Relazione dell'Analisi del Dataset
Traccia D7 Stelle ricette

Introduzione

Scopo dell'Analisi

Questa relazione fornisce un'analisi dettagliata del dataset **Recipe Reviews and User Feedback Dataset** con l'obiettivo di predire le stelle lasciate come feedback dagli utenti, partendo da poche semplici informazioni di contorno. L'analisi è stata condotta tramite il software KNIME, utilizzando i seguenti modelli:

- Decision tree
- Random forest
- Probabilistic neural network (PNN)
- Natural Language Processing (NLP)

Descrizione del Dataset

Il dataset è composto da 18182 righe rappresentanti i vari commenti alle 100 ricette descritte, per ogni riga ci sono i seguenti campi:

Variable Name	Type	Description
Variable 1	Integer	number of records
recipe_number	Integer	placement of the recipe on the top 100 recipes list
recipe_code	Integer	unique id of the recipe used by the site
recipe_name	Categorical	name of the recipe the comment was posted on
comment_id	Categorical	unique id of the comment
user_id	Categorical	unique id of the user who left he comment
user_name	Categorical	name of the user
user_reputation	Integer	internal score of the site, roughly roughly quantifying the past behaviour of the user
created_at	Integer	time at which the comment was posted as unix timestamp
reply_count	Integer	number of replies to the comment
thumbs_up	Integer	number of up-votes the comment has received
thumbs_down	Integer	number of down-votes the comment has received

stars	Integer	he score on a 1 to 5 scale that the user gave to the recipe. A score of 0 means that no score was given
best_score	Integer	score of the comment, likely used by the site the help determine the order the comments appear in
text	Categorical	the text content of the comment

I commenti sono così suddivisi:
0 stelle → 1696 → 9.3%
1 stelle → 280 → 1.5%
2 stelle → 232 → 1.3%
3 stelle → 490 → 2.7%
4 stelle → 1655 → 9.1%
5 stelle → 13829 → 76.1%
Notiamo quindi che il dataset è fortemente sbilanciato, questo potrebbe portare all'overfitting.

Metodologia
Tramite le informazioni fornite dai creatori del dataset possiamo capire le operazioni su esso svolte e la metodologia di classificazione
Individuiamo una serie di parametri ritenuti "superflui" poiché non comporterebbero modifiche alle analisi del dataset :

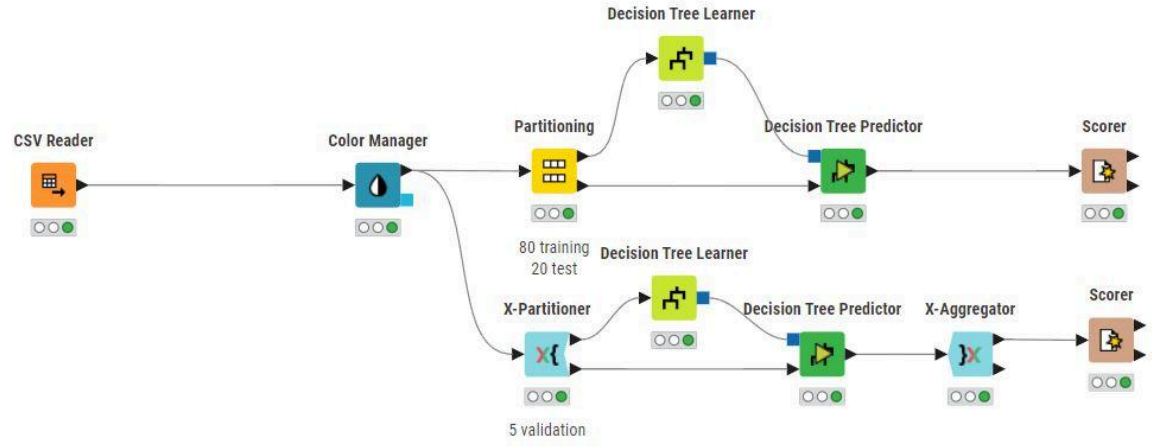
- Variable 1
- recipe_numbe
- recipe_code
- recipe_name
- comment_id
- user_id
- user_name
- created_at

Inizialmente verranno mantenuti con la possibilità di essere esclusi in futuro per un alleggerimento del dataset.

