

Opis projektu zaliczeniowego do zajęć

nr przedmiotu: 220310

„Credit Scoring - automatyzacja procesu biznesowego”

Ogólny opis zadań

Celem projektu jest zapoznanie studenta z budową modeli skoringowych oraz z trudnościami w ich implementacji. Projekt pozwala także zaobserwować, jak istotnie wpływają na wyniki wnioski odrzucone, lub ogólnie jak zaburza estymację stary proces. Wreszcie finalne raporty i wyniki pozwalają wyrobić intuicję i rozwinąć doświadczenie wpływu listy reguł silnika decyzyjnego ze wskaźnikami finansowymi całego procesu akceptacji, który składa się z dwóch procesów: akwizycji – kredytu ratального i sprzedaży związanej – kredytu gotówkowego. Jest to unikatowe doświadczenie budowy pełnego procesu w pełni automatycznego, gdzie podejmowanie decyzji następuje w oparciu o modele predykcyjne całkowicie automatycznie. Jest to bardzo ważna nauka dla studenta, gdyż w najbliższym czasie tylko takie procesy będą w biznesie i trzeba umieć je konfigurować i utrzymywać.

Lista zadań

Projekt powinien wykonywać pięcio-osobowy zespół, inna liczba studentów w projekcie wymaga konsultacji z prowadzącym tak, by każdy uczestnik miał podobne zaangażowanie w projekcie.

Celem projektu jest zbudowanie optymalnej strategii procesu akceptacji, by zmaksymalizować finalne zyski z obu produktów w okresie dostępnym 1975-87. Należy polepszyć strategię wzorcowe, które przynoszą zysk 663 327 PLN w okresie 1975-87.

Lista pojedynczych zadań sprowadza się do:

1. Ulepszenia jakiegokolwiek z etapów budowy modelu, poprzez napisanie własnego kodu w SAS 4GL. Może to być np.: policzenie dodatkowej statystyki, zrobienie dodatkowego raportu graficznego itp. Można też poprawiać lub proponować dodatkowe kody w Pythonie do budowy karty skoringowej lub innych elementów budowy modeli, włączając w to kody modeli AI, lub wyrafinowane metody selekcji zmiennych. Mile widziane są też elementy XAI, czyli interpretowania modeli, tworzenia dobrej i zwizualizowanej dokumentacji modelu AI.
2. Zbudowania modelu ryzyka mierzącego prawdopodobieństwo zdarzenia default₁₂=1 dla kredytu ratального (product='ins'), oznaczenie modelu PD Ins.
3. Zbudowania modelu ryzyka mierzącego prawdopodobieństwo zdarzenia default₁₂=1 dla kredytu gotówkowego (product='css'), oznaczenie modelu PD Css.

4. Zbudowania modelu ryzyka mierzącego prawdopodobieństwo zdarzenia default_cross12=1 dla kredytu gotówkowego w momencie aplikowania o kredyt ratalny, oznaczenie modelu PD Css Cross.
5. Zbudowania modelu marketingowego mierzącego prawdopodobieństwo zdarzenia cross_response=1 w momencie aplikowania o kredyt ratalny i/lub gotówkowy, oznaczenie modelu PR Css Cross.
6. Stworzenia reguł silnika decyzyjnego i wdrożenia wszystkich opisanych modeli. Uruchomienia strategii procesu akceptacji kredytowej z nowymi regułami i wyliczenia finalnych raportów zysków i strat dla okresu 1975-87.

Modele mogą być budowane albo przez ASB_SAS, albo przez ASB_PYTHON. Dopuszcza się, że jeden model może być modelem AI, czyli nie kartą scoringową, przy czym musi być przedstawiony dobry i interpretowalny opis modelu. Wymaga to też napisanie własnego narzędzia scoring_code.sas, który będzie z poziomu SAS uruchamiał klasę Pythona.

Można też budować więcej niż cztery modele poprzez właściwą segmentację portfela wnioskującego.

