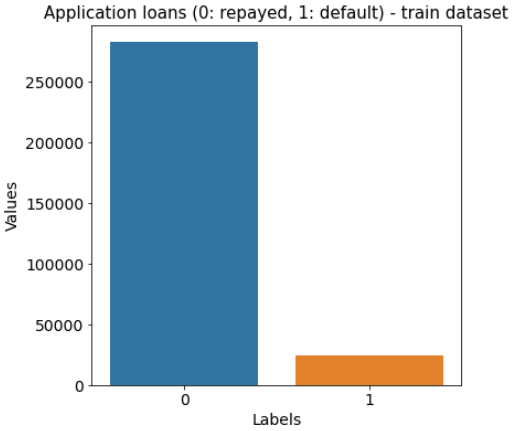
Méthodologie du modèle IA de conseils à l'attribution de crédits

« Home Credit AI advice »

SOMMMAIRE

1. Conception et entraînement du modèle

1.1. Problématique



La problématique posée est de définir si le client va rembourser son prêt ou faire défaut à partir des données de précédents clients. C'est un problème supervisé de classification binaire. Les données sont fortement disproportionnées avec plus de 280000 clients ayant remboursé leur prêt et un peu moins de 25000 ayant fait défaut.

1.2. Traitement des données

La base de données de Home Credit comporte initialement plus de 300 000 clients et leur données personnelles récoltées lors de leur demande de prêt, leur historique de remboursement, l'historique de leur précédents crédits à Home Credit et dans d'autres établissement bancaires. Ces données sont réparties sur 7 datasets, plus 1 fichier de description des variables. Les datasets sont liés par les identifiants clients ou les identifiants des demandes de prêts antérieures. L'ensemble des datasets comporte 206 variables. Dans un premier temps, l'analyse exploratoire a permis de déterminer les variables ayant un impact significatif sur la proportion de clients remboursant ou non leur prêt. Les variables catégorielles ayant un impact ont été transformées :

- soit en variables ordinales avec une valeur plus forte lorsque le risque de défaut est plus grand,
- soit en score allant de 0 à 10 proportionnel au pourcentage de clients ayant fait défaut, un score élevé indiquant un risque de défaut plus grand,
- soit laissée sous forme de variable catégorielle pour les variables du fichier "application" qui seront ensuite transformées avec un One Hot Encoder.

La transformation en variables ordinales/numériques à ce stade est surtout nécessaire pour l'agrégation des datasets (i.e. un client peut avoir plusieurs prêts antérieurs à agréger en une seule ligne). Les datasets sont donc agrégés pour obtenir un seul fichier avec une ligne par client, tout en ajoutant des variables par feature engineering. Les colonnes créées sont de type minimum, maximum, moyenne, somme ou écart-type des variables de base.

Les valeurs manquantes sont gérées par suppression des variables avec un taux de remplissage inférieur à 80%, excepté la variable EXT_SOURCE_3. Cette variable conservée malgré un taux de remplissage proche de 50% car elle ressort comme une variable importante quelque soit le type de modèle utilisé. Les valeurs manquantes de la variable OCCUPATION_TYPE (profession) sont remplacées par 'Unknown'. Puis les lignes ayant encore des valeurs manquantes sont supprimées. Cela suppose que les variables conservées sont toutes nécessaires pour la modélisation (pas d'imputation ou de remplacement des NaN).

Une étude approfondie de la corrélation entre les variables (matrices de Spearman) mène à supprimer de nombreuses variables trop corrélées entre elles (supprimées si coefficient de corrélation supérieur ou égal à 0,6).

A ce stade, le dataset comporte 129 variables + l'identifiant client + la target.

1.3. Preprocessing

Les variables catégorielles restantes sont transformées avec un One Hot Encoder, les variables numériques sont centrées/réduites avec un Standard Scaler. Le dataset comporte alors 229 variables.

Les données sont séparées en train (80%) et test (20%).

