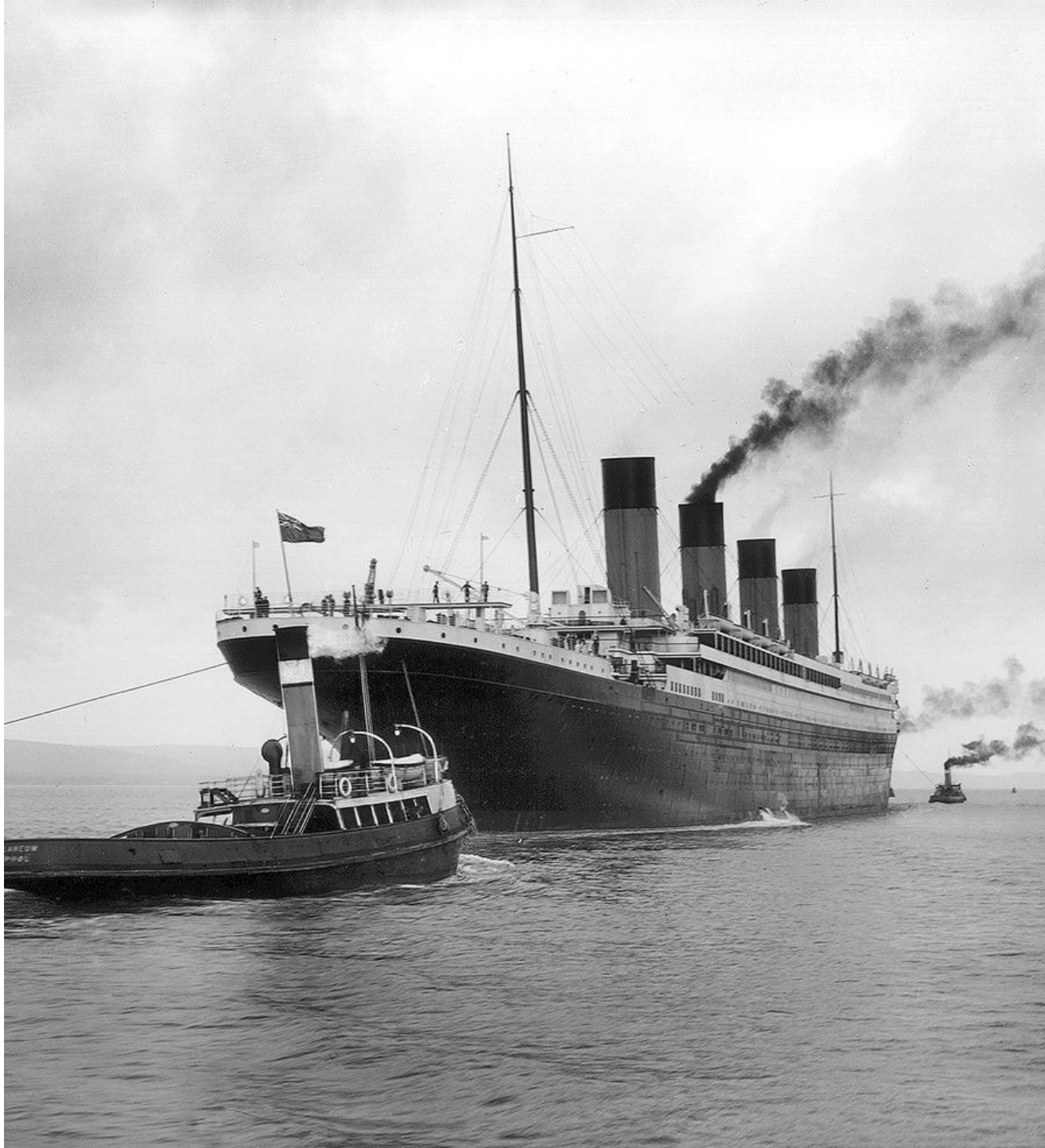
Бинарная классификация на примере CatBoost и датасета «Титаник»

Работа студента группы М8О-309—23
Книги Тимофея



Что вообще такое CatBoost?