Modellazione

Come detto in precedenza, sono state utilizzate diverse tecniche:

1) *DECISION TREE*



Come primo approccio abbiamo utilizzato un decision tree, con un partitioning 80-20, ottenendo i seguenti risultati:

stars \ Prediction (stars)	0	1	2	3	4	5
0	98	3	1	3	20	214
1	9	9	1	4	5	28
2	7	6	2	3	1	28
3	12	3	3	5	8	67
4	19	3	3	5	41	260
5	201	11	17	31	190	2316

Correct classified: 2.471 Wrong classified: 1.166

Accuracy: 67,941% Error: 32,059%

Cohen's kappa (κ): 0,145%

Confusion Matrix senza validation

Accuracy statistics (Table)

Rows: 7 | Columns: 11

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
1	0	98	248	3050	241	0.289	0.283	0.289	0.925	0.286		
2	1	9	26	3555	47	0.161	0.257	0.161	0.993	0.198		
3	2	2	25	3565	45	0.043	0.074	0.043	0.993	0.054		
4	3	5	46	3493	93	0.051	0.098	0.051	0.987	0.067		
5	4	41	224	3082	290	0.124	0.155	0.124	0.932	0.138		
6	5	2316	597	274	450	0.837	0.795	0.837	0.315	0.816		
7	Over...										0.679	0.145

Tabella scorer senza validation

Accuracy statistics (Table)

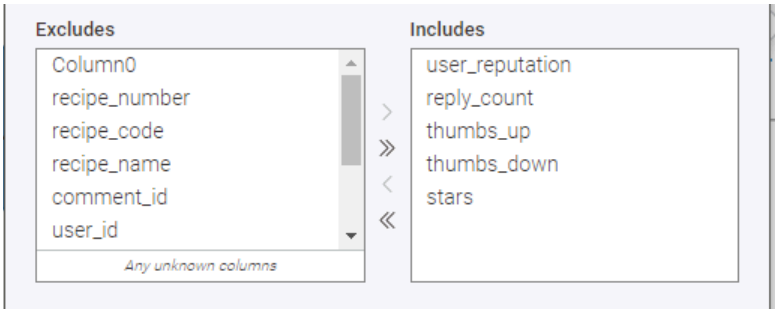
Rows: 7 | Columns: 11

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
1	0	460	1185	15301	1236	0.271	0.28	0.271	0.928	0.275	?	?
2	1	37	112	17790	243	0.132	0.248	0.132	0.994	0.172	?	?
3	2	7	79	17871	225	0.03	0.081	0.03	0.996	0.044	?	?
4	3	20	243	17449	470	0.041	0.076	0.041	0.986	0.053	?	?
5	4	215	1193	15334	1440	0.13	0.153	0.13	0.928	0.14	?	?
6	5	11681	2950	1403	2148	0.845	0.798	0.845	0.322	0.821	?	?
7	Over...	?	?	?	?	?	?	?	?	?	0.683	0.148

Tabella scorer con validation

Possiamo notare come i valori non abbiamo subito grosse modifiche.

Procediamo aggiungendo un nodo “column filter”, così settato:



Ottenendo questi risultati:

Accuracy statistics (Table)

