Обнаружение мошеннических банковских операций

Групповой проект

Жоголева Елена
Тенякова Роза
Матюков Петр
Маслоед Ирина

Описание задачи

Предсказать факты мошеннических транзакций

Дано: набор данных для машинного обучения, которой содержит данные о реальных транзакциях электронной коммерции Vesta.

Первичный анализ данных. Состав

Данные для обучения

- 433 признака
- 590540 наблюдений
- 45.1% пропущено
- 1.9 Гб памяти
- 388 вещественных признака
- 45 категориальных признака

Тестовые данные

- 433 признака
- 506691 наблюдений
- 41.1% пропущено
- 1.6 Гб памяти
- 388 вещественных признака
- 45 категориальных признака

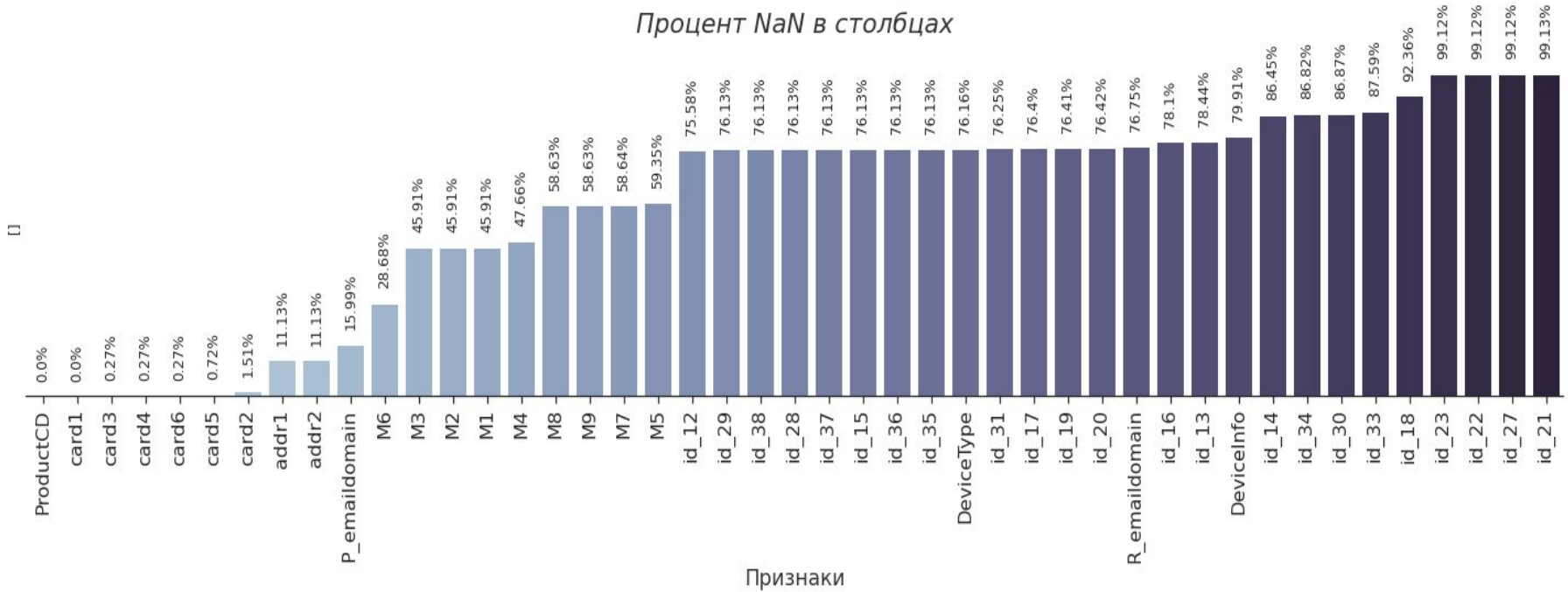
Состав данных

- ✓ дата, сумма, ID транзакции
- ✓ информация о банковской карте
- ✓ адрес проведения транзакции

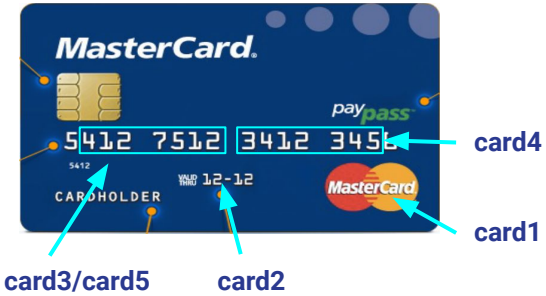
Особенности

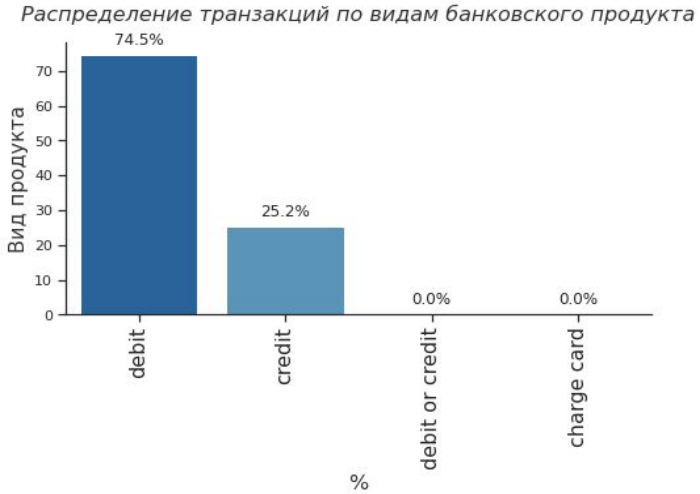
- ✓ много пропущенных значений
- ✓ много зашифрованных значений

