Allstate Claims Severity

Алексей Носков

Задача

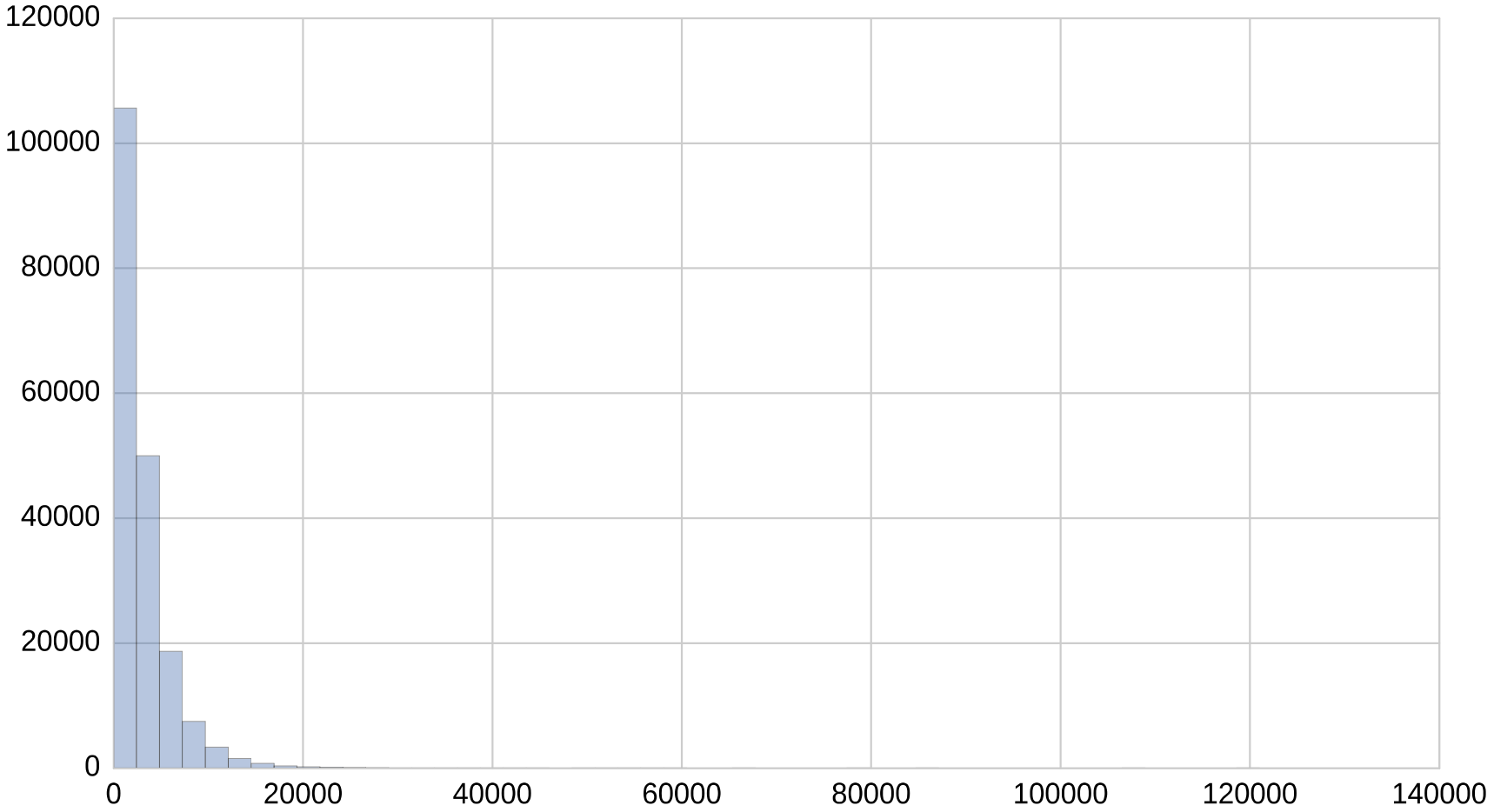
Прогнозирование серьезности страховых случаев