Le train set est alors rééquilibré avec SMOTE par oversampling. On préfère cette méthode à l'undersampling ou une combinaison undersampling/oversampling car notre traitement des valeurs manquantes a considérablement réduit le dataset de base. On obtient un train set d'environ 119000 lignes et un test set d'environ 16000 lignes après preprocessing (Fig. 1).

Figure 1. Répartition des target avant et après oversampling.

1.4. Modèles testés, optimisation et évaluation

Plusieurs types de modèles ont été testés : des modèles linéaires (régressions logistiques, SVC linéaire) et ensemblistes (Random Forest, LightGBM). Les réseaux de neurones ont été écartés en raison de leur manque d'explicabilité : les résultats de l'algorithme d'aide à la décision doivent pouvoir être justifiés de façon compréhensible pour les assureurs et les clients.

Plusieurs metrics sont calculés pour l'évaluation : accuracy, balanced accuracy, précision, rappel, F-mesure, spécificité, ROC AUC. La mesure privilégiée dans un premier temps pour classer les modèles est l'aire sous la courbe ROC (ROC AUC).

Un premier run avec les paramètres de base donne un premier aperçu des performances des modèles. Ceux-ci sont entraînés sur le train set, testés sur le test set. Les modèles linéaires donnent des résultats significativement meilleurs que les modèles ensemblistes (AUC sur le test set d'environ 0.68 contre environ 0.50).

On cherche ensuite à optimiser les hyperparamètres et à tester la fiabilité des résultats avec des cross-validations. Pour la régression logistique, le solver saga ne converge jamais, et sag a des difficultés à converger. Ceci réduit les possibilités aux régularisations L1 ou L2 (elastic net impossible) et aux autres solvers (Newton-cg, LBFGS, Liblinear). Les meilleurs résultats sont obtenus avec une régularisation L2, le solver newton-cg et une valeur de C = 100 (AUC sur test set = 0,687). A la cross-validation, les scores sur les sets de train et validation se recoupent et sont aux alentours de 0,8.

Les autres modèles ont des résultats inférieurs, y compris les modèles ensemblistes après réglage fin des hyperparamètres (Fig. 2).

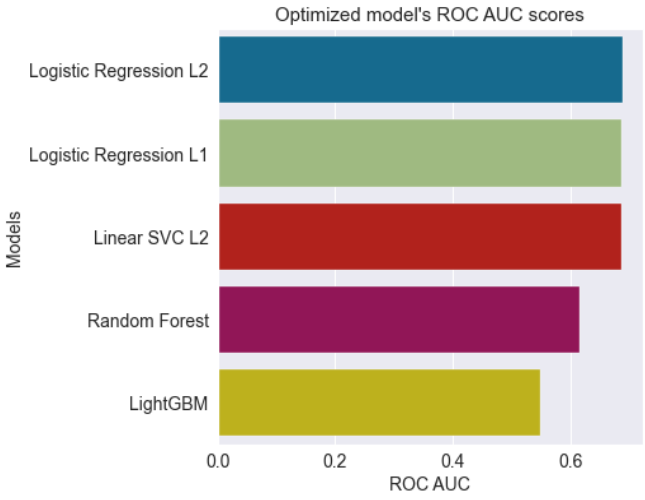


Figure 2. ROC AUC des modèles testés

2. Un modèle adapté aux assurances : fonction coût métier

Afin d'avoir un modèle sur-mesure adapté à la problématique métier, on implémente une fonction coût personnalisée. Cette fonction est utilisée sur la régression logistique pour recalculer les meilleurs hyperparamètres.

Le but est de calculer les gains ou pertes engendrés pour chaque possibilité de la matrice de confusion (vrai négatif (TN), faux négatif (FN), vrai positif (TP), faux positif (FN)) que l'on multiplie par le nombre de clients de chaque catégorie (Fig. 3).