Первичный анализ. Категориальные признаки

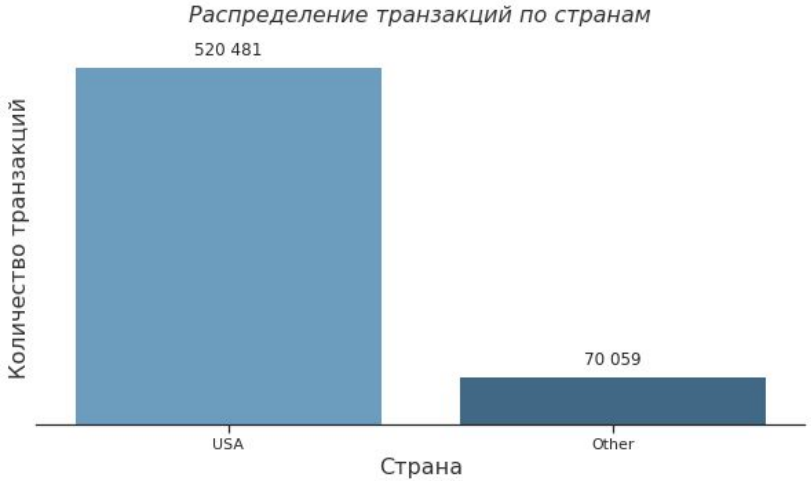


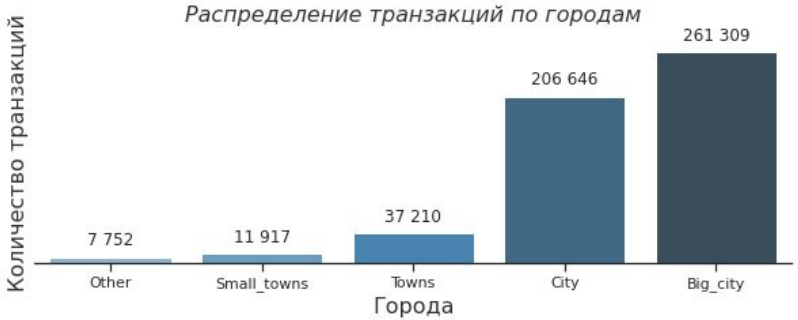
Первичный анализ. Информация о карте

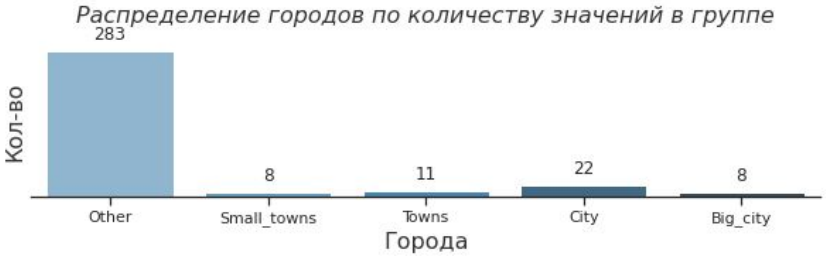




Первичный анализ. Данные об адресе

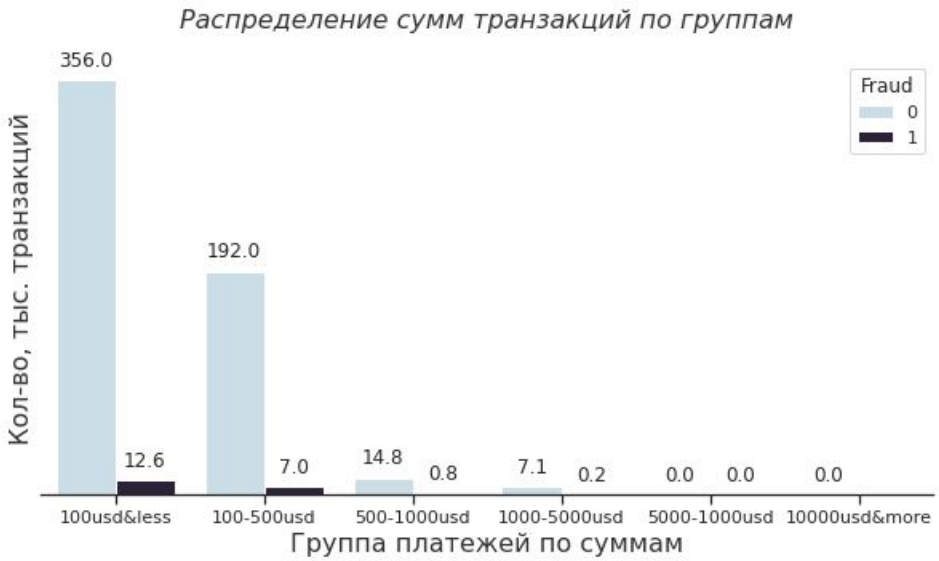






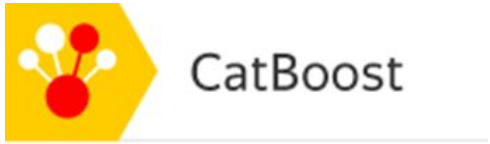
Первичный анализ. Сумма транзакции

✓ **Самые частые транзакции - до 100 USD**



✓ **3,4% - мошеннические операции**

Выбор алгоритма для модели предсказания



Почему мы выбрали CatBoost?