188318 примеров в тренировочной выборке

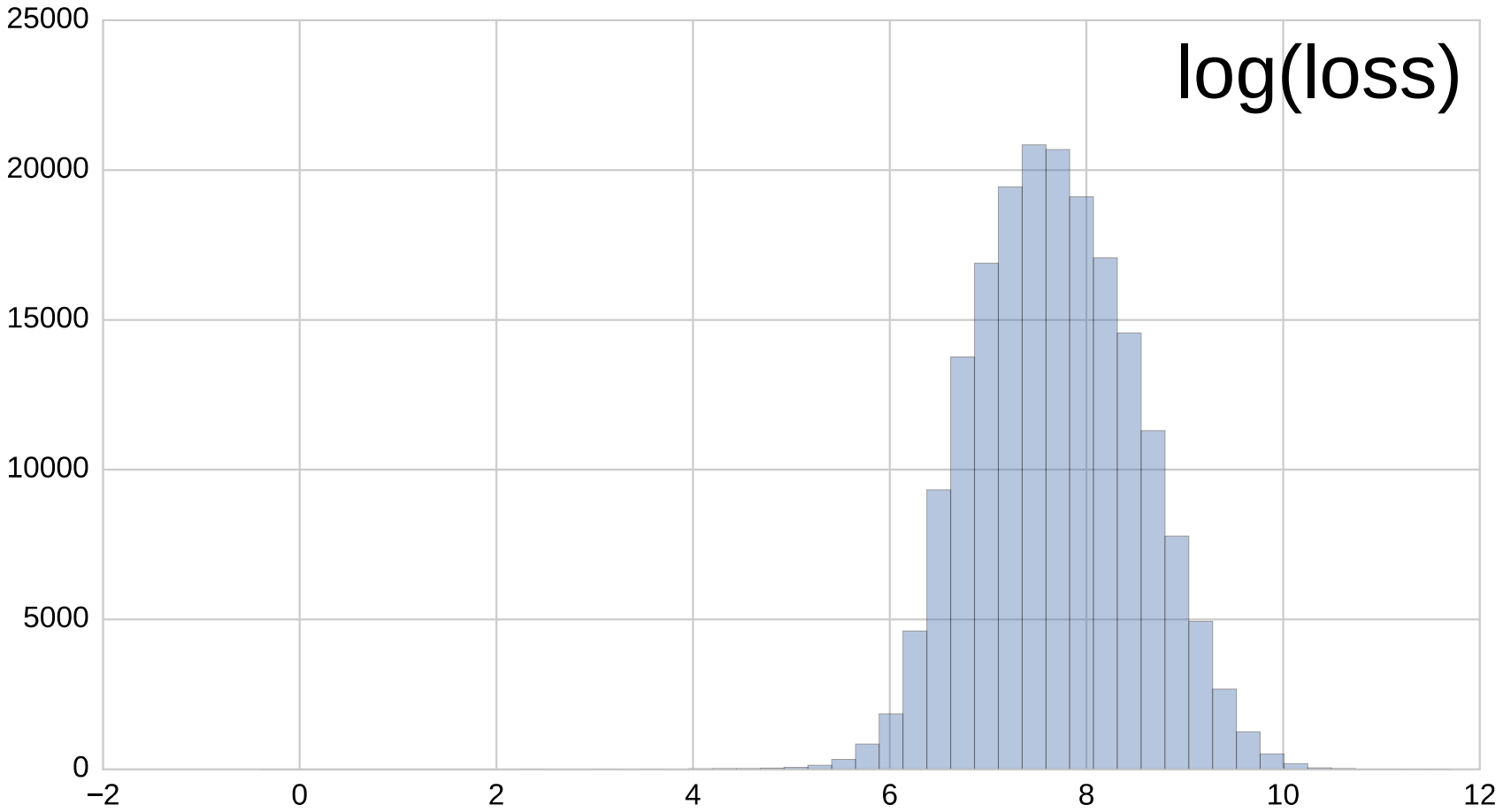
125546 в тестовой

