Text Mining and Machine Learning techniques for job reports analysis

Master Candidate: Luca Dal Zotto

Supervisor: Prof. Bruno Scarpa

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

15th December 2021

Università degli Studi di Padova

Contents

MIRIADE
PLAY THE CHANGE

- IT consulting company with 20 years of experience in the field of Data Engineering, Data Science and Cloud Strategy
- offices in Thiene (VI), Padova and Mestre (VE)
- clients in North-Eastern Italy, both in private and public sector

Introduction

Presentation of the internship project

Job reports

- recording of each employee's daily activities
- goal 1: monitor working processes
- goal 2: generate bills and invoices
- manually checked monthly by project managers

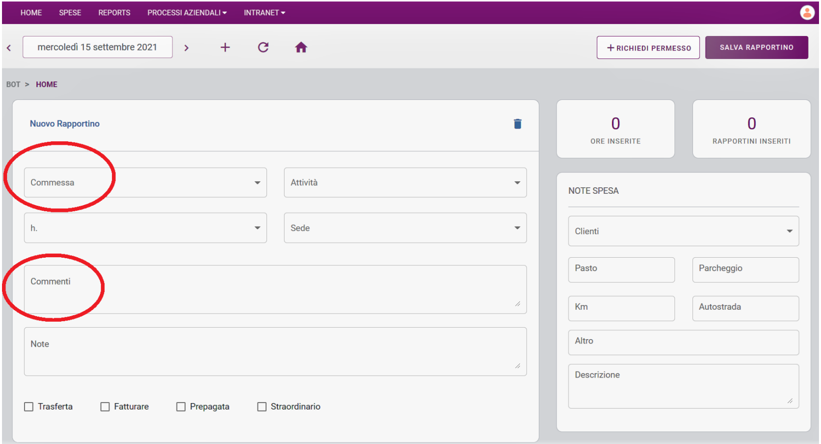
Problem

- 90 employees $\Rightarrow$ thousands of job reports to be checked
- long and tedious activity

Project's purpose

- insert a tool inside the company intranet which automatically verifies the correctness of job reports, highlighting those which are likely to be wrong
- achieved by using text mining to extract information from the descriptions provided and constructing a learning model able to classify job reports

Data Collection: company bot web portal



Data stored in a Postgres database

Data Exploration and Pre-processing

Università degli Studi di Padova

Dataset of 21,802 recordings

Variable name	Description	
tipo_update	approved or rejected job report	**binary response**
jobid	project id	identifiers
jobtaskid	activity id	
resid	resource id	
custid	customer id	
jobtaskdt	activity date	date
data_ins	insertion date	
qty	activity duration	quantitative
sede	workplace	categorical
flg_trasferta	True if at customer site	boolean flags
pay	True if used for billing	
flg_prepagato	True if prepaid activity	
flg_straordinario	True if overtime	
workdesc	activity description	**textual variable**

Imbalanced dataset

90% job reports are approved

- resampling approaches
- different classification metrics

Manual feature engineering

- resources with very different error rates
  - `area`
  - `date of recruitment`
- time related variables:
  - `month`
  - `day`
  - `delay` = insertion date - activity date

Examples:

- *modifiche ansible mongodb per integrazione icinga. Supporto replicaset/single server. Creazione automatica utente monitoring*
- *Gestione ticket SDCS-3420, Controllo VPN + accesso ai server per prolab, nicelabel e sintesi*
- *Call con Mosaico Group per possibile collaborazione su progetto IoT per inventario GDO (Unicomm)*
- *Ticket 10214850 - BI:MIR - Vendite di gruppo su tabella Oracle*
  Ticket 10334006 - BI: KAFKA - Vendite da Webshop a Me.R.sy. Wiki
  Ticket 10344351 - BI: ICT ID MGMT - DQ anagrafica dipendenti mensile
  Ticket 10302804 - BI:MIR:KONCENTRO - Aggiornamento omniadoc

Observations

- schematic language, many acronyms and ticket numbers
- interested in individual classification, no global description

Text cleaning: remove punctuation, stop words, URLs, email addresses, numbers, links and apply stemming.