- эффективен на больших объемах данных и признаков;
- работает с категориальными признаками “из коробки”;
- best practice в ML;
- применяет кросс-валидацию и подбор параметров “из коробки”;
- оптимизирует время обучения за счет использования GPU;
- содержит подробную документацию;

Построение модели. Шаги

Тайминг: 1.5 - 2 часа

№	Изменения
Baseline	Набор данных без изменений
Step 1	Удаление признаков по результатам анализа важности признаков и корреляции
Step 2	Добавление 3х агрегированных столбцов
Step 3	Удаление около 150 столбцов “V_”
Step 4	Добавление столбца с группами по суммам операций
Step 5	Добавление столбцов “день” и “час” на основе признака “ TransactionDT”
Step 6	Blending&Stacking

Анализ полученных результатов

BaseLine → Step 1

Много пропущенных значений

Решение:

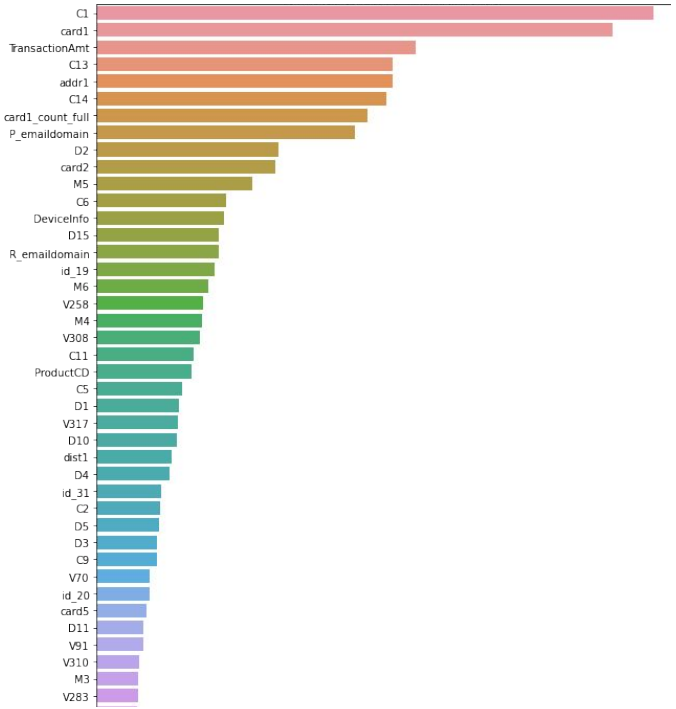
удалить столбцы с пропущенными, более чем 95% значений:

Много признаков с нулевым вкладом в модель и высокой корреляцией с другими

Решение:

удалить признаки с более чем 80% пропущенных значений (id_)

Важность признаков



Анализ полученных результатов

Step2 → Step 3

Связки признаков

card1-card6, addr1-addr2,

dist1-dist2 уникальны

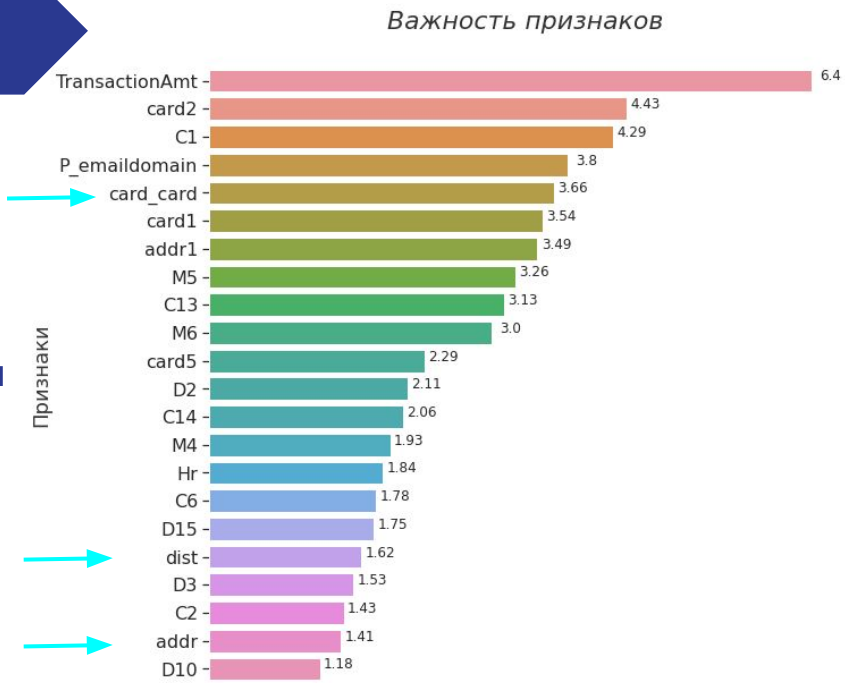
Решение:

Создание новых признаков card, addr, dist,

Много признаков с нулевым вкладом в модель и высокой корреляцией с другими

Решение:

удалить признаки с вкладом в результат менее чем 15%



Анализ полученных результатов

Step 4

Создание новых признаков

Решение:

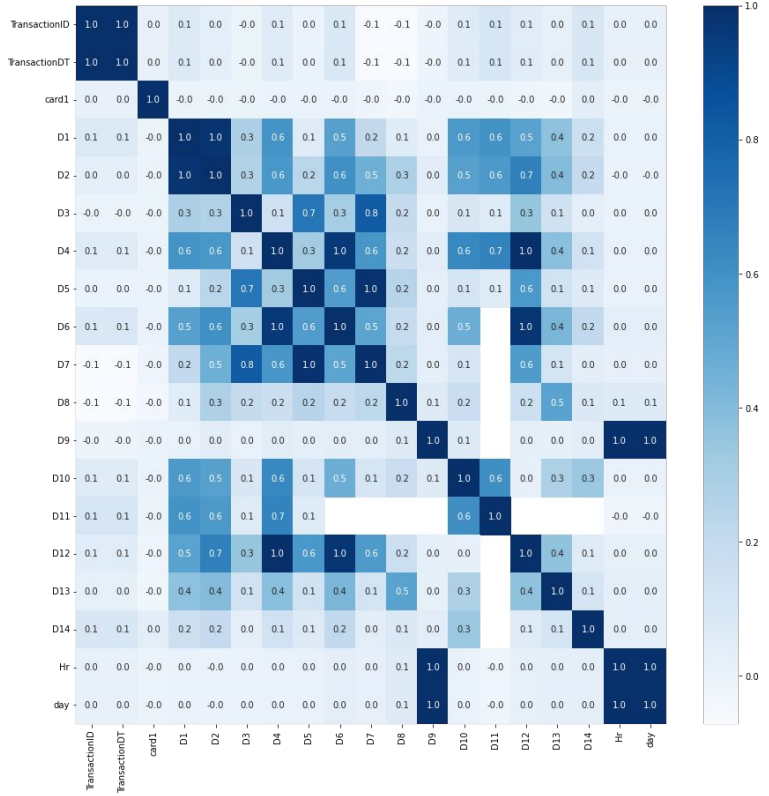
Добавили признак с ранжированием суммы операций

Step 5

Создание новых признаков

Решение:

Добавление признаков D и Hr



Созданные поля Hr и Day не коррелируют с другими.

Анализ полученных результатов

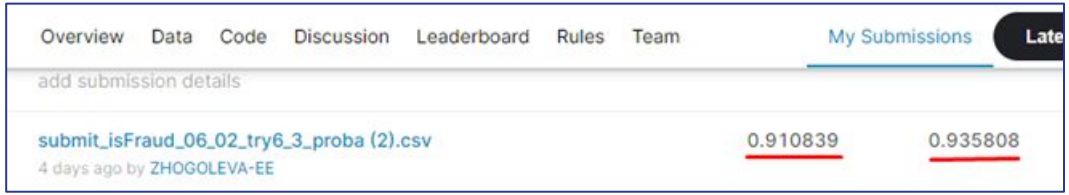
Step 6

Blending&Stacking

Модель	Kaggle private	Kaggle public
Базовый Catboost (минимальная очистка данных, без настроек)	0.903084	0.928801
Logistic Regression блендинг + Polynomial Features + 2 Random Forest	0.893371	0.923297
Logistic Regression блендинг + Polynomial Features + 2 Random Forest + 2 Catboost	0.913425	0.939829
Logistic Regression блендинг + Polynomial Features + 2 Random Forest + 5 Catboost	0.913484	0.939870
Logistic Regression блендинг + Polynomial Features + 5 Random Forest + 5 Catboost	0.909794	0.937535
Logistic Regression блендинг + Polynomial Features + 5 Catboost	0.910282	0.936811

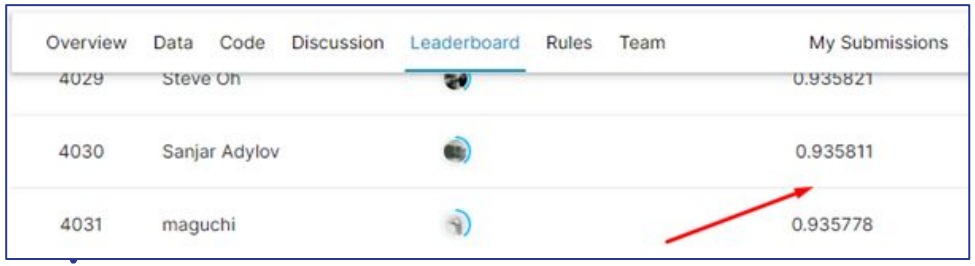
Достигнутый результат

Лучший результат



Личный рейтинг
- ✓ 3190 место из 6355
- ✓ лучше, чем 50%

Публичный рейтинг
- ✓ 4031 место из 6355
- ✓ лучше, чем 35%



Выводы

- Работа в команде более эффективна. Можно параллельно проверять разные гипотезы.

- Обезличенные данные и большое количество пропусков ухудшают качество модели - грамотная работа с пропущенными значениями и поиск возможности восстановить пропущенные значения - эффективны. Однако feature engineering затруднен тем, что невозможно оценить смысл признаков.

- Blanding&Stacking - хорошее решение для увеличения эффективности.