CatBoost (или **Cat**egorical **Boost**ing) – это современный алгоритм **градиентного бустинга**, разработанный Яндексом и изначально созданный для **работы с категориальными признаками** — без необходимости их кодировать вручную.

CatBoost строит ансамбль **решающих деревьев**, где каждое следующее дерево исправляет ошибки предыдущих, используя метод градиентного спуска.

К главным особенностям относятся

- эффективная обработка категориальных данных,
- устойчивость к переобучению.

Почему CatBoost для задачи с Титаником?

- Размер датасета
  Titanic — маленький датасет, что может легко привести к **переобучению**. CatBoost достаточно устойчив из-за встроенного порядка обучения.

- Хорошо работает с разнородными фичами
  В датасете есть **несколько** видов признаков: категориальные, числовые, текстовые. Градиентный бустинг по деревьям, как CatBoost, справляется с таким без проблем.

- Часто даёт топовый результат
  Titanic — классический табличный датасет. На таких данных бустинги обычно обгоняют нейросети и линейные модели при равной заботе о признаках.

Подготовка данных | Возраст

Пропущено: 177

Обычно при обработке датасета возраст заполняют медианой или модой по всему датасету. Но это не совсем точно и слишком сильно обобщает признак.

Гораздо более разумной идеей будет выбирать медиану в какой-то **более мелкой группе**.

Наше предложение – брать группы **по полу пассажира и классу** его билета, потому что, например, люди более высокого класса обычно старее, чем пассажиры более низких классов. И также женщины и мужчины стареют по-разному.

Порт посадки

Пропущено: 2

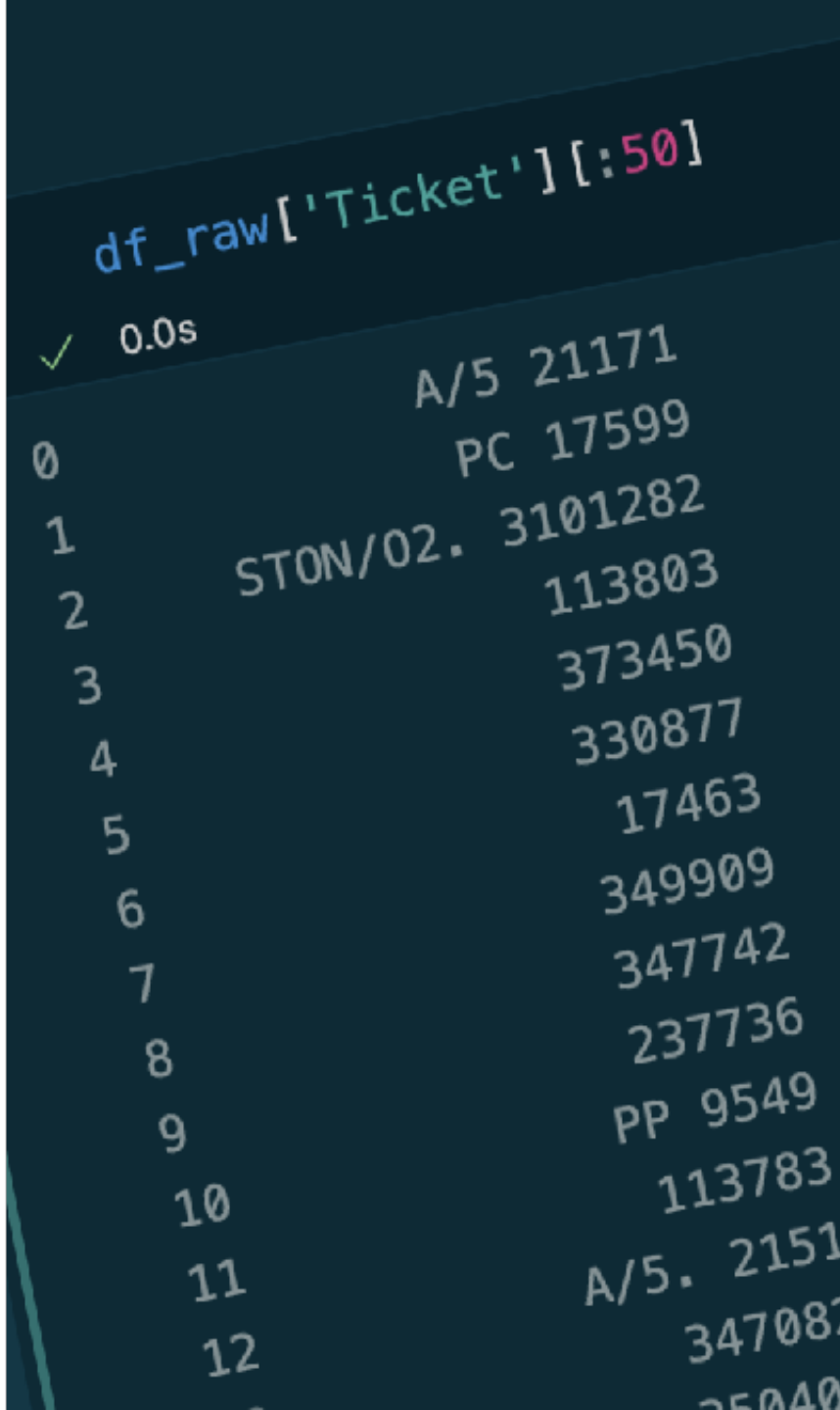
Так как пропущенных значений крайне мало, из достаточно заполнить модой по всему датасету