Analysis of cleaned text:

- 6674 distinct stems
- low average frequency (14 samples)
- many frequent terms shared by the two classes

$\Rightarrow$ Apply minimum and maximum document frequency thresholds

Encoding approaches

BoW representation (*n*-grams)

- terms frequency matrix
- TF-IDF feature matrix

- Logistic Regression
- Support Vector Machine
- K-Nearest Neighbours
- Random Forest
- Gradient Boosting
- Feed-forward Neural Network
- Recurrent Neural Network + fully connected layers
- Recurrent Neural Network + Gradient Boosting
- PCA + Gradient Boosting

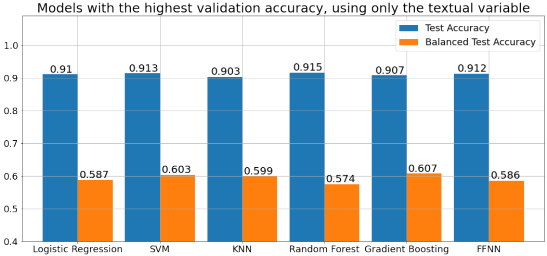
Experiments and results

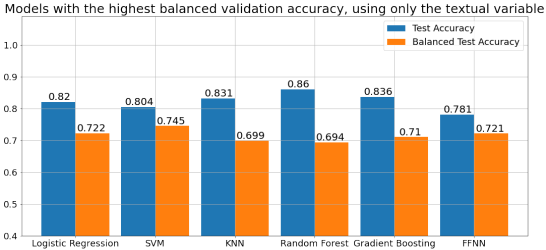
Implementation and model selection

- Consider 2 datasets:
  - only textual variable
  - textual variable + other process-related variables
- Optimization of tuning parameters using random search
- For each model, keep 2 configurations:
  - highest accuracy in the validation set
  - highest balanced accuracy in the validation set
- Evaluate in the test set

Experiments and results

Results and comparisons - only textual variable

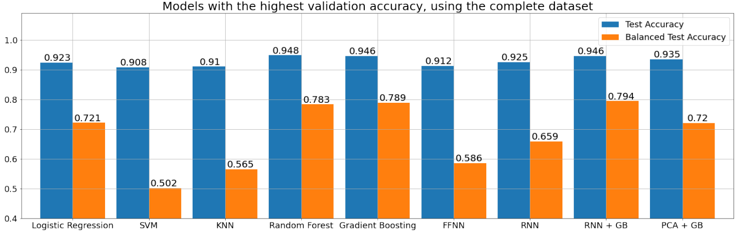


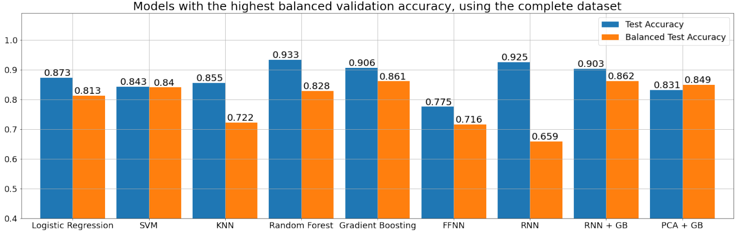


Experiments and results

Results and comparisons - complete dataset

UNIVERSITÀ DEGLI STUDI DI PADOVA





- At insertion time: need for high accuracy, few false positives and fast predictions

Models with the highest validation accuracy (complete dataset)

	LR	SVM	KNN	RF	GB	FFNN	RNN	RNN+GB	PCA+GB
accuracy	0.923	0.908	0.911	0.948	0.946	0.912	0.925	0.946	0.935
balanced accuracy	0.721	0.502	0.565	0.783	0.789	0.586	0.659	0.794	0.72
specificity	0.969	0.995	0.989	0.986	0.982	0.986	0.985	0.981	0.983
sensitivity	0.474	0.005	0.141	0.581	0.596	0.186	0.333	0.608	0.457
fitting time	295.0	195.0	0.2	44.9	388.4	581.1	202.9	452.8	461.5
predict time	0.5	102.0	100.0	3.1	1.3	15.2	16.5	1.6	3.6

- At checking time: need for high balanced accuracy, and few false negatives

Models with highest balanced validation accuracy (complete dataset)

	LR	SVM	KNN	RF	GB	FFNN	RNN	RNN + GB	PCA + GB
accuracy	0.873	0.843	0.855	0.933	0.906	0.775	0.903	0.831	0.908
balanced accuracy	0.813	0.84	0.722	0.828	0.861	0.716	0.862	0.849	0.518
specificity	0.887	0.843	0.885	0.957	0.916	0.789	0.912	0.827	0.996
sensitivity	0.739	0.836	0.558	0.7	0.806	0.643	0.811	0.871	0.040