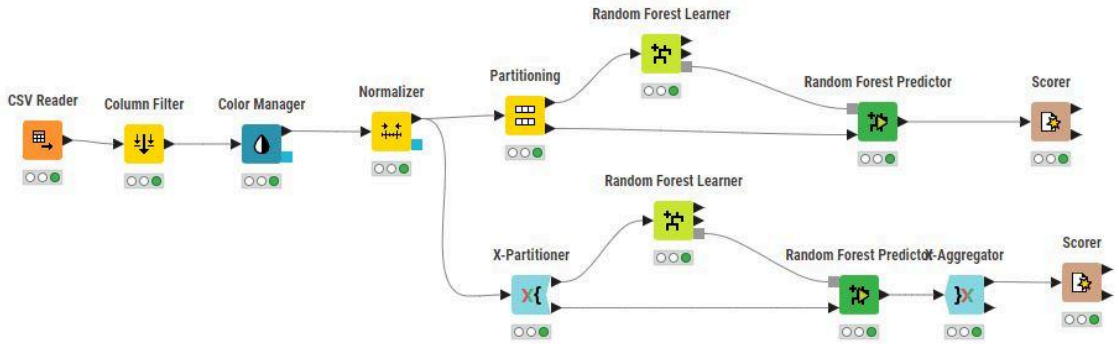
Rows: 7 | Columns: 11

#	R...	TruePositiv... Number (integer)	FalsePositi... Number (integer)	TrueNegati... Number (integer)	FalseNegati... Number (integer)	Recall Number (double)	Precision Number (double)	Sensitivity Number (double)	Specificity Number (double)	F-measure Number (double)	Accuracy Number (double)	Cohen's kap... Number (double)
1	0	15	85	3213	324	0.044	0.15	0.044	0.974	0.068	?	?
2	1	7	11	3570	49	0.125	0.389	0.125	0.997	0.189	?	?
3	2	0	7	3583	47	0	0	0	0.998	?	?	?
4	3	1	18	3521	97	0.01	0.053	0.01	0.995	0.017	?	?
5	4	5	85	3221	326	0.015	0.056	0.015	0.974	0.024	?	?
6	5	2623	780	91	143	0.948	0.771	0.948	0.104	0.85	?	?
7	Over...	?	?	?	?	?	?	?	?	?	0.729	0.043

Tabella scorer senza validation

Possiamo notare un aumento per quanto riguarda l’accuracy, ma un peggioramento nel Cohen’s kappa, per quanto riguarda invece gli specifici casi, notiamo una diminuzione di tutte le recall tranne nei commenti a 5 stelle dove raggiunge lo 0.948 vs 0.837 precedentemente ottenuto.

2) *RANDOM FOREST*



iniziamo con 100 alberi, ottenendo i seguenti risultati:

Accuracy statistics (Table)

Rows: 7 | Columns: 11

#	RowID	TruePositi... Number (integ...	FalsePosit... Number (integ...	TrueNega... Number (integ...	FalseNega... Number (integ...	Recall Number (doub...	Precision Number (doub...	Sensitivity Number (doub...	Specificity Number (doub...	F-measure Number (doub...	Accuracy Number (doub...	Cohen's k... Number (doub...
1	0	5	12	3286	334	0.015	0.294	0.015	0.996	0.028	?	?
2	1	10	20	3561	46	0.179	0.333	0.179	0.994	0.233	?	?
3	2	0	3	3587	47	0	0	0	0.999	?	?	?
4	3	1	5	3534	97	0.01	0.167	0.01	0.999	0.019	?	?
5	4	0	1	3305	331	0	0	0	1	?	?	?
6	5	2752	828	43	14	0.995	0.769	0.995	0.049	0.867	?	?
7	Over...	?	?	?	?	?	?	?	?	?	0.761	0.047

Tabella scorer senza validation

Accuracy statistics (Table)

Rows: 7 | Columns: 11

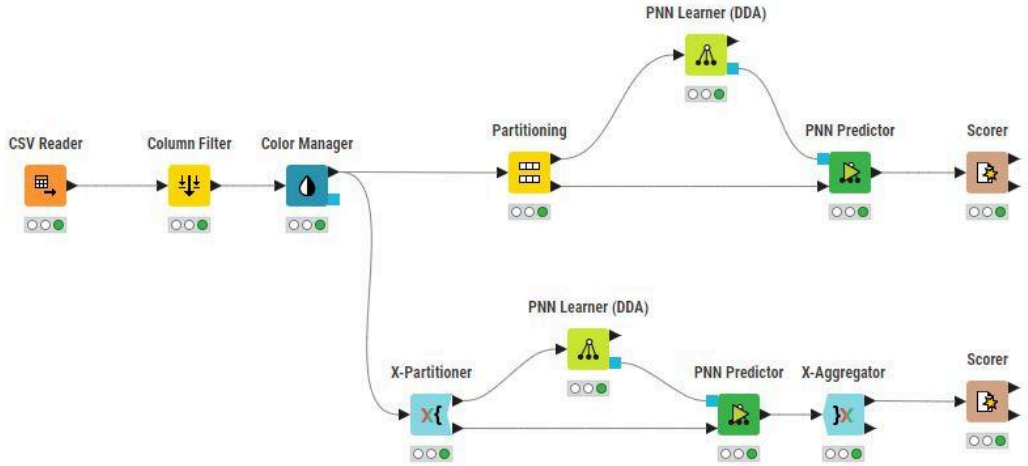
#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
1	0	16	73	16413	1680	0.009	0.18	0.009	0.996	0.018	?	?
2	1	44	77	17825	236	0.157	0.364	0.157	0.996	0.219	?	?
3	2	0	11	17939	232	0	0	0	0.999	?	?	?
4	3	2	28	17664	488	0.004	0.067	0.004	0.998	0.008	?	?
5	4	3	15	16512	1652	0.002	0.167	0.002	0.999	0.004	?	?
6	5	13774	4139	214	55	0.996	0.769	0.996	0.049	0.868	?	?
7	Over...	?	?	?	?	?	?	?	?	?	0.761	0.044