Подготовка данных | Билет

С виду может показаться, что билет может кодировать какую-то полезную информацию для заполнения других запущенных признаков.

Однако, числа в билете просто являются порядковым номером билета, купленного у агентства. Никакой полезной для нас информации там не закодировано.

Также стоит обратить внимание на аббревиатуры слева от номера. Это **аббревиатура агентства**. Но и эта информация не даёт никаких гарантий. Поэтому билеты тут **никак не помогут.**



Подготовка данных | Кабины

Пропущено: 687

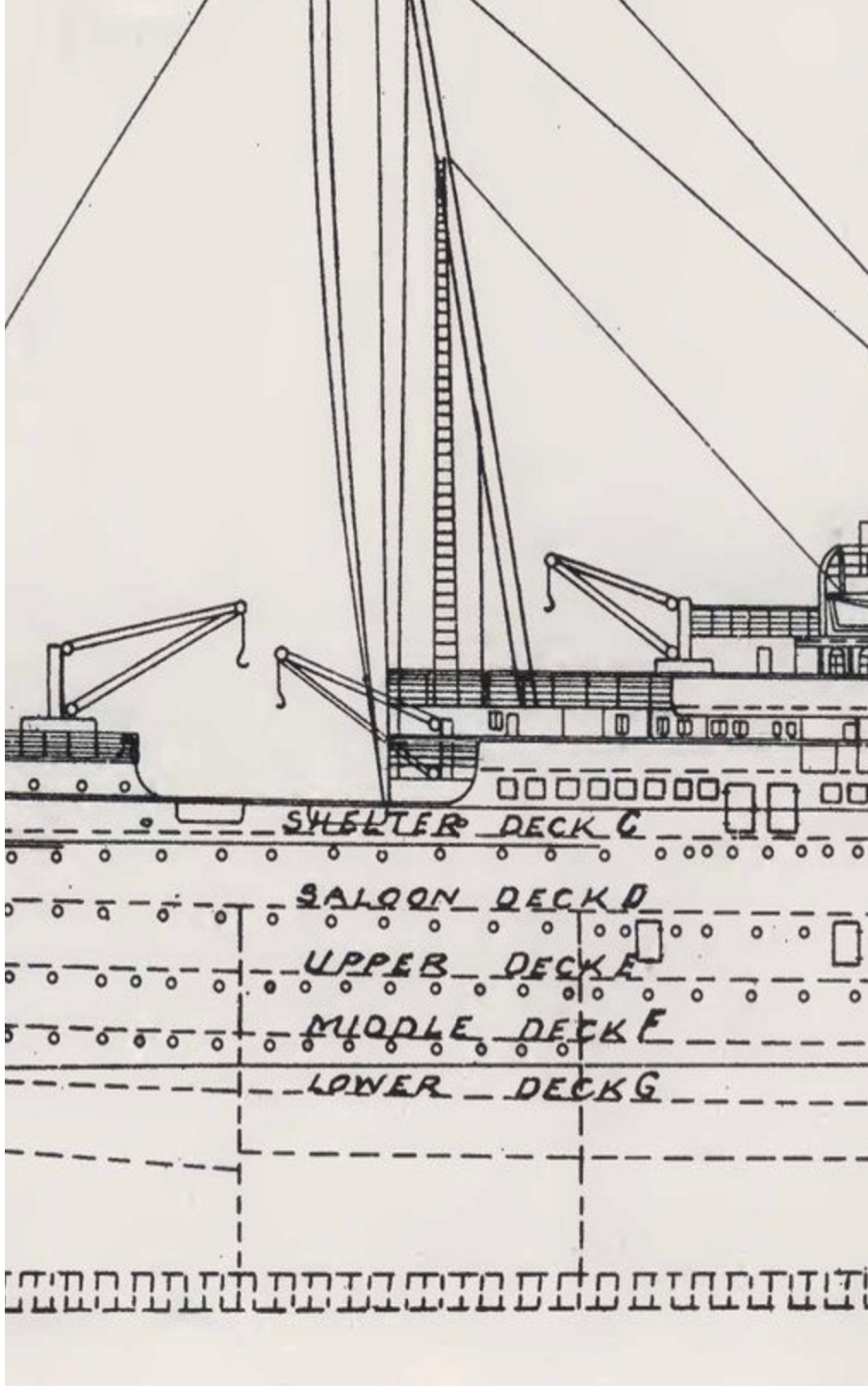
Обратим внимание на то, что в Титанике чем ближе буква кабины к «A», ты ближе пассажир находился к палубе, где и находились спасательные шлюпки.