		True class	
		0 (repaid)	**1 (default)**
Predicted class	**0 (repaid)**	TN * (+ interests)	FN * (-loan debt -interest debt)
	1 (default)	FP *(- interests)	TP: neutral

Figure 3. Matrice de confusion et construction de la fonction coût métier

Il y a donc 4 possibilités :

- TN (vrai négatif) : prêt correctement prédit comme remboursé, gain = montant des intérêts du prêt

$$TN\ gain = +\frac{crédit\ moyen * taux\ d'intérêt\ moyen}{100}$$

- FP (faux positif) : prêt prédit comme défaut, mais aurait été en réalité remboursé, perte = montant des intérêts du prêt

$$FP\ perte = -\frac{crédit\ moyen * taux\ d'intérêt\ moyen}{100}$$

- FN (faux négatif) : prêt prédit comme remboursé, en réalité le client fait défaut, perte = partie du prêt non remboursé + partie des intérêts non remboursés

$$FN\ perte = -\frac{pourcentage\ du\ crédit\ non\ remboursé * (crédit\ moyen + taux\ d'intérêt\ moyen)}{100}$$

- TP (vrai positif) : prêt prédit comme défaut, le client aurait effectivement fait défaut en réalité, ni gain ni perte car dans ce cas le prêt n'est pas accordé.

A partir des informations du dataset, on peut calculer que les clients faisant défaut sont en moyenne endettés de 38,6% de leur crédit, le crédit moyen est de 602648 $ et le taux d'intérêt moyen est de 4,81%. On obtient :

- $TN\ gain = +28\,983\ \$$
- $FP\ perte = -28\,983\ \$$
- $FN\ perte = -225\,751\ \$$

Ces valeurs sont utilisées comme coefficients (Fig. 4).

		True class	
		0 (repaid)	**1 (default)**
Predicted class	**0 (repaid)**	TN *(+28983)	FN * (-216428)
	1 (default)	FP * (-28983)	

Figure 4. Matrice de confusion et coefficients appliqués

On cherchera donc à maximiser :

$$Score\ function = \arg\max(TN * 28983 - FP * 28983 - FN * 225751)$$

Avec TN, FP et FN le nombre de clients true negative, false positive et false negative respectivement.

Avec cette fonction, le nouveau meilleur modèle sélectionné est une régression logistique L2 avec solver LBFGS et C = 21.544.

3. Interprétabilité des résultats

3.1. Interprétation globale

Pour mieux comprendre les résultats du modèle, on peut regarder les coefficients de la régression logistique appliqués à chaque variable (Fig. 5). Les coefficients les plus forts sont majoritairement appliqués aux différentes classes de ORGANIZATION_TYPE (type d'organisation pour laquelle le client travaille). Les sources externes, le type de travail et d'éducation (EXT_SOURCE_*, OCCUPATION_TYPE, NAME_EDUCATION_TYPE) ressortent également dans le haut du tableau. Les variables de bases relatives au crédit (montant, annuité, taux de paiement), ont également des coefficients importants. Enfin, le document 3 de l'application semble une feature importante pour déterminer si le client fera défaut ou non.

