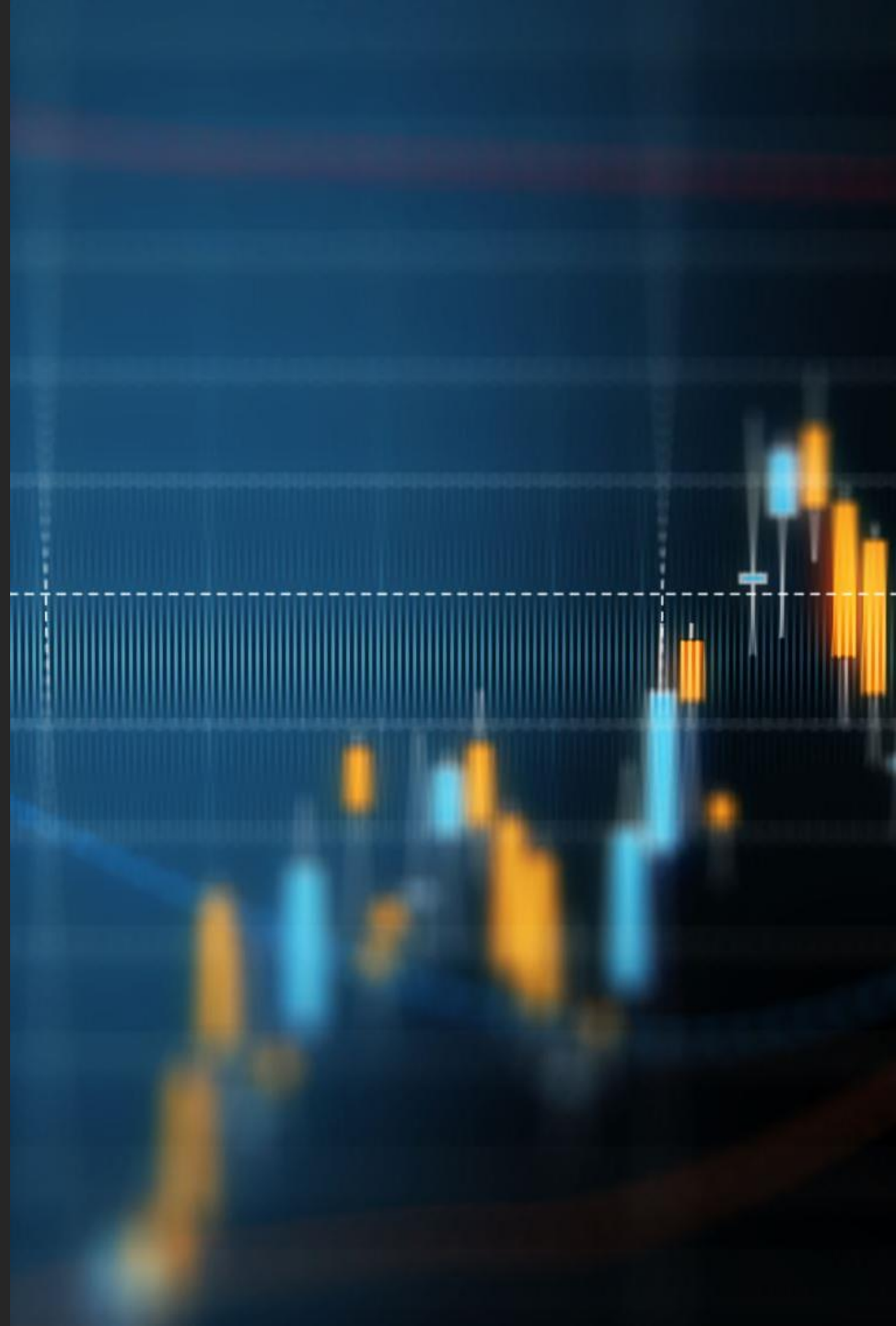
Akceptacja kredytowa

PROBLEM KLASYFIKACYJNY

Opis problemu

W XXI wieku ilość aplikacji o kredyt uniemożliwia rozważanie każdego pojedyńczego wniosku od początku przez pracownika banku, wymagana jest wstępna filtracja wniosków.

Posiadając sporą porcję danych składanych na wstępnym etapie wnioskowania, wyposażonych w werdykt banku, za cel zostało nam postawione napisanie algorytmu uczenia maszynowego przewidującego werdykt banku w pełni automatycznie. Każdy wniosek złożony bankowi jest ostatecznie rozważany przez bardziej złożone algorytmy biorące pod uwagę również inne czynniki, nasz model ma zostać jednak udostępniony na stronie bankowej dla każdej osoby mając na celu ograniczenie irracjonalnych wniosków oraz pomoc klientom w podjęciu decyzji. Interakcja z naszym modelem nie jest wiążąca dla żadnej ze stron.



Opis danych

Data Set Characteristics:	Multivariate	**Number of Instances:**	690
Attribute Characteristics:	Categorical, Integer, Real	**Number of Attributes:**	15
Associated Tasks:	Classification	**Missing Values?**	Yes

Nasz model otrzymywać będzie przesłane na stronie zaszyfrowane dane klientów. Ich znaczenie nie zostało nam jednak udostępnione.

Pracujemy z danymi mogącymi zawierać brakujące wartości. Są reprezentowane przez atrybuty rzeczywiste, tekstowe jak I binarne.

