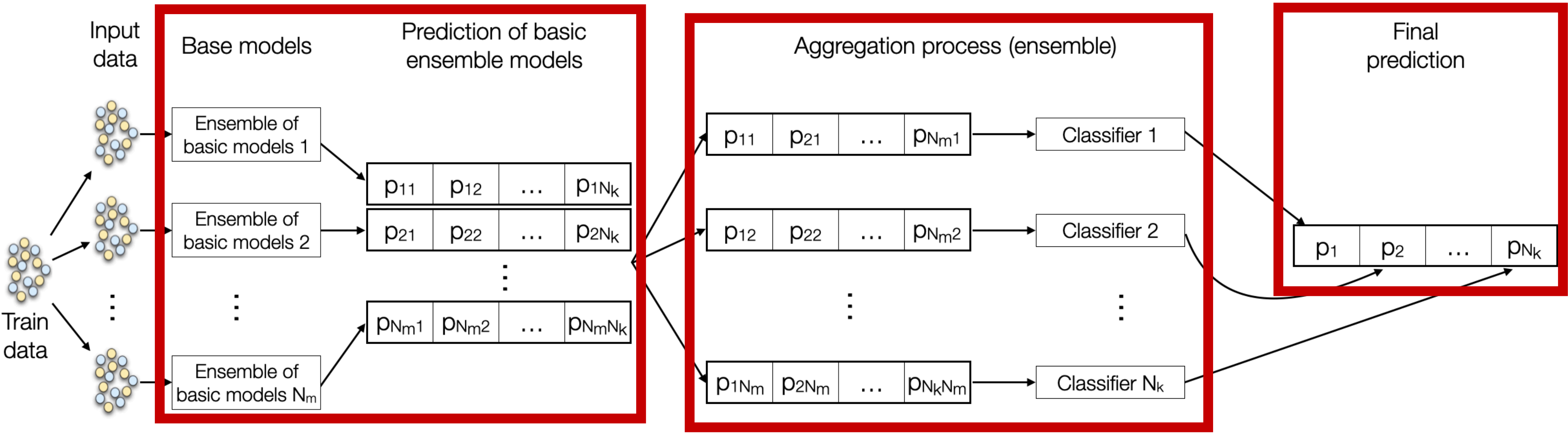
- One versus rest approach give us less information.

	class 1	class 2	class 3	class 4	class 5
Ensemble	0.93	0.96	0.45	0.04	0.30

How can we combine these approaches?

Stacked generalization basic ensemble models

- We propose a stacked generalization process with two levels (the first and second levels are ensemble models instead of basic models).
- We change the basic models for ensemble models (obtaining k probabilities).
- New classifier learns to trust the prediction of the ensembles.
- The final prediction corresponds to the class with highest probability.



Stacked generalization basic ensemble models, example

- Example, probabilities using 4 models, with a dataset with 5 classes.
- Each ensemble learned a probability for each class.

	class 1	class 2	class 3	class 4	class 5
Ensemble 1	0.93	0.96	0.45	0.04	0.30
Ensemble 2	0.94	0.97	0.01	0.02	0.20
Ensemble 3	0.98	0.12	0.05	0.89	0.22
Ensemble 4	0.01	0.03	0.02	0.02	0.92