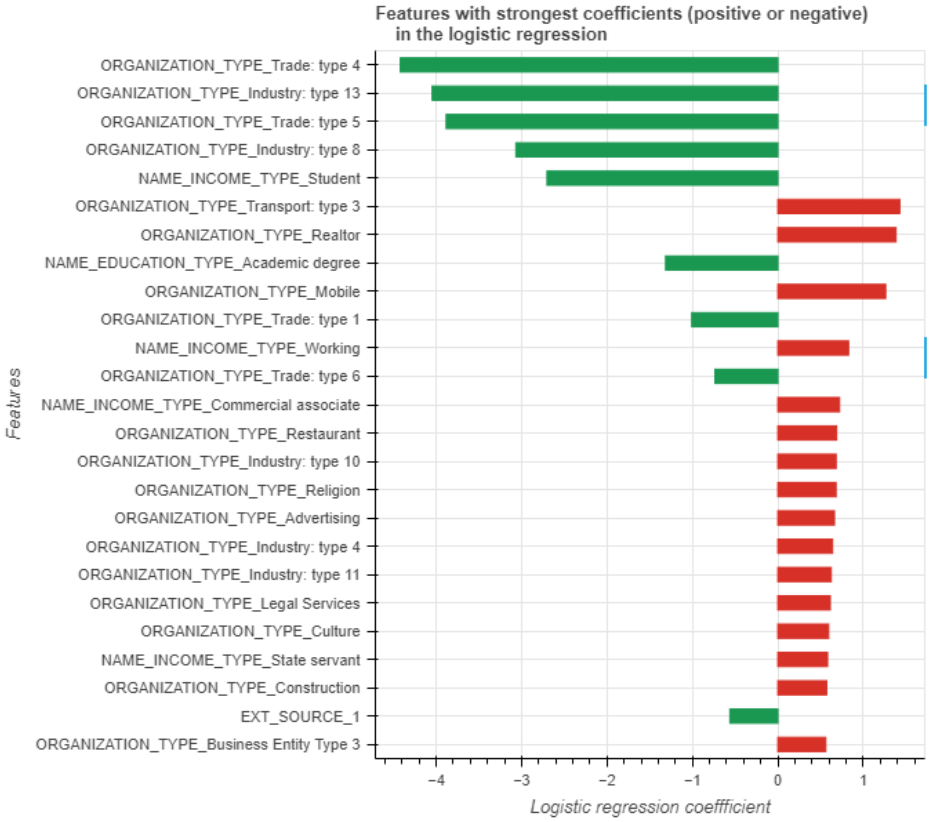


Figure 5. Top 25 des features avec les coefficients de régression logistique les plus importants

3.2. Interprétation locale

L'utilisation de la librairie shap permet de tester l'importance locale des features, c'est-à-dire pour un client en particulier. Shap fonctionne en enlevant une feature à la fois et en regardant l'impact sur la prédiction. Par exemple, pour le client ci-dessous (Fig. 6), les features les plus importantes sont les sources externes. Les sources 1 et 2 sont favorables au remboursement, alors que la source 2 tend à montrer un risque de défaut. Les autres features importantes et favorables au remboursement sont le montant du crédit, le type d'éducation, le type d'organisation. En revanche le taux de remboursement et certaines variables relatives à un ou plusieurs crédits précédents vont plutôt dans le sens d'un risque de défaut. Ces données permettent de mieux comprendre la prédiction de l'algorithme pour un client en particulier et d'orienter la recherche dans les données personnelles du client.

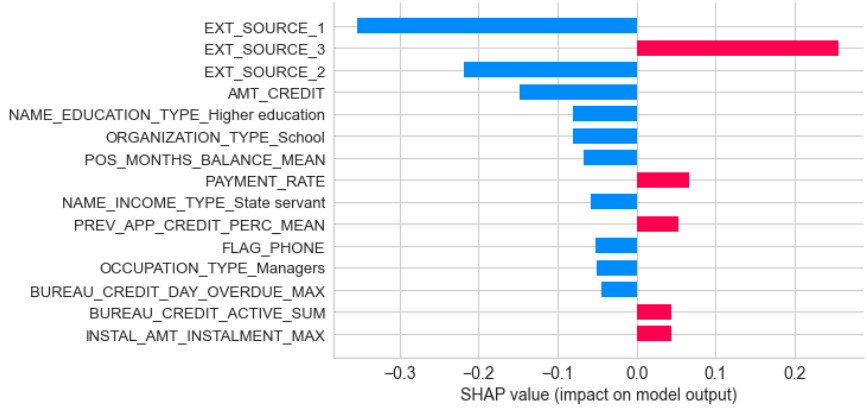


Figure 6. Top 15 des features les plus importantes pour un client

3. Limites et perspectives

La principale limite actuelle du modèle est qu'il ne comporte pas de traitement spécifique des valeurs manquantes. Les variables d'entrée (cf. Annexe) sont donc toutes indispensables à son bon fonctionnement.

Le feature engineering a été effectué un peu à l'aveugle, sans connaissances réelles du secteur banquaire. Les nouvelles variables créées restent des transformations basiques des variables de base (moyenne, minimum, maximum, écart-type et quelques taux).