Метрика - MAE

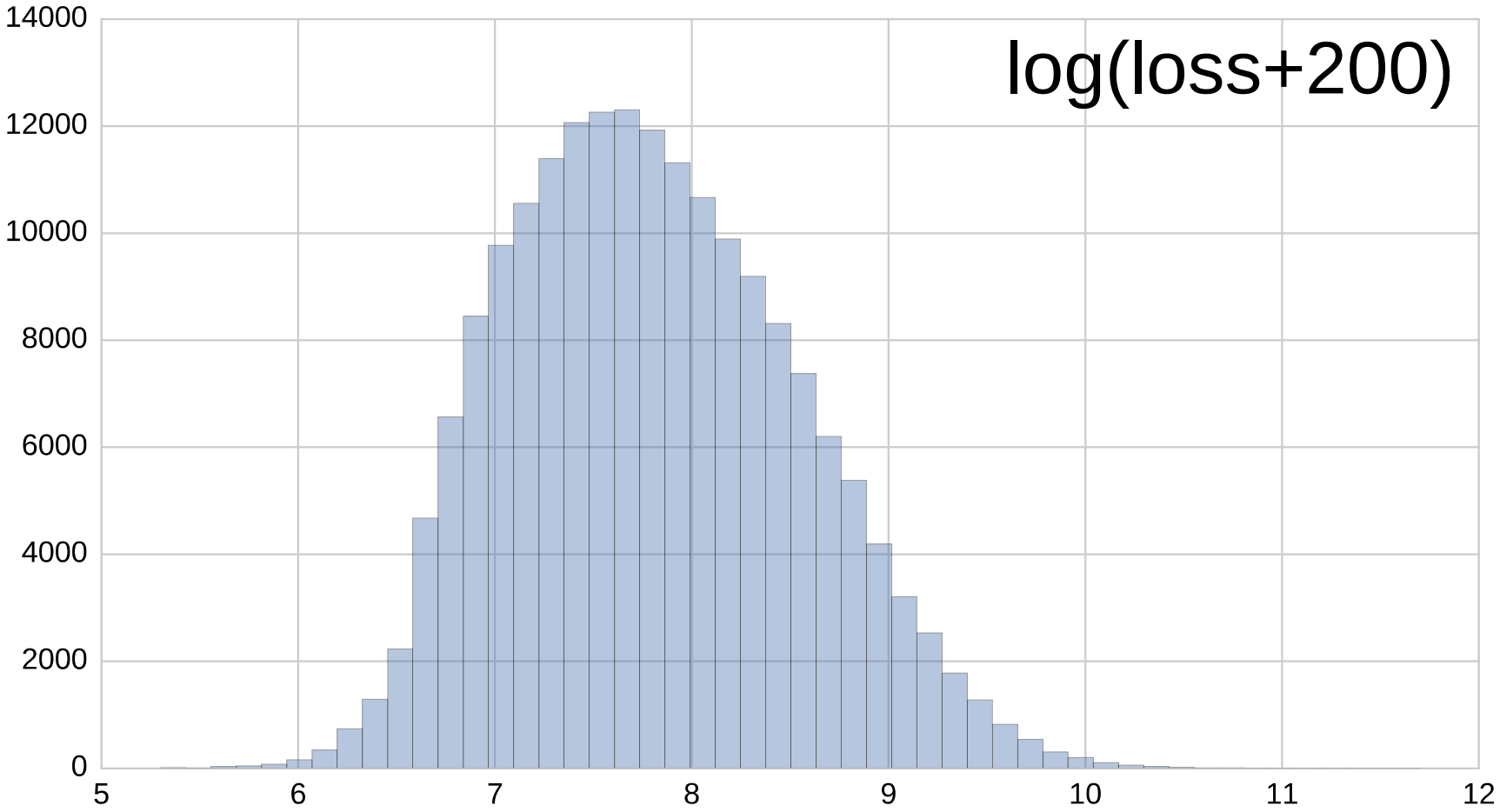
Целевая переменная (loss)