Также стоит заметить, что исторически верхние кабины занимал первый класс, посередине – второй класс, а нижние отводились третьему классу.

Таким образом, с помощью класса билета можно **предсказать, в какой кабине** находится пассажир.

Для заполнения можно использовать вероятности встречи какого-либо типа кабины среди классов.

Оставим только тип кабины, номер кабины нас не интересует.



Добавление признаков

Полиномиальные признаки вводить не будем: в случае преимущественно категориальных признаков полиномиализация взрывает размерность, создаёт искусственных комбинаций без осмысленной интерпретации и повышает риск переобучения без особого выигрыша в точности.

Введём два смысловых признака:

- FamilySize — размер семейной группы. На шансы спасения влияли семейные связи (могут помогать/мешать выбраться).
- IsAlone = 1— флаг одиночного путешествия. Это даёт модели простой и интерпретируемый сигнал (“один/не один”), который часто оказывается сильнее, чем отдельные (как, например, SibSp и Parch).

CatBoost + Optuna

Optuna - это инструмент, который автоматически подбирает лучшие гиперпараметры модели.

Если раньше приходилось вручную крутить начальные парамеры в надежде на улучшения качества модели, то теперь это можно доверить специальному оптимизатору, который сделает это за нас.

Поэтому помимо «ванильной» модели на основе CatBoost мы также делаем более оптимизированную версию.

+

Примечание: реализация Optuna для CatBoost находится в ноутбуке от лабораторной работы.

Результаты кросс-валидации нескольких видов моделей

Линейные модели

```
=== SVM ===
Accuracy: 0.845 ± 0.060
F1-score: 0.582 ± 0.155
ROC-AUC: 0.795 ± 0.109
```

```
=== KNN ===
Accuracy: 0.832 ± 0.038
F1-score: 0.561 ± 0.142
ROC-AUC: 0.802 ± 0.096
```

Бустинги

```
=== Boosting ===
Accuracy: 0.836 ± 0.016
F1-score: 0.769 ± 0.023
ROC-AUC: 0.883 ± 0.014
```

```
=== XGBoost ===
Accuracy: 0.818 ± 0.020
F1-score: 0.742 ± 0.032
ROC-AUC: 0.885 ± 0.021
```