Main results

- the inclusion of other process-related variables in addition to the textual one sensibly improves the results
- advanced forms of text encoding are not justified
- best results are obtained with tree-based models
- two gradient boosting models have been selected: one to be used at insertion time, the other to be used at checking time

Future developments

- larger dataset may justify more complex models
- create a model suggesting how to correct wrong job reports

Thank you for the attention

Università degli Studi di Padova

Appendix

TF-IDF

$$\text{TF}(t, d) = \frac{\text{number of times term } t \text{ appears in document } d}{\text{number of terms in document } d}$$

$$\text{IDF}(t) = \log\left(\frac{\text{total number of documents}}{\text{number of documents containing stem } t}\right)$$

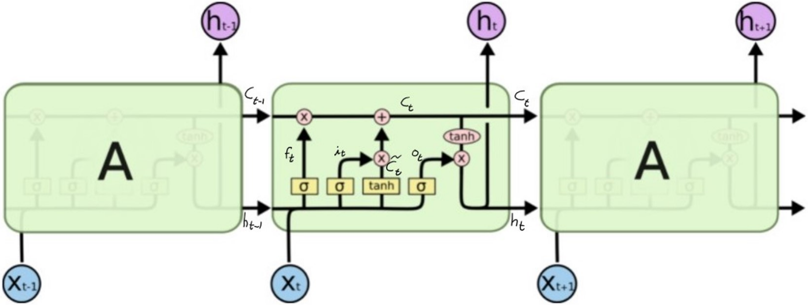


Figure: The repeating module in a LSTM (adapted from: Christopher, 2015).

Appendix

Exploration

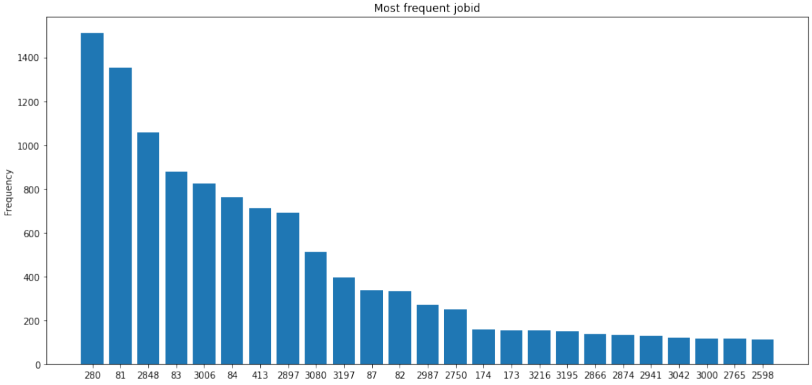


Figure: Frequency of the 25 most common project identifiers.

Appendix

Exploration

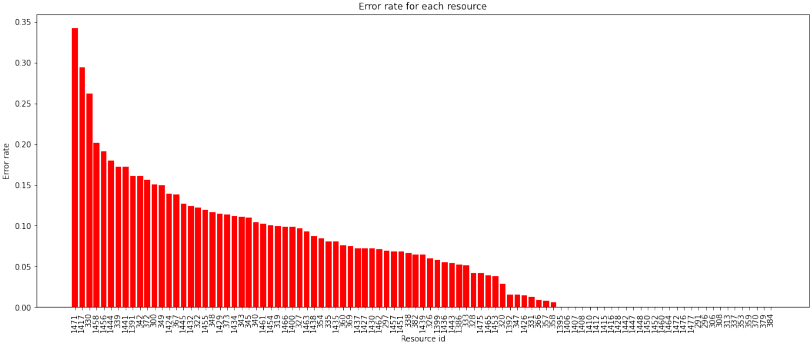


Figure: Error rate for each resource.

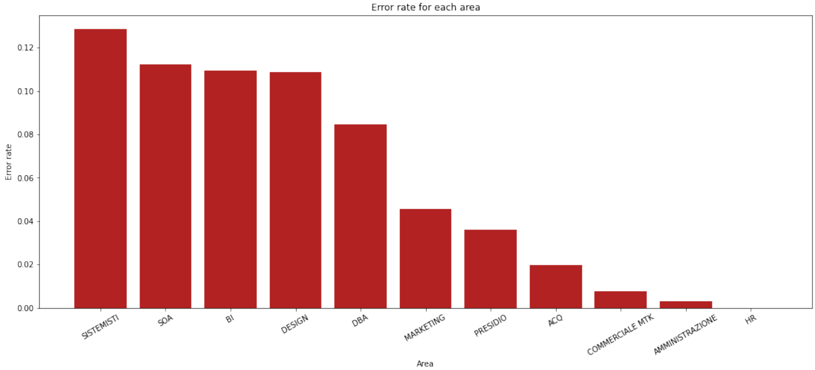


Figure: Error rate for each area.