Целевая переменная



Целевая переменная



Признаки

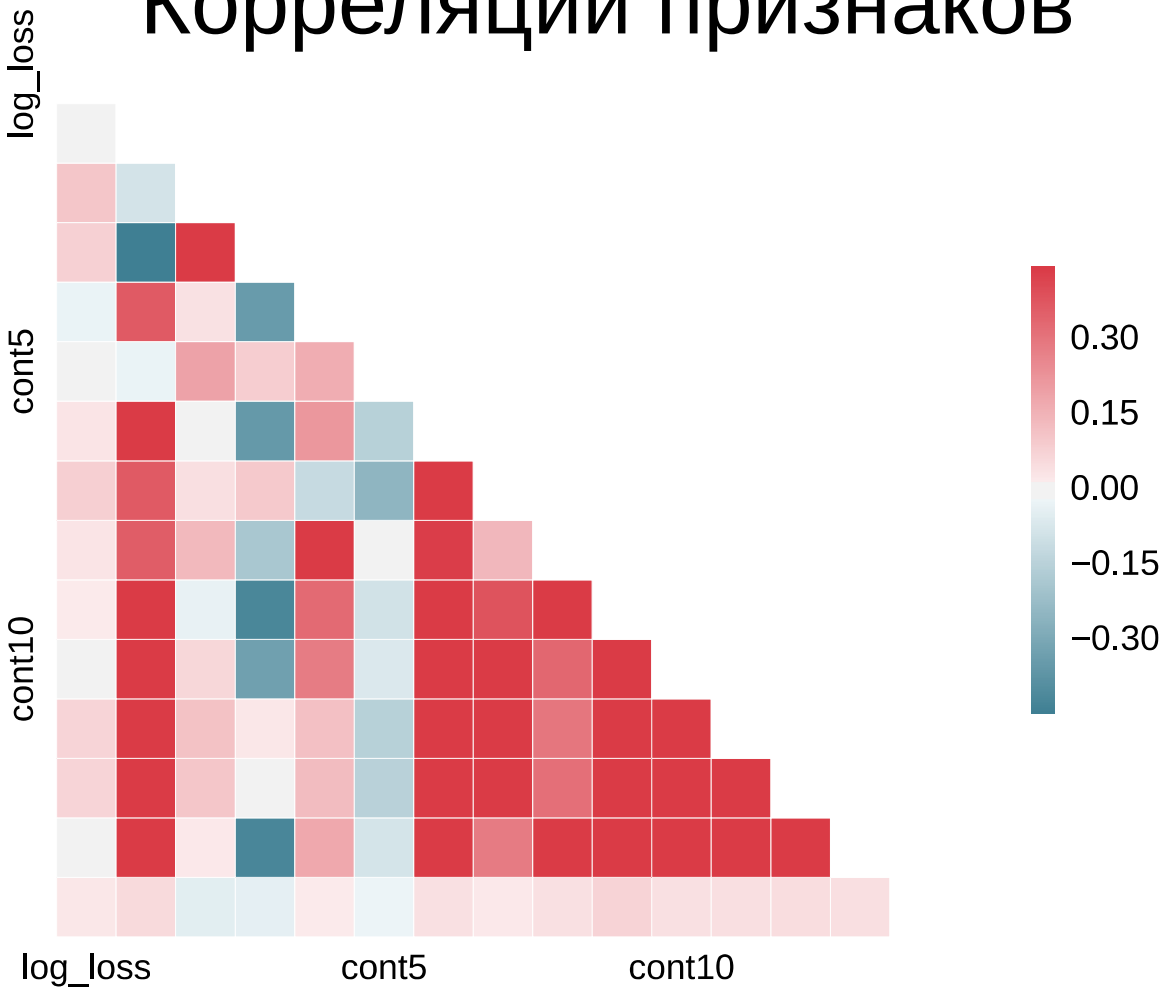
14 числовых признаков

116 категориальных

72 из них — на самом деле бинарные

Все признаки зашифрованы

Корреляции признаков



Категориальные признаки

Кодирование категориальных признаков:

- A — 1, B — 2, ... , Z — 26, AA — 27, ...