La sélection des variables corrélées est ici basée sur le coefficient de corrélation de Spearman, mais une meilleure expertise métier permettrait d'effectuer un tri plus cohérent, notamment en sélectionnant les variables non corrélées qui ont le plus de sens pour les assureurs.

Enfin, le modèle comporte 129 features en entrée, 229 une fois les variables catégorielles transformées, et un certain nombre de ces features ne semblent au final que faiblement importantes dans la régression logistique. Une réduction du nombre de features nécessaires pourrait améliorer le temps de clacul de l'algorithme.

Les axes d'amélioration sont donc :

> La mise en place d'une étape de traitement des valeurs manquantes, adaptée aux gros datasets, avec une étude du biais introduit dans les données par une telle méthode (un ré-échantillonage post-traitement pour conserver les distributions du dataset de base pourrait être nécessaire).

> Un échange avec les experts métier permettrait :

- d'effectuer un feature engineering plus pertinent, par exemple avec le calcul d'indicateurs spécifiques au monde des assurances,
- de mieux sélectionner les variables corrélées à enlever ou conserver en fonction des connaissances métier,
- réduire le nombre de variables pour optimiser le temps de calcul en sélectionnant celles qui ont de faibles coefficients dans la régression logistique et un faible intérêt du point de vue métier.

ANNEXE : Données clients nécessaires à l'algorithme

Fichiers d'entrée	Variables	Variables après feature engineering
Application	CNT_CHILDREN	CNT_CHILDREN
	AMT_INCOME_TOTAL	AMT_INCOME_TOTAL
	AMT_CREDIT	AMT_CREDIT
	AMT_ANNUITY	AMT_ANNUITY
	REGION_POPULATION_RELATIVE	REGION_POPULATION_RELATIVE
	DAYS_BIRTH	DAYS_BIRTH
	DAYS_EMPLOYED	DAYS_EMPLOYED
	DAYS_REGISTRATION	DAYS_REGISTRATION
	DAYS_ID_PUBLISH	DAYS_ID_PUBLISH
	REGION_RATING_CLIENT	REGION_RATING_CLIENT
	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT_W_CITY
	HOUR_APPR_PROCESS_START	HOUR_APPR_PROCESS_START
	REG_REGION_NOT_LIVE_REGION	REG_REGION_NOT_LIVE_REGION
	REG_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION
	LIVE_REGION_NOT_WORK_REGION	LIVE_REGION_NOT_WORK_REGION
	REG_CITY_NOT_LIVE_CITY	REG_CITY_NOT_LIVE_CITY
	REG_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY
	LIVE_CITY_NOT_WORK_CITY	LIVE_CITY_NOT_WORK_CITY
	EXT_SOURCE_1	EXT_SOURCE_1
	EXT_SOURCE_2	EXT_SOURCE_2
	EXT_SOURCE_3	EXT_SOURCE_3
	OBS_30_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE
	DEF_30_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE
	OBS_60_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE
	DEF_60_CNT_SOCIAL_CIRCLE	DEF_60_CNT_SOCIAL_CIRCLE
	DAYS_LAST_PHONE_CHANGE	DAYS_LAST_PHONE_CHANGE
	AMT_REQ_CREDIT_BUREAU_HOUR	AMT_REQ_CREDIT_BUREAU_HOUR
	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_DAY
	AMT_REQ_CREDIT_BUREAU_WEEK	AMT_REQ_CREDIT_BUREAU_WEEK
	AMT_REQ_CREDIT_BUREAU_MON	AMT_REQ_CREDIT_BUREAU_MON
	AMT_REQ_CREDIT_BUREAU_QRT	AMT_REQ_CREDIT_BUREAU_QRT
	AMT_REQ_CREDIT_BUREAU_YEAR	AMT_REQ_CREDIT_BUREAU_YEAR
	NAME_CONTRACT_TYPE	NAME_CONTRACT_TYPE
	CODE_GENDER	CODE_GENDER
	FLAG_OWN_CAR	FLAG_OWN_CAR
	FLAG_OWN_REALTY	FLAG_OWN_REALTY
	NAME_INCOME_TYPE	NAME_INCOME_TYPE
	NAME_EDUCATION_TYPE	NAME_EDUCATION_TYPE
	NAME_FAMILY_STATUS	NAME_FAMILY_STATUS
	NAME_HOUSING_TYPE	NAME_HOUSING_TYPE
	OCCUPATION_TYPE	OCCUPATION_TYPE
	WEEKDAY_APPR_PROCESS_START	WEEKDAY_APPR_PROCESS_START
	ORGANIZATION_TYPE	ORGANIZATION_TYPE
	FLAG_MOBIL	FLAG_MOBIL
	FLAG_EMP_PHONE	FLAG_EMP_PHONE