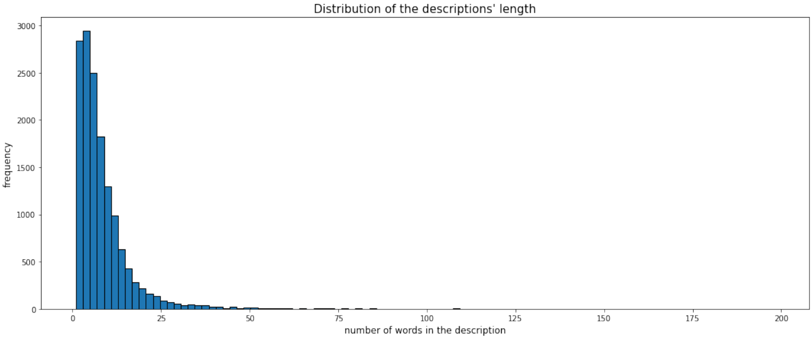


Figure: Distribution of the description length.

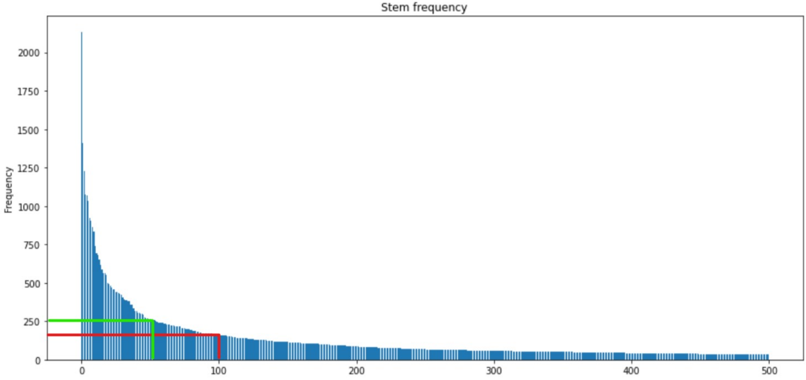


Figure: Distribution of the stem frequency.

Appendix

Exploration

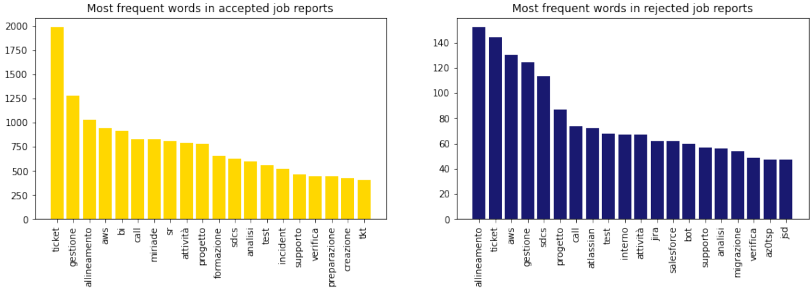


Figure: Most frequent stems of the two classes.

Appendix

Exploration

Figure: Word clouds of the two classes.

$$\begin{aligned}
\mathit{accuracy} &= \frac{TP + TN}{TP + FP + TN + FN} \\
\mathit{precision} &= \frac{TP}{TP + FP} \\
\mathit{sensitivity/recall} &= \frac{TP}{TP + FN} \\
\mathit{specificity} &= \frac{TN}{TN + FP} \\
\mathit{balanced\ accuracy} &= \frac{\mathit{sensitivity} + \mathit{specificity}}{2} \\
F_1\text{-}\mathit{score} &= 2 \times \frac{\mathit{precision} \times \mathit{sensitivity}}{\mathit{precision} + \mathit{sensitivity}} \\
\underset{\mathit{(MCC)}}{\mathit{Matthews\ Correlation\ Coefficient}} &= \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}
\end{aligned}$$

Appendix

Tuning parameters

model	tuning parameter
Logistic Regression	inverse of the regularization coefficient
	maximum number of iterations
SVM	regularization coefficient
	kernel type
	the kernel parameter γ
KNN	number of neighbors to use
	distance metric to use
Random Forest	number of trees in the forest
	criterion used for the information gain
	maximum depth of the tree
	minimum number of samples required to split an internal node
	minimum information gain to allow the split of a node
Gradient Boosting	number of boosting stages to perform
	the maximum depth of the individual estimators
	fraction of samples to be used

Table: List of model-specific tuning parameters.