Помогает использование комбинаций признаков

- Хорошие комбинации можно найти в дампе XGBoost с помощью Xgbfi

Кросс-валидация

Обычный K-Fold, разбиение на 8 частей

Разница с Public LB порядка 17-18

Модели: линейная регрессия

One-hot encoding для категориальных:

- 1237.43406 CV / 1223.28163 LB

SVD с 500 компонентами (99.8% дисперсии):

- Приблизительно те же метрики
- Но работает значительно быстрее

Модели: линейная регрессия

Фичи из кластеризации:

- Кластеризуем примеры (25 / 50 / 75)
- Посчитаем расстояния до центров кластеров
- Применим exp(-distance)

Кластеризация + SVD:

- 1202.70592 CV / 1189.64998 LB

Модели: RF и ET

- ET
  - 1199.82233 CV / 1176.44433 LB
- RF
  - 1193.61802 CV / 1172.10504 LB

- ET с комбинациями
  - 1194.75138 CV / 1174.94098 LB
- RF с комбинациями
  - 1186.23675 CV / 1166.85340 LB

Проблема:

MAE в RF / ET
слишком медленный

Модели: Gradient Boosting (Sklearn)

Поддерживает LAD

- Работает без трансформации целевой переменной

1151.11060 CV / 1126.30971 LB

Модели: LibFM

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

One-hot-encoding:

- 1196.11333 CV / 1155.68632 LB

После SVD:

- 1177.69251 CV / 1150.37290 LB

Модели: XGBoost

Усреднение нескольких моделей с разными сидами

Базовый вариант (глубина 12)

- 1133.00048 CV / 1112.86570 LB

Лучший (глубина 12)