Tabella scorer con validation

Procediamo aumentando il numero degli alberi, ottenendo però risultati pressoché identici, questo ci porta alle seguenti conclusioni:

A. Il nostro modello è saturo: aggiungere ulteriori alberi non comporta nessun miglioramento
B. I dati non sono sufficientemente descrittivi, e non portano miglioramenti.

3) *PNN*



Accuracy statistics (Table)

Rows: 7 | Columns: 11

#	RowID	TruePositi... Number (integ...	FalsePosit... Number (integ...	TrueNegat... Number (integ...	FalseNega... Number (integ...	Recall Number (doub...	Precision Number (doub...	Sensitivity Number (doub...	Specificity Number (doub...	F-measure Number (doub...	Accuracy Number (doub...	Cohen's k... Number (doub...
1	0	18	33	3265	321	0.053	0.353	0.053	0.99	0.092		
2	1	2	2	3579	54	0.036	0.5	0.036	0.999	0.067		
3	2	0	3	3587	47	0	0	0	0.999			
4	3	2	2	3537	96	0.02	0.5	0.02	0.999	0.039		
5	4	3	15	3291	328	0.009	0.167	0.009	0.995	0.017		
6	5	2718	839	32	48	0.983	0.764	0.983	0.037	0.86		
7	Over...										0.754	0.034

Tabella scorer senza validation

Accuracy statistics (Table)

Rows: 7 | Columns: 11

#	RowID	TruePositi... Number (integ...	FalsePosit... Number (integ...	TrueNegat... Number (integ...	FalseNega... Number (integ...	Recall Number (doub...	Precision Number (doub...	Sensitivity Number (doub...	Specificity Number (doub...	F-measure Number (doub...	Accuracy Number (doub...	Cohen's ka... Number (doub...
1	0	233	344	16142	1463	0.137	0.404	0.137	0.979	0.205		
2	1	9	40	17862	271	0.032	0.184	0.032	0.998	0.055		
3	2	0	49	17901	232	0	0	0	0.997			
4	3	5	83	17609	485	0.01	0.057	0.01	0.995	0.017		
5	4	53	252	16275	1602	0.032	0.174	0.032	0.985	0.054		
6	5	13224	3890	463	605	0.956	0.773	0.956	0.106	0.855		
7	Over...										0.744	0.083

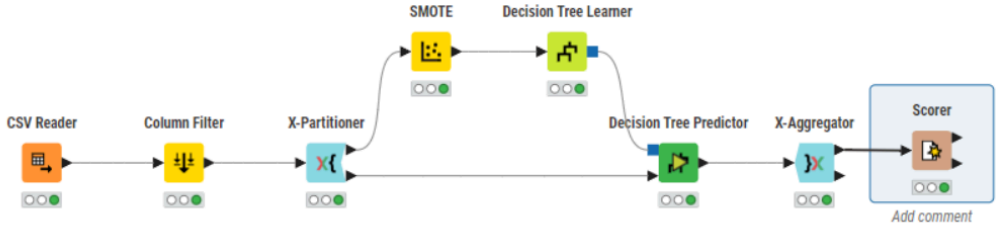
Tabella scorer con validation

Notiamo che, anche con questo approccio, otteniamo dei risultati simili ai precedenti: una accuracy abbastanza buona (sul 75%), un Cohen's kappa molto basso, delle precision molto basse (tranne nel caso a 5 stelle) e delle recall bassissime (tranne nel caso a 5 stelle).

I nostri modelli si comportano bene nei casi a 5 stelle (abbiamo avuto precision di poco inferiori all'80% con quasi il 99% di recall) ma molto male negli altri casi, ci chiediamo quindi se questo possa essere dovuto allo sbilanciamento presente nel dataset, notiamo che l'accuracy dei nostri modelli, in media del 73%, è molto vicina alla presenza di commenti a 5 stelle (76.1%), i modelli non riescono quindi ad adattarsi bene soprattutto nei casi 1, 2 e 3 stelle (rispettivamente 1.5%, 1.3% e 2.7% dei commenti totali). Siamo quindi andati in contro ad un caso di overfitting, dove il modello non riesce a generalizzare bene per le classi poco rappresentate a discapito di quella maggioritaria.