Stacked generalization basic ensemble models, example

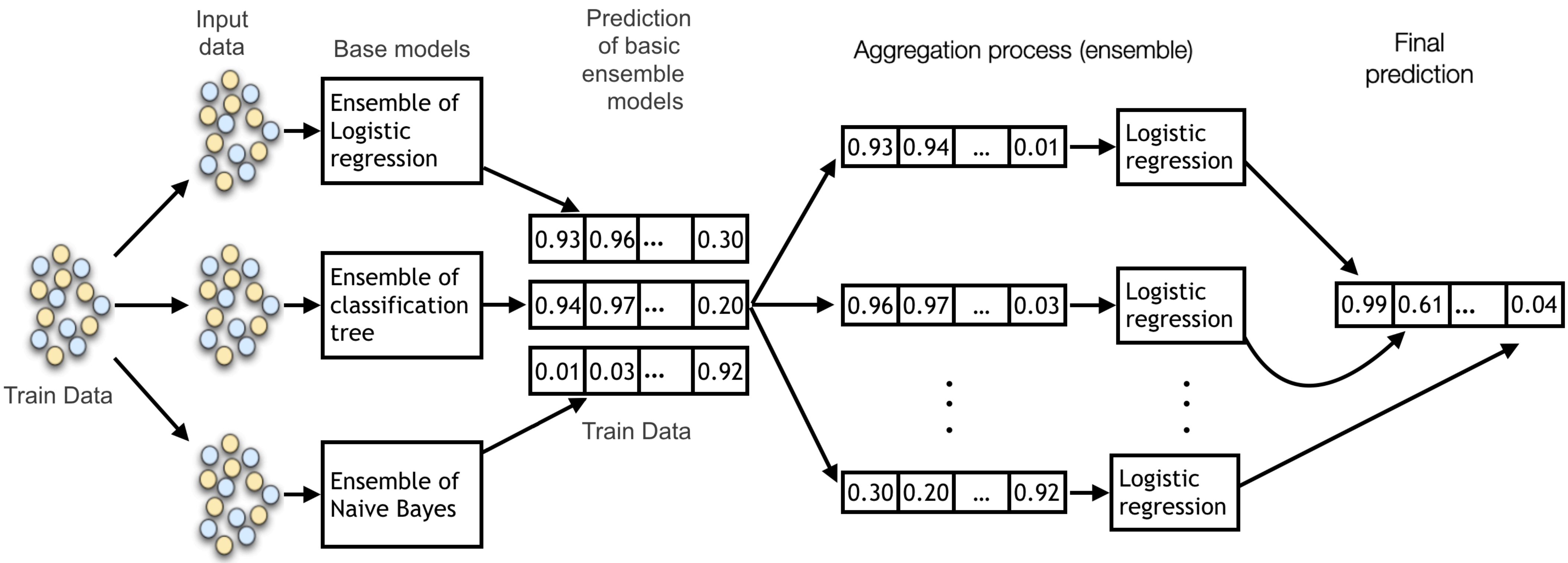
- We generate a “new dataset” based on the predictions.

- We use another ensemble model for the aggregation step, i.e., a new model learns to make the final prediction based on probabilities.

	Ens M.	Ens M. 2	Ens M. 3	Ens M. 4	Est Prob.
class 1	0.93	0.94	0.98	0.01	**0.99**
class 2	0.96	0.97	0.12	0.03	0.61
class 3	0.45	0.01	0.05	0.02	0.20
class 4	0.04	0.02	0.89	0.02	0.05
class 5	0.30	0.20	0.22	0.92	0.04

Stacked generalization basic ensemble models (SGBEM), final model

- For this paper, we use 3 ensembles (Logistic regression, classification tree, and Naive Bayes), and logistic regression for the aggregation process.



Methodology

- We compare SGBEM, our proposed ensemble, against:
  - Random baselines.
  - Multilabel version of basic individual models (KNN, Classification Tree, Logistic Regression, Naive Bayes, SVM).
  - One vs. rest ensemble versions of these basic models.
  - basic Stacked Generalization.
  - Random Forest (random selection of the variables).
  - “Facebook”, an ensemble to predict clicks.
  - XGBoost.

Results, accuracy

- 50 repetitions (70% training and **30% test**).
- Random models perform very bad (complex problem).
- Low and similar performance for basic models.
- Ensembles perform similar to basic models.
- SGBEM obtains, in average, the highest accuracy, and it is statistically significant.

TABLE IV: Accuracy for all models.

Model	Top 1	Top 2	Top 3
SGBEM	**67.0% ± 2.6%**	**80.6% ± 2.5%**	**86.7% ± 1.7%**
Random			
Simple	9.6% ± 1.5%	19.6% ± 2.4%	29.9% ± 2.7%
Weights	28.1% ± 2.5%	43.5% ± 3.0%	53.8% ± 3.2%
KNN	60.0% ± 2.2%	72.2% ± 1.7%	80.2% ± 1.9%
MLR	55.8% ± 2.4%	73.9% ± 2.2%	84.7% ± 1.7%
CT	58.2% ± 2.2%	71.7% ± 2.5%	81.6% ± 2.2%
NB	49.9% ± 2.1%	69.7% ± 2.6%	81.6% ± 2.4%
MLR-NN	56.2% ± 2.4%	70.3% ± 2.0%	80.3% ± 2.0%
SVM	55.4% ± 2.1%	72.0% ± 2.1%	81.4% ± 2.0%
Ensemble			
KNN	54.5% ± 2.3%	68.9% ± 2.2%	78.9% ±2.0%
LR	57.5% ± 2.1%	71.7% ± 2.2%	82.3% ± 1.9%
CT	56.9% ± 2.2%	68.2% ± 2.4%	77.0% ± 2.6%
NB	56.2% ± 2.1%	72.0% ± 2.2%	82.2% ± 1.8%
MLR	56.3% ± 2.3%	70.6% ± 2.4%	80.8% ± 2.2%
SVM	54.7% ± 2.9%	66.5% ± 3.0%	77.1% ± 2.8%
bSG	47.7% ± 15.7%	68.6% ± 10.9%	78.1% ± 13.0%
RF	59.0% ± 2.2%	74.7% ± 2.4%	85.2% ± 2.0%
Facebook	54.6% ± 3.4%	58.0% ± 3.3%	62.4% ± 3.8%
XGBoost	58.5% ± 2.4%	74.6% ± 1.9%	84.9% ± 1.6%