Wszelkie możliwe wartości przyjmowane przez dane nienumeryczne zostały zawarte w udostępnionej nam bazie danych.

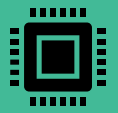
Każdy dalej opisany krok jest "uczony" tylko na części danych, niezawierające się w nim rekordy używane są jako wskaźniki jakości wykonanej przez nas pracy.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	
a	58.67	4.46	u	g	q	h	3.04	t	t	6	f	g	43	560	+
a	24.50	0.5	u	g	q	h	1.5	t	f	0	f	g	280	824	+
b	27.83	1.54	u	g	w	v	3.75	t	t	5	t	g	100	3	+
b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	120	0	+
b	32.08	4	u	g	m	v	2.5	t	f	0	t	g	360	0	+
b	33.17	1.04	u	g	r	h	6.5	t	f	0	t	g	164	31285	+
a	22.92	11.585	u	g	cc	v	0.04	t	f	0	f	g	80	1349	+
b	54.42	0.5	y	p	k	h	3.96	t	f	0	f	g	180	314	+
b	42.50	4.915	y	p	w	v	3.165	t	f	0	t	g	52	1442	+

Przetwarzanie danych

Aby dane mogły być użyte w modelach klasyfikacyjnych muszą zostać zakodowane pod postacią liczb.

Wszelkie brakujące dane zostają uzupełnione przez najczęściej występującą wartość w ich kategorii dla danych tekstowych I średnią dla danych numerycznych.

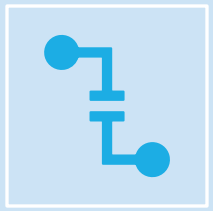


Wszelkie dane tekstowe zostają zakodowane za pomocą metowy One Hot Encoding. Tam, gdzie to możliwe (tj dane są I tekstowe I binarne) zatrzymujemy tylko jednen wektor odkodowanych wartości, aby zmniejszyć liczbę końcowych atrybutów.

Kolumny w pełni numeryczne zostają znormalizowane.

Modelowanie danych

One Hot Endocing pozostawia nas z relatywnie dużą ilością atrybutów (42) w porównaniu do ilości rekordów na których się "uczymy" (552). Przy na tyle relatywnie małym zbiorze danych może (aczkolwiek nie musi) być to problemem, dla tego każdy szkolony przez nas model może rozpatrywać atrybuty jedynie najbardziej powiązane z naszym celem (przewidzeniem werdyktu banku). Jest to traktowane jako parametr każdego modelu, paramter którego najwydajniejszą wartość (rozumianą jako ile I jakie atrybuty odrzucić) będziemy szukać w procesie uczenia.



Analogiczne podejście zostało zastosowane dla rekordów najbardziej odbiegających od standardowego wejścia w naszym zestawie danych. Wszystkie modele będą szkolone na zbiorach danych zmniejszonych o maksymalnie 15%.

Rozważane modele - wstęp

Wstępne rozważania paru modeli mają na celu pozwolenie na wybór 3 najbardziej obiecujących, aby można było im poświęcić więcej czasu I mocy obliczeniowej by udoskonalić ich wyniki.

Uwzględniając wagę modeli, ich wyniki na wstępnej ewaluacji I czas ich szkolenia na wejściu odrzucamy (1), (2), (5).

	Nazwa (angl)	Dokładność <0, 1>
1	Logistic Regression	0.8333
2	Multi-Layer Perceptron Classifier	0.8406
3	K Nearest Neighbors Classifer	0.7971
4	Support Vector Classifier	0.8406
5	Ada Boost Classifier	0.7971
6	Random Forest Classifier	0.8623

	Nazwa (angl)	Dokładność <0, 1>
1	K Nearest Neighbors Classifer	0.8551
2	Random Forest Classifier 1	0.8406
3	Random Forest Classifier 2	0.8551
4	Random Forest Classifier 3	0.8623
5	Support Vector Classifier	0.8406

Rozważane modele - dane szczegółowe

Przy późniejszym wyborze modelu końcowego uwzględniono:

- (1) jest najszybszy I najlżejszy

- (2), (3), (4) mimo swojej wagi są zależne od losowych wartości generowanych podczas szkolenia, jest szansa na znalezienie lepszych

Rozważane modele - osiągi

	Nazwa (angl)	Accuracy <0, 1>	Precision	Recall	F1
1	K Nearest Neighbors Classifer	0.8551	0.8529	0.8529	0.8529
2	Random Forest Classifier 1	0.8406	0.8382	0.8382	0.8382
3	Random Forest Classifier 2	0.8551	0.8243	0.8971	0.8592
4	Random Forest Classifier 3	0.8623	0.8356	0.8971	0.8652
5	Support Vector Classifier	0.8406	0.8382	0.8382	0.8382
6	Voting Classifier (all together)	0.8478	0.8406	0.8529	0.8467

Nazwa (angl)	Accuracy <0, 1>	Precision	Recall	F1
Random Forest Classifier3	0.9507	0.944	0.9687	0.9562

Podsumowanie

Po zmienieniu random state Random Forest Classifier osiągnął dokładność ~87.7% na danych treningowych, 95% na wszystkich danych. Został wybrany jako finalny ze względu na najlepszy f1 score spośród rozważanych kandydatów.