Data l'impossibilità di raccogliere ulteriori dati per le classi poco rappresentate, proviamo quindi a fare il resampling del dataset per ottenere dei modelli migliori:

A. aggiungiamo copie delle classi sotto-rappresentate
   Utilizziamo 2 strategie per farlo:
   A1. *SMOTE*
   Genera sinteticamente nuovi campioni per le classi sotto-rappresentate, si basa sul concetto di neighbor.



0 stelle da 1529 → a 12431
1 stelle da 255 → a 12431
2 stelle da 214 → a 12431
3 stelle da 445 → a 12431
4 stelle da 1489 → a 12431
5 stelle da 12431 → a 12431

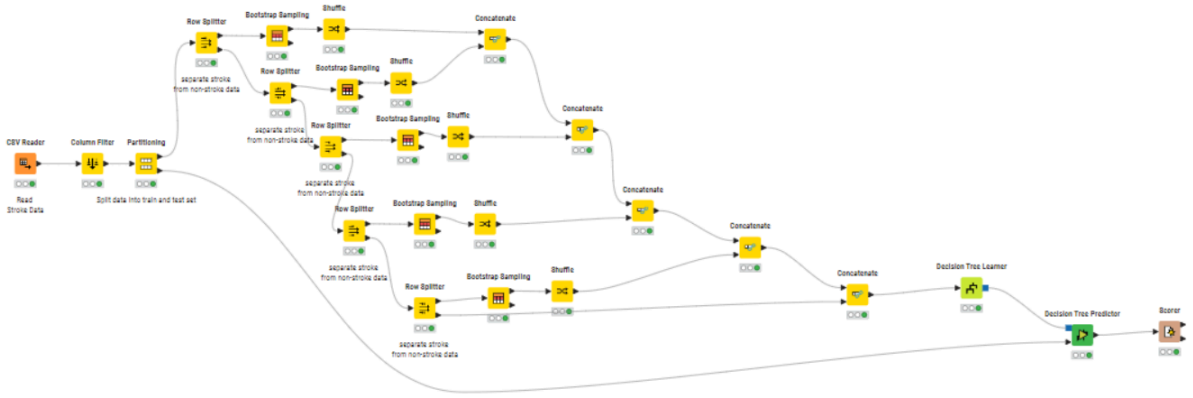
s: 7 | Columns: 11

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k Number (do
1	0	80	383	16103	1616	0.047	0.173	0.047	0.977	0.074		
2	1	60	478	17424	220	0.214	0.112	0.214	0.973	0.147		
3	2	14	298	17652	218	0.06	0.045	0.06	0.983	0.051		
4	3	33	1284	16408	457	0.067	0.025	0.067	0.927	0.037		
5	4	93	727	15800	1562	0.056	0.113	0.056	0.956	0.075		
6	5	11522	3210	1143	2307	0.833	0.782	0.833	0.263	0.807		
7	Over...										0.649	0.063

Possiamo notare un peggioramento generale del modello: il nodo smote non ha portato ad alcun miglioramento, proviamo quindi con altre tecniche.

A2. *BOOTSTRAP SAMPLING*

Genera un certo numero di campioni, ognuno dei quali può differire leggermente rispetti a quelli di input a causa del campionamento casuale con sostituzione.



0 stelle da 1357 → a 10000
1 stelle da 224 → a 10000
2 stelle da 185 → a 10000
3 stelle da 392 → a 10000
4 stelle da 1325 → a 10000
5 stelle da 11063 → a 11063