Appendix

Tuning parameters

model	tuning-parameter	value
Logistic Regression	number of stems considered	3725
	n-gram range	1 — 3
	word vectorizer	countVectorized
	resampling approach	none
	inverse of the regularization coefficient	10
	maximum number of iterations	2048
SVM	number of stems considered	250
	n-gram range	1 — 1
	word vectorizer	TfidfVectorizer
	resampling approach	none
	regularization coefficient	0.01
	kernel type	linear
KNN	number of stems considered	2703
	n-gram range	1 — 3
	word vectorizer	countVectorizer
	resampling approach	none
	number of neighbors to use	5
	distance metric to use	minkowski

Table: List of tuning parameters and values used by the models with the highest validation accuracy.

model	tuning-parameter	value
Random Forest	number of stems considered	500
	n-gram range	1 — 2
	word vectorizer	TfidfVectorizer
	resampling approach	over
	number of trees in the forest	200
	criterion used for the information gain	Gini
	maximum depth of the tree	100
	min number of samples to split an internal node	2
	min information gain to allow the split of a node	0
Gradient Boosting	number of stems considered	2703
	n-gram range	1 — 2
	word vectorizer	countVectorizer
	resampling approach	none
	number of boosting stages to perform	50
	the maximum depth of the individual estimators	50
	fraction of samples to be used	1

Table: List of tuning parameters and values used by the models with the highest validation accuracy.

Appendix

Tuning parameters

model	tuning-parameter	value
Logistic Regression	number of stems considered	4935
	n-gram range	1 — 2
	word vectorizer	countVectorized
	resampling approach	mid strategy
	inverse of the regularization coefficient	1
	maximum number of iterations	8192
SVM	number of stems considered	2703
	n-gram range	1 — 2
	word vectorizer	TfidfVectorizer
	resampling approach	mid strategy
	regularization coefficient	0.1
	kernel type	linear
KNN	number of stems considered	4935
	n-gram range	1 — 3
	word vectorizer	TfidfVectorizer
	resampling approach	oversampling
	number of neighbors to use	3
	distance metric to use	minkowski

Table: List of tuning parameters and values used by the models with the highest balanced validation accuracy.

model	tuning-parameter	value
Random Forest	number of stems considered	3725
	n-gram range	$1 - 2$
	word vectorizer	TfidfVectorizer
	resampling approach	mid strategy
	number of trees in the forest	200
	criterion used for the information gain	Gini
	maximum depth of the tree	200
	min number of samples to split an internal node	2
	min information gain to allow the split of a node	0
Gradient Boosting	number of stems considered	4935
	n-gram range	$1 - 3$
	word vectorizer	countVectorizer
	resampling approach	mid strategy
	number of boosting stages to perform	100
	the maximum depth of the individual estimators	10
	fraction of samples to be used	1

Table: List of tuning parameters and values used by the models with the highest balanced validation accuracy.

Appendix

Tuning parameters

Università degli Studi di Padova

model	tuning-parameter	value
FFNN (highest accuracy)	number of stems considered	4935
	n-gram range	$1-1$
	word vectorizer	countVectorized
	resampling approach	none
	number of hidden layers	2
	hidden neurons	2048 — 2048
	activation	*GELU*
	optimizer	*RMSprop*
	learning rate	$1e-6$
	regularization	none
	number of training epoch	75

Table: List of the tuning parameters and values used by the best neural network architectures.

Appendix

Tuning parameters

model	tuning-parameter	value
FFNN (highest balanced accuracy)	number of stems considered	4935
	n-gram range	$1-1$
	word vectorizer	TfidfVectorized
	resampling approach	mid strategy
	number of hidden layers	2
	hidden neurons	$256-256$
	activation	*GELU*
	optimizer	*RMSprop*
	learning rate	$1e-4$
	regularization	none
	number of training epoch	15

Table: List of the tuning parameters and values used by the best neural network architectures.