	FLAG_WORK_PHONE	FLAG_WORK_PHONE
	FLAG_CONT_MOBILE	FLAG_CONT_MOBILE
	FLAG_PHONE	FLAG_PHONE
	FLAG_EMAIL	FLAG_EMAIL
	FLAG_DOCUMENT_2	FLAG_DOCUMENT_2
	FLAG_DOCUMENT_3	FLAG_DOCUMENT_3
	FLAG_DOCUMENT_4	FLAG_DOCUMENT_4
	FLAG_DOCUMENT_5	FLAG_DOCUMENT_5
	FLAG_DOCUMENT_6	FLAG_DOCUMENT_6
	FLAG_DOCUMENT_7	FLAG_DOCUMENT_7
	FLAG_DOCUMENT_8	FLAG_DOCUMENT_8
	FLAG_DOCUMENT_9	FLAG_DOCUMENT_9
	FLAG_DOCUMENT_10	FLAG_DOCUMENT_10
	FLAG_DOCUMENT_11	FLAG_DOCUMENT_11
	FLAG_DOCUMENT_12	FLAG_DOCUMENT_12
	FLAG_DOCUMENT_13	FLAG_DOCUMENT_13
	FLAG_DOCUMENT_14	FLAG_DOCUMENT_14
	FLAG_DOCUMENT_15	FLAG_DOCUMENT_15
	FLAG_DOCUMENT_16	FLAG_DOCUMENT_16
	FLAG_DOCUMENT_17	FLAG_DOCUMENT_17
	FLAG_DOCUMENT_18	FLAG_DOCUMENT_18
	FLAG_DOCUMENT_19	FLAG_DOCUMENT_19
	FLAG_DOCUMENT_20	FLAG_DOCUMENT_20
	FLAG_DOCUMENT_21	FLAG_DOCUMENT_21
	DAYS_EMPLOYED/DAYS_BIRTH =	DAYS_EMPLOYED_PERC
	AMT_INCOME_TOTAL/AMT_INCOME_CREDIT =	INCOME_CREDIT_PERC
	AMT_INCOME_TOTAL/CNT_FAM_MEMBERS =	INCOME_PER_PERSON
	AMT_ANNUITY/AMT_INCOME_TOTAL =	ANNUITY_INCOME_PERC
	AMT_ANNUITY/AMT_CREDIT =	PAYMENT_RATE
Bureau	DAYS_CREDIT	BUREAU_DAYS_CREDIT_MIN
		BUREAU_DAYS_CREDIT_MAX
		BUREAU_DAYS_CREDIT_MEAN
	DAYS_CREDIT_ENDDATE	BUREAU_DAYS_CREDIT_ENDDATE_MAX
		BUREAU_DAYS_CREDIT_ENDDATE_MEAN
	CREDIT_DAY_OVERDUE	BUREAU_CREDIT_DAY_OVERDUE_MAX
	AMT_CREDIT_SUM	BUREAU_AMT_CREDIT_SUM_MAX
	AMT_CREDIT_SUM_DEBT	BUREAU_AMT_CREDIT_SUM_DEBT_MEAN
	AMT_CREDIT_SUM_OVERDUE	BUREAU_AMT_CREDIT_SUM_OVERDUE_MEAN
	AMT_CREDIT_SUM_LIMIT	BUREAU_AMT_CREDIT_SUM_LIMIT_SUM
	CNT_CREDIT_PROLONG	BUREAU_CNT_CREDIT_PROLONG_SUM
	CREDIT_ACTIVE	BUREAU_CREDIT_ACTIVE_MEAN
		BUREAU_CREDIT_ACTIVE_SUM
	CREDIT_CURRENCY	BUREAU_CREDIT_CURRENCY_MEAN
	CREDIT_TYPE	BUREAU_CREDIT_TYPE_MEAN
		BUREAU_CREDIT_TYPE_MAX
Bureau balance	MONTHS_BALANCE	BUREAU_MONTHS_BALANCE_SIZE_SUM