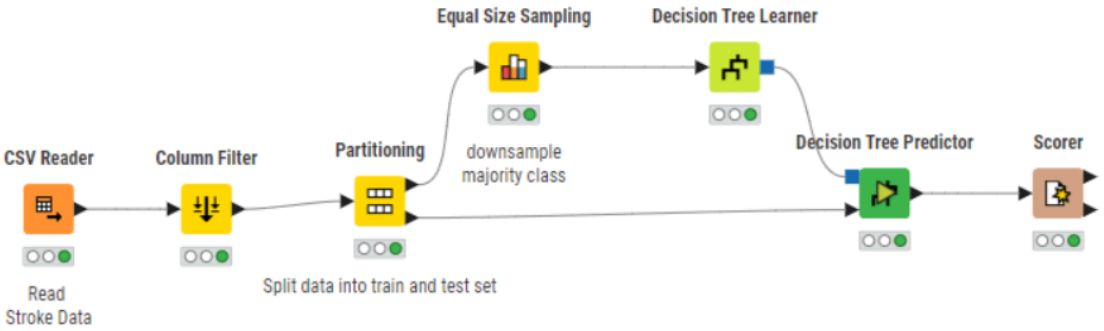
s: 7 | Columns: 11

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's Number (d
1	0	13	90	3208	326	0.038	0.126	0.038	0.973	0.059	?	?
2	1	12	57	3524	44	0.214	0.174	0.214	0.984	0.192	?	?
3	2	5	212	3378	42	0.106	0.023	0.106	0.941	0.038	?	?
4	3	12	295	3244	86	0.122	0.039	0.122	0.917	0.059	?	?
5	4	7	62	3244	324	0.021	0.101	0.021	0.981	0.035	?	?
6	5	2246	626	245	520	0.812	0.782	0.812	0.281	0.797	?	?
7	Over...	?	?	?	?	?	?	?	?	?	0.631	0.058

Otteniamo risultati molto simili a quelli precedenti: anche questa tecnica non si è rivelata troppo proficua

B. UNDERSAMPLING

Eliminiamo delle istanze della classe sovra-rappresentata



0 stelle da 1357 → a 185
1 stelle da 224 → a 185
2 stelle da 185 → a 185
3 stelle da 392 → a 185
4 stelle da 1325 → a 185
5 stelle da 11063 → a 185

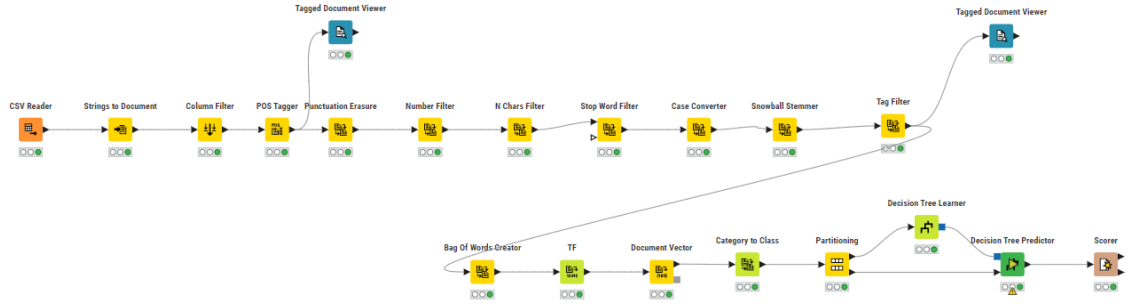
#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's Number (
1	0	35	128	105	11	0.761	0.215	0.761	0.451	0.335	?	?
2	1	18	35	198	28	0.391	0.34	0.391	0.85	0.364	?	?
3	2	4	14	216	45	0.082	0.222	0.082	0.939	0.119	?	?
4	3	3	9	224	43	0.065	0.25	0.065	0.961	0.103	?	?
5	4	1	5	228	45	0.022	0.167	0.022	0.979	0.038	?	?
6	5	7	20	213	39	0.152	0.259	0.152	0.914	0.192	?	?
7	Over...	?	?	?	?	?	?	?	?	?	0.244	0.094

Ottenendo così un modello che si comporta leggermente meglio nel caso 0-4 stelle ma molto peggio nel caso 5 stelle, diminuendo così l'accuracy a 0.244. Questo è dovuto, probabilmente, al fatto che sono stati scartati il 92% dei valori di training,

Anche questa tecnica si è rivelata non determinante.

Cercando possibili soluzioni on-line, ci siamo imbattuti in questo articolo, dove si parla di un problema analogo al nostro: una forte discrepanza tra le rappresentazioni delle classi, anche l'autore prova ad utilizzare le nostre stesse tecniche, purtroppo arriva alla nostra stessa conclusione: la differenza tra le classi è troppo ampia ed, in determinati casi, il resampling non funziona.