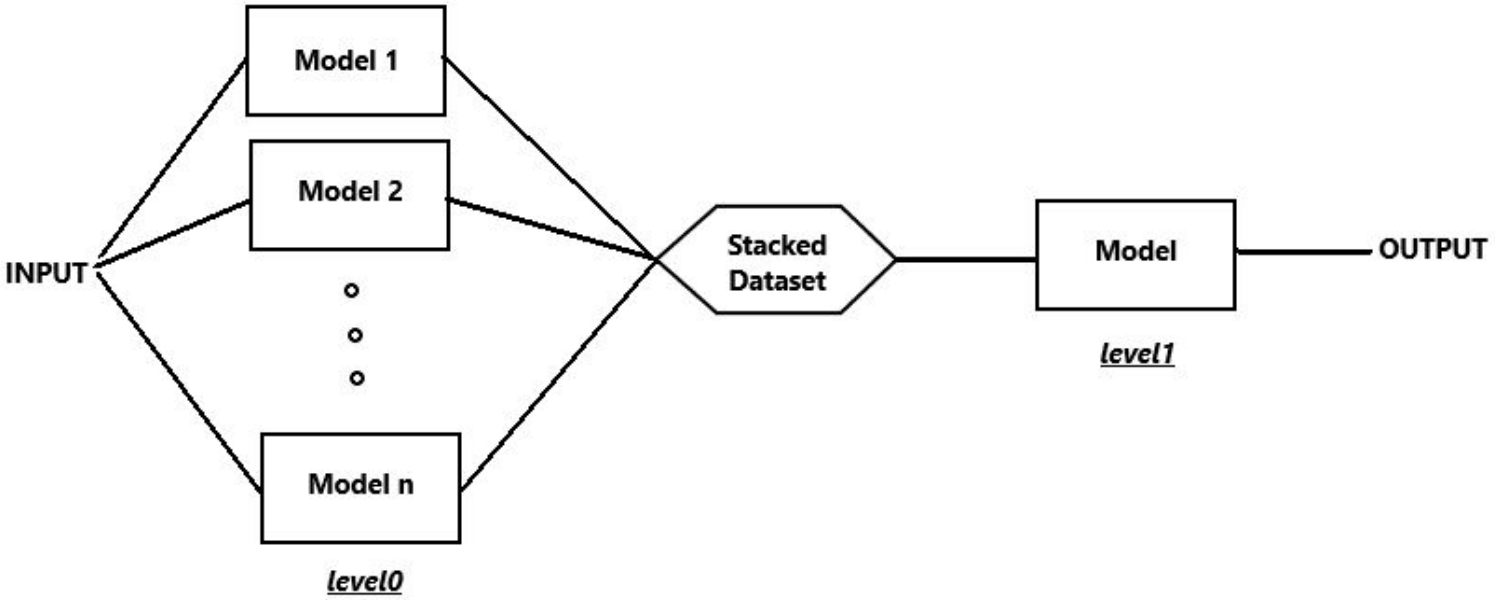
Используемые материалы

- соревнование IEEE-CIS Fraud Detection | Kaggle
- описание данных IEEE-CIS Fraud Detection | Kaggle
- CatBoost documentation CatBoost
- анализ признаков shap slundberg/shap: A game theoretic approach to explain the output of any machine learning model. (github.com)
- статья по приемам feature engineering IEEE-CIS Fraud Detection | Kaggle
- Blanding&Stacking https://machinelearningmastery.com/blending-ensemble-machine-learning-with-python/

Спасибо за внимание!

Приложение 1. Blending & Stacking

Цель: на базовом датасете реализовать ансамбль из нескольких моделей и сравнить результат с традиционным EDA+feature engineering