Results, macro F1-score

- 50 repetitions (70% training and **30% test**).
- Similar behaviors than previous results.
- Classification tree and multiclass logistic regression models obtain good performance (but not in all cases)
- SGBEM obtains, in average, the highest macro F1-score, and it is statistically significant.

TABLE V: F1 score for all models.

Model	Top 1	Top 2	Top 3
SGBEM	**52.2%** $\pm$ **4.1%**	**68.1%** $\pm$ **4.4%**	**77.7%** $\pm$ **3.8%**
Random			
Simple	9.8% $\pm$ 1.5%	18.0% $\pm$ 2.4%	26.4% $\pm$ 2.7%
Weights	26.2% $\pm$ 2.7%	36.2% $\pm$ 2.7%	44.1% $\pm$ 3.3%
KNN	49.2% $\pm$ 3.3%	61.5% $\pm$ 2.9%	73.6% $\pm$ 3.1%
MLR	44.6% $\pm$ 3.9%	63.6% $\pm$ 3.3%	**76.2%** $\pm$ **2.7%**
CT	**53.1%** $\pm$ **3.6%**	62.6% $\pm$ 4.5%	73.5% $\pm$ 4.0%
NB	40.2% $\pm$ 3.3%	58.9% $\pm$ 3.2%	73.4% $\pm$ 3.6%
MLR-NN	39.6% $\pm$ 2.7%	58.0% $\pm$ 3.6%	72.4% $\pm$ 2.7%
SVM	36.4% $\pm$ 2.2%	54.9% $\pm$ 2.9%	71.9% $\pm$ 3.5%
Ensemble			
KNN	44.7% $\pm$ 4.0%	56.1% $\pm$ 3.1%	69.0% $\pm$ 3.4%
LR	45.7% $\pm$ 3.0%	59.3% $\pm$ 2.9%	73.3% $\pm$ 2.9%
CT	49.5% $\pm$ 3.3%	60.8% $\pm$ 4.0%	67.7% $\pm$ 4.0%
NB	45.0% $\pm$ 2.9%	60.0% $\pm$ 3.1%	73.9% $\pm$ 2.6%
MLR	38.8% $\pm$ 2.2%	56.8% $\pm$ 3.8%	71.4% $\pm$ 3.4%
SVM	37.0% $\pm$ 2.3%	49.1% $\pm$ 3.1%	64.0% $\pm$ 3.9%
bSG	27.6% $\pm$ 8.7%	61.2% $\pm$ 12.6%	74.9% $\pm$ 8.9%
RF	44.4% $\pm$ 2.7%	62.1% $\pm$ 2.9%	75.6% $\pm$ 3.3%
Facebook	45.6% $\pm$ 3.4%	48.7% $\pm$ 3.1%	53.1% $\pm$ 3.6%
XGBoost	46.2% $\pm$ 3.2%	64.7% $\pm$ 3.0%	**77.1%** $\pm$ **3.0%**

Conclusions

- We presented a new ensemble model (SGBEM) to replicate the assignment of tickets (including bugs) to programmers for help desk support in a software.
- Our approach used a stacked generalization process with two levels
    - The 1st level uses ensemble of models (CT, LT, and NB)
    - The 2nd level (aggregation step) uses a logistic regression.
- We compared our proposed model against 18 baselines including basic individual multiclass classifiers and ensemble models.
- Results showed a statistically significant improvement in 105 of the 108 cases.
- The improvement could be explained by the low variability obtained by the vectors generated in the ensemble used as basic models.

Questions?

International Conference of the Chilean Computer Science Society (JCC)