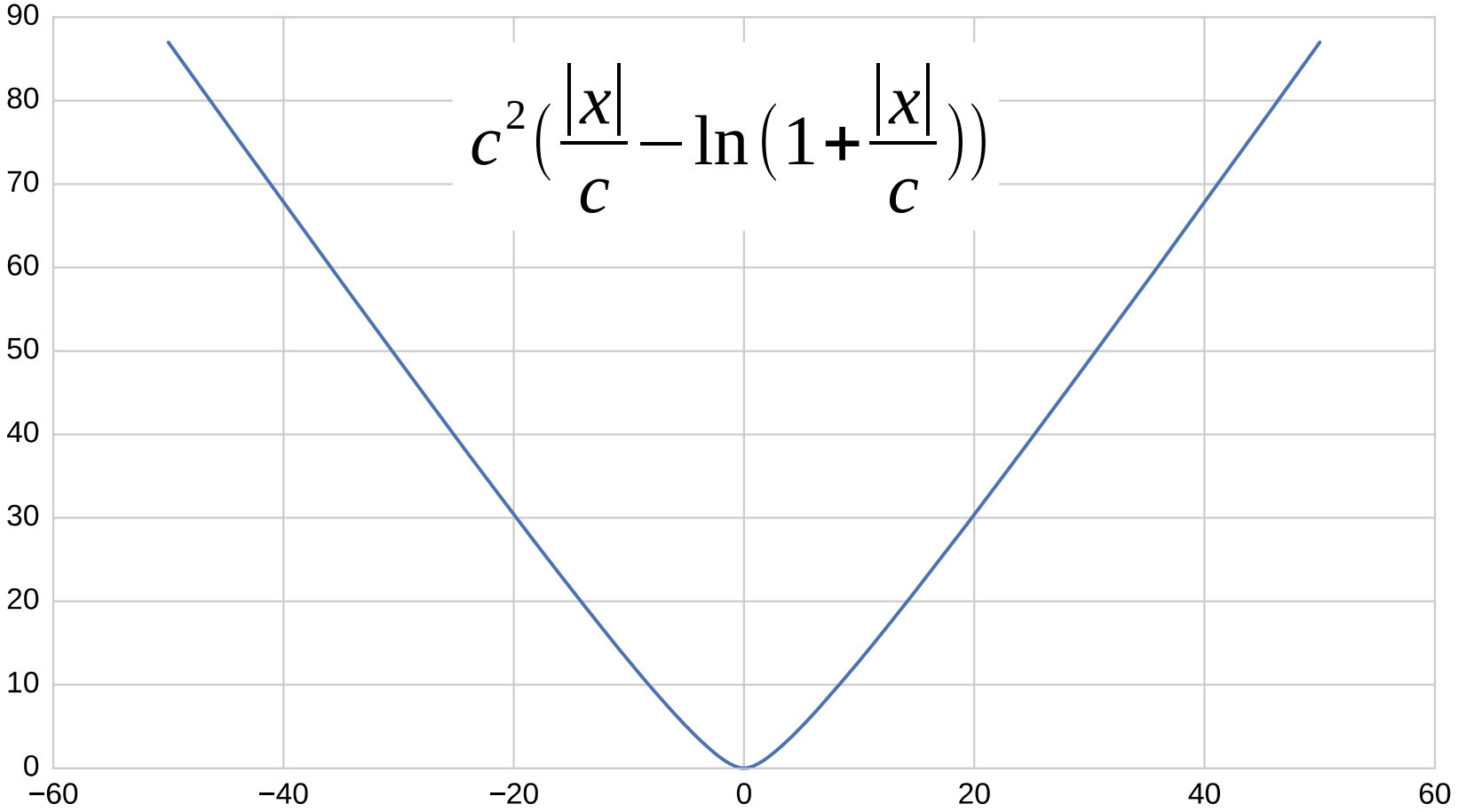
- 1122.64977 CV / 1105.43686 LB

Модели: XGBoost

Что помогло:

- Усреднение
- Комбинации признаков
- Преобразование: loss ^ 0.25
- Другая целевая функция:
- Тюнинг гиперпараметров

«Fair» objective



XGBoost: тюнинг параметров

Важные параметры

- max_depth
- min_child_weight
- colsample_bytree
- subsample
- gamma
- alpha

Хорошее описание:
https://www.kaggle.com/c/santander-customer-satisfaction/forums/t/20662/

Модели: LightGBM

Новая реализация бустинга от Microsoft

- Очень быстрая
- Но значительно меньше возможностей тюнинга

log(200+loss):

- 1129.07923 CV / 1109.30942 LB

loss^0.25:

- 1127.68636 CV / 1107.35775 LB

Модели: NN

3 скрытых слоя по 50-400 нейронов

Усреднение моделей с различными сидами

Базовый вариант:

- 1134.92794 CV / 1116.44915 LB

Лучший вариант:

- 1130.29286 CV / 1110.69527 LB

Модели: NN

Что помогло:

- Усреднение моделей
- Усреднение весов (EMA)
- SVD
- Кластеризация
- Тюнинг параметров
- Batch normalization
- Dropout