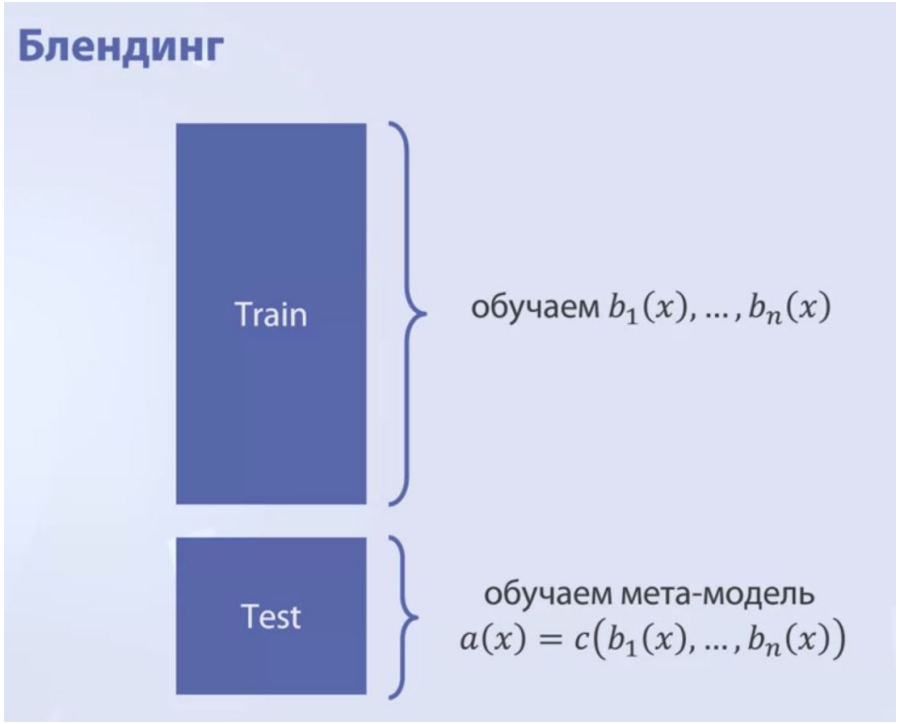


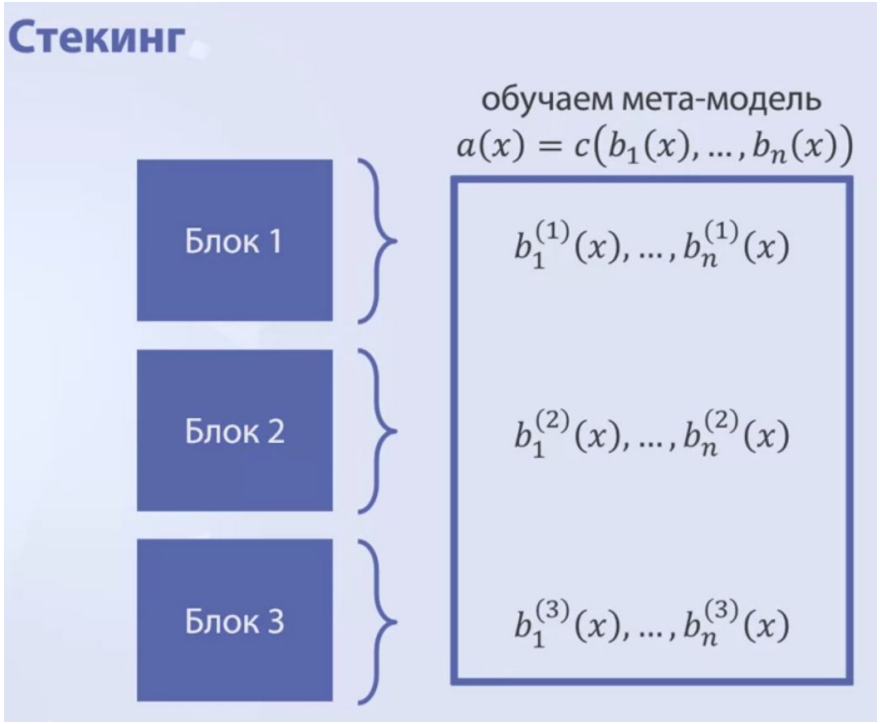
Приложение 1. Blending & Stacking простой

На Kaggle часто применяют смешивание результатов из submission от разных моделей. Смешивают прямо csv файлы сабмитов.

TIP! Результаты проверяют на корреляцию Пирсона и стараются смешивать некоррелируемые результаты. Это дает лучшую оценку.

Приложение 1.Blending & Stacking - с обучением





Приложение 2. Blending & Stacking - результаты

В качестве начальной базовой модели был выбран Catboost. В качестве дополнительной - случайный лес, потому что он показывает хорошие результаты в классификации, сравнимые с бустинговыми моделями.

Модель	Kaggle private	Kaggle public
Базовый Catboost (минимальная очистка данных, без настроек)	0.903084	0.928801
Forest 500	0.881892	0.907618
Блендинг 0.5+0.5	0.892148	0.916442
Блендинг 0.6+0.4	0.884932	0.909389
Блендинг 0.7+0.3	0.885219	0.909625

Приложение 2. Blending & Stacking - результаты

Из-за того, что простой блендинг не обнадежил, было решено попробовать блендинг с обучением. В качестве мета-модели опробованы Catboost, Random Forest, Logistic Regression и по результатам выбрана последняя.

Модель	Kaggle private	Kaggle public
Базовый Catboost (минимальная очистка данных, без настроек)	0.903084	0.928801
Logistic Regression блендинг	0.906407	0.929961
Logistic Regression блендинг + Polynomial Features	0.906649	0.930168