Podział zadań może być dobrany dowolnie, przy jedynym ograniczeniu: pierwsze zadanie nie może być jedynym zadaniem jednego studenta. Powinno się podzielić tak, by każdy był podobnie zaangażowany w projekcie.

Każde zadanie powinno zakończyć się odpowiednią dokumentacją w wersji elektronicznej, potem wysłanej email'em do prowadzącego.

Projekt podlega obronie, podczas której oceniane są wszelkie wyniki, a dokumentacja strategii i modeli w wersji elektronicznej stanowi najważniejszy materiał do dyskusji i finalnej oceny. Obrona polega na przekonywaniu prowadzącego zajęcia do otrzymanych wyników, danej postaci modelu, nierówności w kategoriach, wzorów w regułach strategii itp. Liczyć się tu będzie głównie lista argumentów poruszanych przez studentów. Najlepiej, aby każdy z uczestników projektu opowiadał o swojej części.

Dalsze rozdziały opisują w szczegółach wymienione wyżej zadania. Dodatkowo dwa rozdziały pomagają uruchomić pierwszy raz projekt ze strategią wzorcową oraz zbudować nową.

Punktacja i finalna ocena projektu

Projekt jest oceniany przez prowadzącego następującymi kryteriami (poniżej podano maksymalną liczbę punktów z danego kryterium):

| Kategoria | Punkty |
|---|-----------|
| Wynik finansowy: do 6pt. | 6 |
| Metoda argumentowania za przyjętą strategią akceptacji: do 5 pt. | 5 |
| Metoda prezentacji i argumentacji dobrych i mocnych stron modeli: do 4pt. | 4 |
| Uchybienia wskaźników modeli: do 3pt. | 3 |
| Uchybienia raportów: do 2pt. | 2 |
| Suma | 20 |

Jak uruchomić projekt ze strategią wzorcową

W katalogu ...\\CS-AUT\\software\\PROCSS_SIMULATION\\process\\codes\\ znajduje się wzorcowy kod `decision_engine.sas` (100% akceptacji), który powinien być modyfikowany przez studenta. Kod ten odwołuje się do podkatalogów katalogu ...\\CS-AUT\\software\\PROCSS_SIMULATION\\process\\calibration\\, w których znajdują się pliki konfiguracyjne zbudowanych modeli, które też student powinien wymienić na nowe, budowane przez zespół projektowy. W każdym katalogu modelu znajduje się najważniejszy kod o nazwie `scoring_code.sas`, który przez instrukcję `%include` jest dołączany w kodzie `decision_engine.sas`. W kodzie tym cały procesowany miesiąc jest skorowany czterema modelami, doliczając nie tylko wartości ocen punktowych, ale także wartości prawdopodobieństw modelowanych zdarzeń. Wreszcie finalny dataset umieszczony na końcu tego kodu wylicza właściwe reguły definiując finalne decyzje akceptacji oraz przyczyny odrzutów.

Cały proces uruchamia się plikiem `batch.bat` z katalogu ...\\CS-AUT\\software\\PROCSS_SIMULATION\\codes\\. Polecenie batchowe odwołuje się do kodu `all_contents.sas`. Finalny raport ze wskaźnikami finansowymi tworzy się w katalogu ...\\CS-AUT\\software\\PROCSS_SIMULATION\\process\\reports\\ i nosi nazwę `profit_1975_1987.html`.