4)*Natural Language Processing NLP*



Analizziamo ora una variabile fino ad ora ignorata : **text**
per far questo è necessario attuare, prima della fase di learning, una fase di preparazione del testo dove ogni commento viene preso, suddiviso in parole, ognuna viene catalogata (nome, aggettivo, avverbio, verbo…), si toglie la punteggiatura, i numeri ed alcune parole sono rimosse in modo da lasciare solo le più significative.
Ottenendo, per ogni commento una struttura simile a questa:

Dopodiché costruiamo un bitvector per rappresentare la relazione tra i commenti e le parole in essi contenute.

Rows: 18182 | Columns: 21029

	#	RowID	Document Text document	amish[JJ... Number (dou...	breakfast... Number (dou...	casserol[... Number (dou...	39d[VBD(... Number (dou...	love[VB(... Number (dou...	cut[VBN(... Number (dou...	half[DT(P... Number (dou...
☐	1	Row0	"amish breakf...	1	1	1	1	1	1	1
☐	2	Row1	"amish breakf...	1	1	1	0	0	0	0
☐	3	Row2	"amish breakf...	1	1	1	0	0	0	0
☐	4	Row3	"amish breakf...	1	1	1	0	0	0	0
☐	5	Row4	"amish breakf...	1	1	1	0	0	0	0
☐	6	Row5	"amish breakf...	1	1	1	0	0	0	0
☐	7	Row6	"amish breakf...	1	1	1	0	0	0	0
☐	8	Row7	"amish breakf...	1	1	1	0	0	0	0
☐	9	Row8	"amish breakf...	1	1	1	0	0	0	0
☐	10	Row9	"amish breakf...	1	1	1	0	0	0	0
☐	11	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	12	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	13	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	14	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	15	Row...	"amish breakf...	1	1	1	0	1	0	0
☐	16	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	17	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	18	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	19	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	20	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	21	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	22	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	23	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	24	Row...	"amish breakf...	1	1	1	0	0	0	0
☐	25	Row...	"amish breakf...	1	1	1	0	0	0	0

Costruiamo ora un albero decisionale:

| Columns: 11

RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
0	57	236	3072	272	0.173	0.195	0.173	0.929	0.183	?	?
1	7	41	3532	57	0.109	0.146	0.109	0.989	0.125	?	?
2	0	19	3575	43	0	0	0	0.995	?	?	?
3	7	42	3500	88	0.074	0.143	0.074	0.988	0.097	?	?
4	37	192	3126	282	0.116	0.162	0.116	0.942	0.135	?	?
5	2391	608	242	396	0.858	0.797	0.858	0.285	0.826	?	?
Over...	?	?	?	?	?	?	?	?	?	0.687	0.118

Purtroppo anche questo modello non ci permette di raggiungere i dati sperati, questo potrebbe esser dovuto a diversi fattori:

- Si cerca la correlazione tra il testo e le stelle, tralasciando gli altri valori (like/dislike, reputazione,...)
- Non è stato possibile utilizzare modelli più complessi (Random forest…) oppure utilizzare ad esempio la 5 fold cross validation, a causa dell'enorme sforzo computazionale che è richiesto per questo tipo di analisi.

- Come detto già in precedenza, il dataset risulta essere troppo sbilanciato.

Conclusioni
Come precedentemente evidenziato, tutti gli approcci che abbiamo utilizzato ci portano ad avere modelli che rispondono in maniera abbastanza simile, comportandosi bene nei casi di commente a 5 stelle, ma molto male negli altri. Nonostante i vari tentativi di resampling effettuati (con diverse tecniche), arriviamo alla conclusione che il dataset è troppo sbilanciato per poter creare un modello che risponda bene in tutti i casi, la soluzione migliore sarebbe quindi ampliare il dataset per rappresentare meglio le classi minoritarie e solo successivamente riprovare un'analisi.

In vista di una futura analisi, potrebbe essere utile avere maggiori informazioni sui campi del dataset: best-score e user-reputation.
Conoscendo più approfonditamente il loro funzionamento potrebbero essere integrati nel calcolo per la predizione delle stelle

Una possibile conclusione che possiamo trarre da questa analisi (ipotizzando che la raccolta dei dati sia avvenuta senza errori metodologici o di forma) è l'esistenza di un bias negli utenti del sito preso in considerazione, che li porta a lasciare maggiormente recensioni quando esse risultano fortemente positive tralasciando quelle neutre o negative.