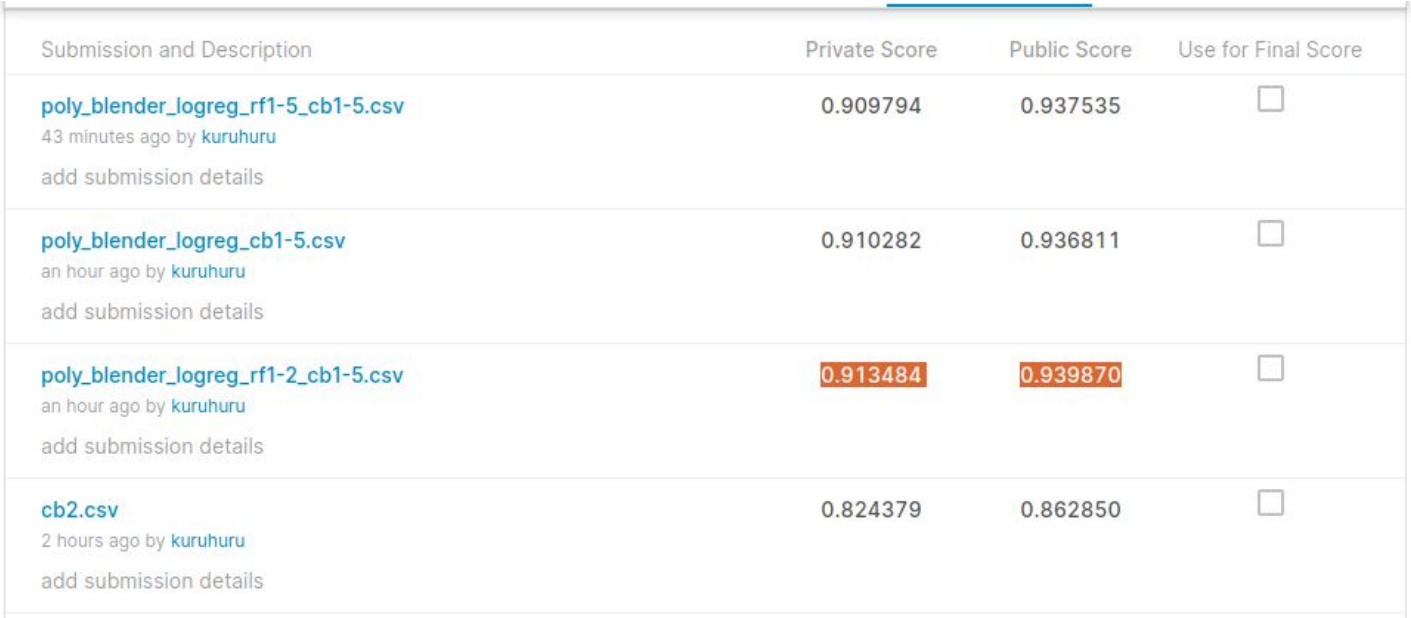
Приложение 2. Blending & Stacking результаты

Логично теперь попробовать увеличить количество моделей. Причем стараясь их варьировать по гипер-параметрам и random_state

Модель	Kaggle private	Kaggle public
Базовый Catboost (минимальная очистка данных, без настроек)	0.903084	0.928801
Logistic Regression блендинг + Polynomial Features + 2 Random Forest	0.893371	0.923297
Logistic Regression блендинг + Polynomial Features + 2 Random Forest + 2 Catboost	0.913425	0.939829
Logistic Regression блендинг + Polynomial Features + 2 Random Forest + 5 Catboost	0.913484	0.939870
Logistic Regression блендинг + Polynomial Features + 5 Random Forest + 5 Catboost	0.909794	0.937535
Logistic Regression блендинг + Polynomial Features + 5 Catboost	0.910282	0.936811

Приложение 2. Blending & Stacking - выводы

Вывод Блендинг, в дополнение к EDA и Feature engineering, может улучшить итоговый результат на Kaggle. Но это потребует подобрать несколько отличающихся друг от друга моделей и провести ряд экспериментов.



Submission and Description	Private Score	Public Score	Use for Final Score
poly_blender_logreg_rf1-5_cb1-5.csv 43 minutes ago by kuruhuru add submission details	0.909794	0.937535	☐
poly_blender_logreg_cb1-5.csv an hour ago by kuruhuru add submission details	0.910282	0.936811	☐
poly_blender_logreg_rf1-2_cb1-5.csv an hour ago by kuruhuru add submission details	0.913484	0.939870	☐
cb2.csv 2 hours ago by kuruhuru add submission details	0.824379	0.862850	☐

Приложение 3. Тайминг на обучение

- CatBoost без использования GPU - 1,5 - 2 часа, в зависимости от количества признаков.

```
995:    test: 0.9773506 best: 0.9773884 (985)   total: 1h 21m 13s       remaining: 19.6s
996:    test: 0.9773613 best: 0.9773884 (985)   total: 1h 21m 17s       remaining: 14.7s
997:    test: 0.9773556 best: 0.9773884 (985)   total: 1h 21m 21s       remaining: 9.78s
998:    test: 0.9773562 best: 0.9773884 (985)   total: 1h 21m 26s       remaining: 4.89s
999:    test: 0.9773517 best: 0.9773884 (985)   total: 1h 21m 31s       remaining: 0us

bestTest = 0.9773883913
bestIteration = 985

Shrink model to first 986 iterations.
<catboost.core.CatBoostClassifier at 0x7f83e47efb90>
```