Jak kalibrować modele do PD i PR oraz jak ustalić punkty odcięcia

Jeśli zbudowane są już wszystkie modele, których może być więcej niż cztery. To każdy model, ze swoim kodem `scoring_code.sas`, powinien być skopiowany do właściwego podkatalogu w katalogu ...\\CS-AUT\\software\\PROCSS_SIMULATION\\process\\calibration\\. W tym też katalogu znajduje się kod `calibration.sas`, który na wzorcowej strategii pokazuje, jak wyznaczyć prawdopodobieństwa modelowanych zdarzeń oraz jak wyznaczyć właściwe punkty odcięcia. Wiele kodów należy tu od komentować lub zmieniać. Podstawowa idea wyznaczenia parametru PD lub PR polega na uruchomieniu właściwej procedury logistic w SAS z opcją `outest=bety`, a następnie skopiować ze zbioru `bety` ich wartości do wzoru z funkcją `exp()` w odpowiednie miejsce. Wyznaczenie punktów odcięcia jest ciekawym wyzwaniem. Można do tego zadania podchodzić jedno-... lub dwu-produktowo. W drugim przypadku wykorzystana jest procedura `rank` i analizuje się różne grupy wniosków ze względu na pierwszy i drugi produkt na bazie parametrów prawdopodobieństw.

Budowa modelu i dokumentacja

Model należy zbudować na właściwie dobranej próbie. Zawsze należy budować na danych pochodzących z okresu 1975-87 (warunek w SAS 4GL `'197501'<=period<='198712'`), oraz na odpowiednio dobranych kryteriach produktu (typu `product='css'` lub `product='ins'`) i decyzji (`decision='A'`). W kodach Python też są te warunki ujęte. Zmienne ABT budowane są tylko na podstawie informacji o zaakceptowanych wnioskach klienta w procesie i tym samym tylko na zaakceptowanych historiach kredytów przez nasz proces, to może powodować obciążenie estymacji i budowy modeli, które dokonują się na portfelu inaczej akceptowanym. Każda przeprocesowana strategia tworzy zbiór `abt_app.sas7bdat` w katalogu ...\\CS-AUT\\software\\PROCSS_SIMULATION\\process\\data\\, który potem należy skopiować do właściwego katalogu modelowania np. do ...\\CS-AUT\\software\\ASB_SAS\\inlib\\ lub do ...\\CS-AUT\\software\\ASB_PYTHON\\. Dobór strategii początkowej do budowy modelu jest jednym z zadań tego etapu.

Finalny model z jego plikami parametryzacyjnymi i raportami należy skopiować do wielu właściwych miejsc:

1. Skopiować właściwe pliki do katalogu procesowania nowej strategii czyli np. do ...\\CS-AUT\\software\\PROCSS_SIMULATION\\process\\calibration\\model_ins_risk\\.
2. Skopiować do właściwych struktur jak w katalogu ...\\CS-AUT\\materials_all\\example_of_project_documentation\\.
3. Odpowiednio zmodyfikować i wypełnić plik model_documentation_PD_INS.xlsx, stanowiący materiał do obrony z katalogu ...\\CS-AUT\\materials_all\\example_of_project_documentation\\example_of_model_SAS\\ lub wykorzystać kod Python I przygotować dokumentację taką, jak dla modelu budowanego w Pythonie: katalog ...\\CS-AUT\\materials_all\\example_of_project_documentation\\example_of_model_PYTHON\\

Budowa strategii i dokumentacja

Na początku trzeba zebrać wszystkie modele, umieścić kody do skorowania w odpowiednich katalogach i wykonać kalibrację. Potem zdefiniować właściwe reguły i zapisać je w kodzie decision_engine.sas. Nie wolno zmieniać reguły „998 not active customer”. Reguły mogą być zależne od czasu, mogą być bardziej lub mniej skomplikowane, byleby opierały się na dostępnej informacji w momencie aplikowania wniosku, czyli w oparciu o dane z ABT z odpowiedniego miesiąca procesowanego i prawdopodobieństwa z modeli. Po uruchomieniu procesowania należy skopiować pliki wynikowe do katalogu jak w przykładzie ...\\CS-AUT\\materials_all\\example_of_project_documentation\\example_of_strategy\\ oraz zmodyfikować plik strategy_documentation.xlsx i przesłać go razem z dokumentacjami modeli do prowadzącego.

Ulepszone kody i dokumentacja

Wystarczy napisać kod i go przesłać prowadzącemu w pakiecie ze strategią i dokumentacjami modeli. Przykład znajduje się w katalogu ...\\CS-AUT\\materials_all\\example_of_project_documentation\\example_of_extra_code\\.