SVR и KNN

Очень долго обучаются, но могут добавить разнообразия в ансамбль

Обучаем на небольших подвыборках, усредняем

SVR: 1224.28418 CV / 1204.97577 LB

KNN: 1364.78537 CV / 1341.84046 LB

Автоматический тюнинг параметров

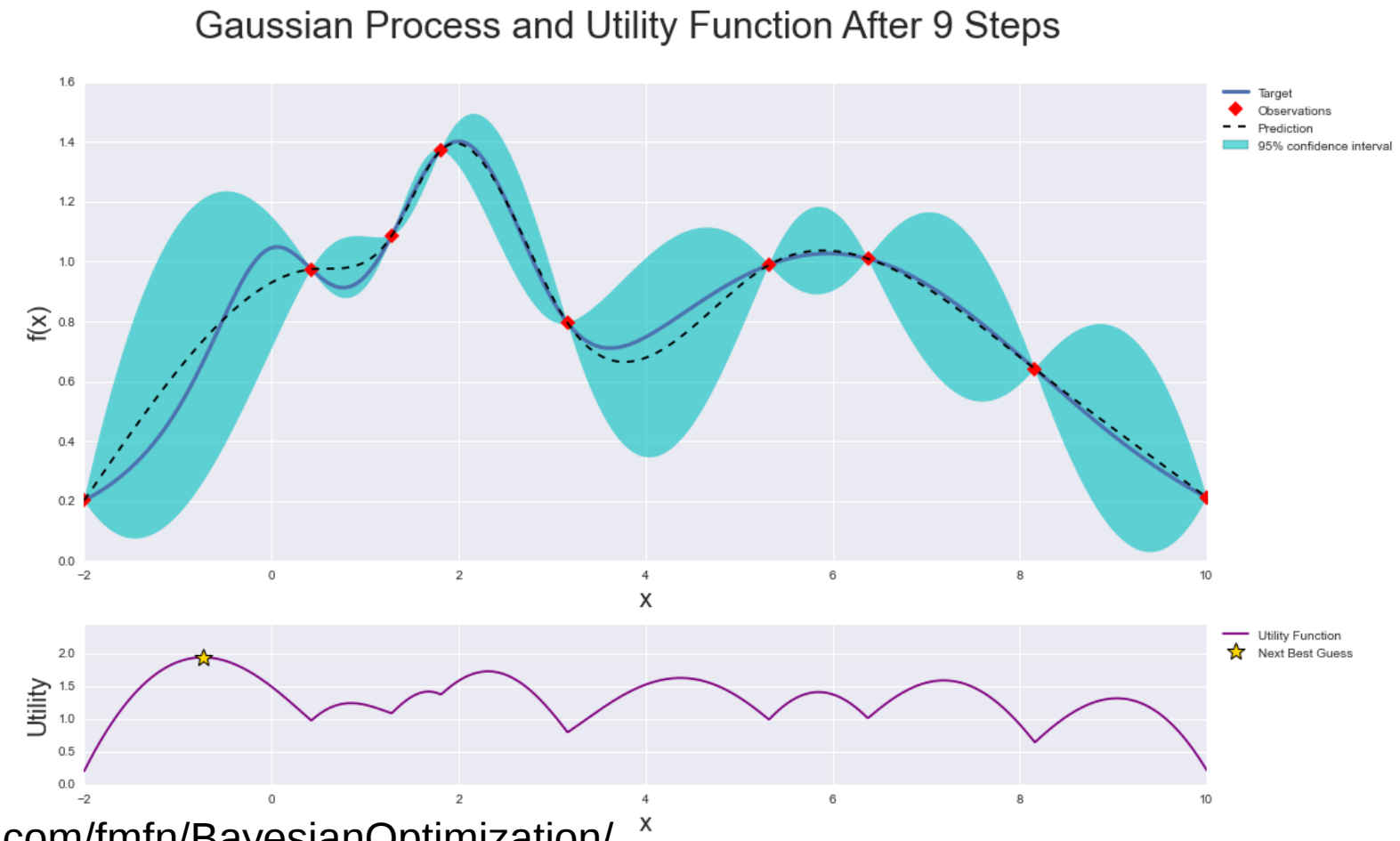
Базовые варианты:

- GridSearch
- RandomSearch

Более продвинутые:

- BayesianOptimization
- HyperOpt

BayesianOptimization



https://github.com/fmfn/BayesianOptimization/

Стекинг: варианты

Linear regression:

- 1118.45564 CV / 1113.08059 LB

XGBoost:

- 1118.16984 CV / 1100.50998 LB

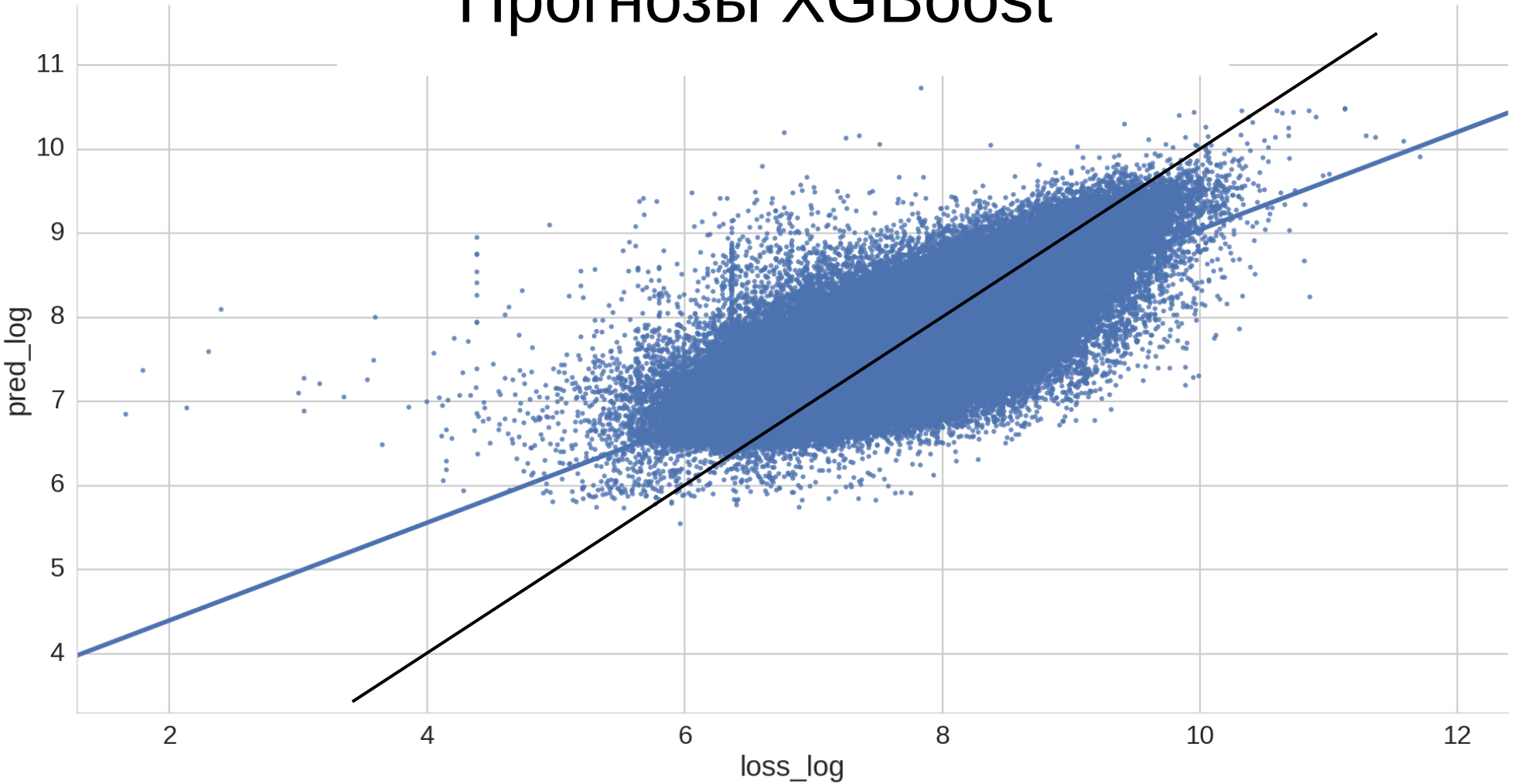
Keras NN:

- 1116.40752 CV / 1098.91721 LB (top-16 в public, top-8 в private)

Sklearn Gradient Boosting:

- 1117.41247 CV / 1099.60251 LB

Прогнозы XGBoost



Постпроцессинг

Попытаемся исправить прогнозы:

$$pred_{corr} = \frac{pred^{p}}{median(pred)^{p-1}}$$

XGBoost: 1117.35084 CV / 1099.63060 LB

Стекинг-2

Нужна максимально простая модель, способная оптимизировать непосредственно MAE

Медианная регрессия

- Реализована в пакете statsmodels
- Нет регуляризации и нестабильные результаты

Стекинг-2: борьба с шумом

Баггинг

- Обучаем на подвыборках и усредняем

Снижение размерности и шума в признаках

- Группируем модели предыдущего уровня и усредняем

В конце усредняем 10 лучших сабмишнов

Итоговый результат:

- 1098.07061 Public, 1110.01364 Private — 2 позиция

Технические аспекты

Используемые ресурсы:

- Ноутбук 2 ядра / 16 Gb памяти
- AWS
- Google Cloud

Время обучения отдельных моделей:

- От часа до двух суток

Другие решения

Преобразования признаков:

- TF-IDF на категориальных признаках
- Count encoding числовых и категориальных признаков
- NN embedding layer
- XGB embedding (индексы листьев как фичи для NN)

Другие решения

Модели:

- Vowpal Wabbit
- Regularized Greedy Forest
- LibFFM

Комбинация моделей верхнего уровня:

- Использование scipy.optimize для выбора весов с регуляризацией

Спасибо за внимание!