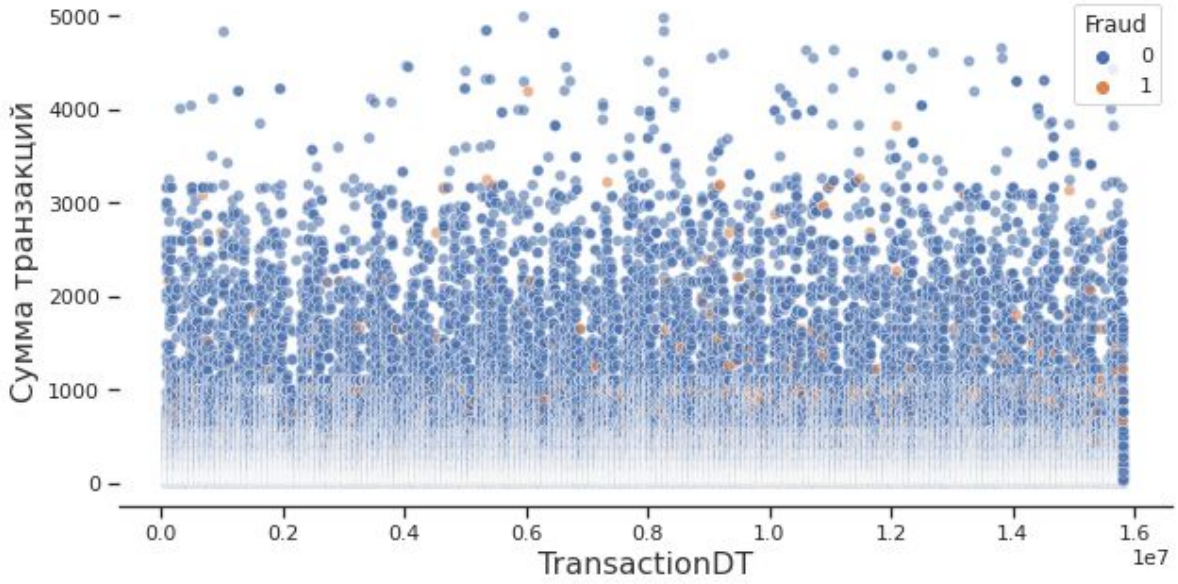
- CatBoost без использования GPU - 1,5 - 2 часа, в зависимости от количества признаков.

```
997:    learn: 0.9726263        test: 0.9711705 best: 0.9711780 (996)   total: 6m 31s   remaining: 785ms
998:    learn: 0.9726425        test: 0.9711669 best: 0.9711780 (996)   total: 6m 32s   remaining: 392ms
999:    learn: 0.9726501        test: 0.9711620 best: 0.9711780 (996)   total: 6m 32s   remaining: 0us
bestTest = 0.9711779952
bestIteration = 996
Shrink model to first 997 iterations.
<catboost.core.CatBoostClassifier at 0x7f33910f87d0>
```

Out[55]:

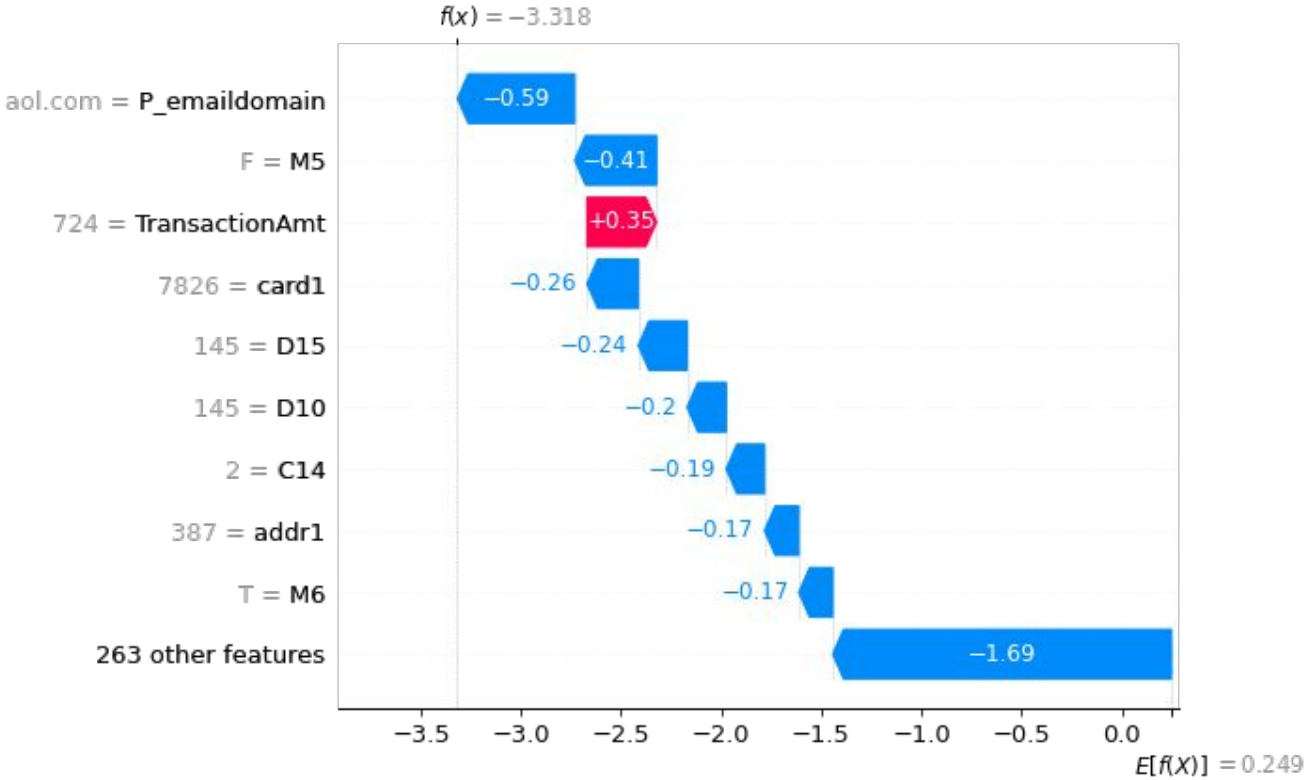
Приложение 4. Первичный анализ данных

Мошеннические операции среди обычных транзакций в разрезе сумм и дат операций



Мошеннических операций - 3,4% от всех операций.

Приложение 5. Feature analysis with shap



Анализ принятия решения моделью по одному экземпляру данных