Appendix

Tuning parameters

model	tuning-parameter	value
RNN (best architecture)	number of recurrent layers	3
	number of fully connected layers	3
	number of feature in the hidden state of the LSTM	128
	hidden neurons	1024 — 1024 — 1024
	activation	*GELU*
	optimizer	*Adam*
	learning rate	$1e-4$
	regularization	dropout
	number of training epoch	30

Table: List of the tuning parameters and values used by the best neural network architectures.

Appendix

Results and comparisons

Highest accuracy (only textual variable)

	LR	SVM	KNN	RF	GB	FFNN
accuracy	0.91	0.913	0.903	0.915	0.907	0.912
balanced accuracy	0.587	0.603	0.599	0.574	0.607	0.586

Highest balanced accuracy (only textual variable)

	LR	SVM	KNN	RF	GB	FFNN
accuracy	0.82	0.804	0.831	0.86	0.836	0.781
balanced accuracy	0.722	0.745	0.699	0.694	0.71	0.721

Highest accuracy (complete dataset)

	LR	SVM	KNN	RF	GB	FFNN	RNN	RNN + GB	PCA + GB
accuracy	0.923	0.908	0.911	0.948	0.946	0.912	0.925	0.946	0.935
balanced accuracy	0.721	0.502	0.565	0.783	0.789	0.586	0.659	0.794	0.72

Highest balanced accuracy (complete dataset)

	LR	SVM	KNN	RF	GB	FFNN	RNN	RNN + GB	PCA + GB
accuracy	0.873	0.843	0.855	0.933	0.906	0.775	0.903	0.831	0.908
balanced accuracy	0.813	0.84	0.722	0.828	0.861	0.716	0.862	0.849	0.518

UNIVERSITÀ DEGLI STUDI DI PADOVA

	accuracy	balanced accuracy	precision	sensitivity	specificity	MCC	F_1-score
LR	0.923	0.721	0.606	0.474	0.969	0.495	0.532
SVM	0.908	0.502	0.995	0.005	0.995	0.067	0.010
KNN	0.911	0.565	0.564	0.141	0.989	0.251	0.226
RF	0.948	0.783	0.807	0.581	0.986	0.658	0.675
GB	0.946	0.789	0.767	0.596	0.982	0.648	0.67
FFNN	0.912	0.586	0.581	0.186	0.986	0.295	0.282
RNN	0.925	0.659	0.694	0.333	0.985	0.447	0.45
RNN + GB	0.946	0.794	0.763	0.608	0.981	0.653	0.677
PCA + GB	0.935	0.72	0.736	0.457	0.983	0.548	0.677

Table: Classification metrics of the models with the highest validation accuracy.

Appendix

Results and comparisons

	accuracy	balanced accuracy	precision	sensitivity	specificity	MCC	F_1-score
LR	0.873	0.813	0.4	0.739	0.887	0.482	0.519
SVM	0.843	0.84	0.352	0.836	0.843	0.476	0.496
KNN	0.855	0.722	0.33	0.558	0.885	0.354	0.415
RF	0.933	0.828	0.624	0.7	0.957	0.624	0.66
GB	0.906	0.861	0.495	0.806	0.916	0.585	0.613
FFNN	0.775	0.716	0.237	0.643	0.789	0.288	0.346
RNN	0.925	0.659	0.694	0.333	0.985	0.447	0.45
RNN + GB	0.903	0.862	0.485	0.811	0.912	0.58	0.607
PCA + GB	0.831	0.849	0.339	0.871	0.827	0.476	0.489

Table: Classification metrics of the models with the highest balanced validation accuracy.

	Fitting time (s)	Predicting time (s)	Total time (s)
Logistic Regression	295.0	0.5	295.5
SVM	195.0	102.0	297.0
KNN	0.2	100.0	100.2
Random Forest	44.9	3.1	48.0
Gradient Boosting	388.4	1.3	389.7
FFNN	581.1	15.2	596.3
RNN	202.9	16.5	219.4
RNN + GB	452.8	1.6	454.4

Table: Running time comparison among the versions of the models with the highest validation accuracy. This is done in terms of fitting time, predicting time and total time. The unit of measurement is the second.

Appendix

Results and comparisons

	accuracy	balanced accuracy
Gradient Boosting (highest accuracy)	0.952	0.648
Gradient Boosting (highest balanced accuracy)	0.864	0.680

Table: Final results in the new test set.