Installments	DAYS_INSTALLMENT-DAYS_ENTRY_PAYMENT If < 0 → 0 Mean, max and sum for all installments	INSTAL_DBD_MAX
		INSTAL_DBD_MEAN
		INSTAL_DBD_SUM
	AMT_INSTALMENT	INSTAL_AMT_INSTALMENT_MAX
	AMT_PAYMENT	INSTAL_AMT_PAYMENT_MIN
	DAYS_ENTRY_PAYMENT	INSTAL_DAYS_ENTRY_PAYMENT_MAX
	(AMT_PAYMENT/AMT_INSTALLMENT)MAX =	INSTAL_PAYMENT_PERC_MAX
	(AMT_INSTALLMENT-AMT_PAYMENT)MEAN =	INSTAL_PAYMENT_DIFF_MEAN
Previous application	AMT_ANNUITY	PREV_AMT_ANNUITY_MIN
		PREV_AMT_ANNUITY_MAX
		PREV_AMT_ANNUITY_MEAN
	AMT_APPLICATION	PREV_AMT_APPLICATION_MIN
	AMT_CREDIT	PREV_AMT_CREDIT_MIN
	AMT_DOWN_PAYMENT	PREV_AMT_DOWN_PAYMENT_MIN
		PREV_AMT_DOWN_PAYMENT_MEAN
	AMT_GOODS_PRICE	PREV_AMT_GOODS_PRICE_MIN
	HOUR_APPR_PROCESS_START	PREV_HOUR_APPR_PROCESS_START_MIN
		PREV_HOUR_APPR_PROCESS_START_MAX
		PREV_HOUR_APPR_PROCESS_START_MEAN
	DAYS_DECISION	PREV_DAYS_DECISION_MIN
		PREV_DAYS_DECISION_MAX
	CNT_PAYMENT	PREV_CNT_PAYMENT_MEAN
	FLAG_LAST_APPL_PER_CONTRACT	PREV_FLAG_LAST_APPL_PER_CONTRACT_MEAN
	NAME_CASH_LOAN_PURPOSE	PREV_NAME_CASH_LOAN_PURPOSE_MEAN
	NAME_CONTRACT_STATUS	PREV_NAME_CONTRACT_STATUS_MAX
	CODE_REJECT_REASON	PREV_CODE_REJECT_REASON_MEAN
	NAME_YIELD_GROUP	PREV_NAME_YIELD_GROUP_MEAN
	PRODUCT_COMBINATION	PREV_PRODUCT_COMBINATION_MEAN
	AMT_APPLICATION/AMT_CREDIT Min, max and mean for all previous app	PREV_APP_CREDIT_PERC_MIN
		PREV_APP_CREDIT_PERC_MAX
		PREV_APP_CREDIT_PERC_MEAN
POS CASH	MONTHS_BALANCE	POS_MONTHS_BALANCE_MAX
		POS_MONTHS_BALANCE_MEAN
	SK_DPD	POS_SK_DPD_MAX
	SK_DPD_DEF	POS_SK_DPD_DEF_MAX
	NAME_CONTRACT_STATUS	POS_NAME_CONTRACT_STATUS_MIN
		POS_NAME_CONTRACT_STATUS_MAX
Credit card balance	No variables from this dataset was used.	