Деревья

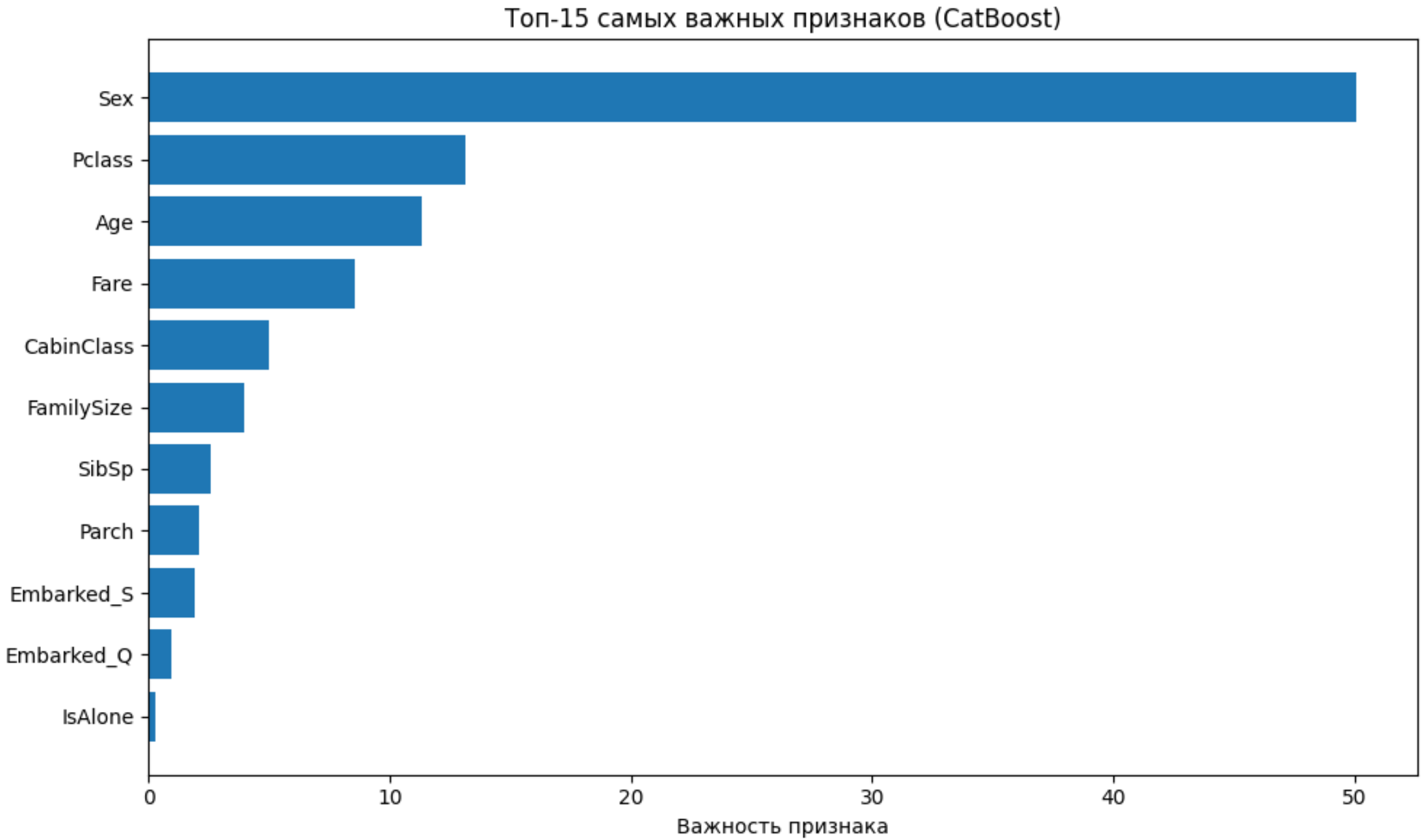
```
=== DesicionTree ===
Accuracy: 0.788 ± 0.032
F1-score: 0.727 ± 0.044
ROC-AUC: 0.779 ± 0.037
```

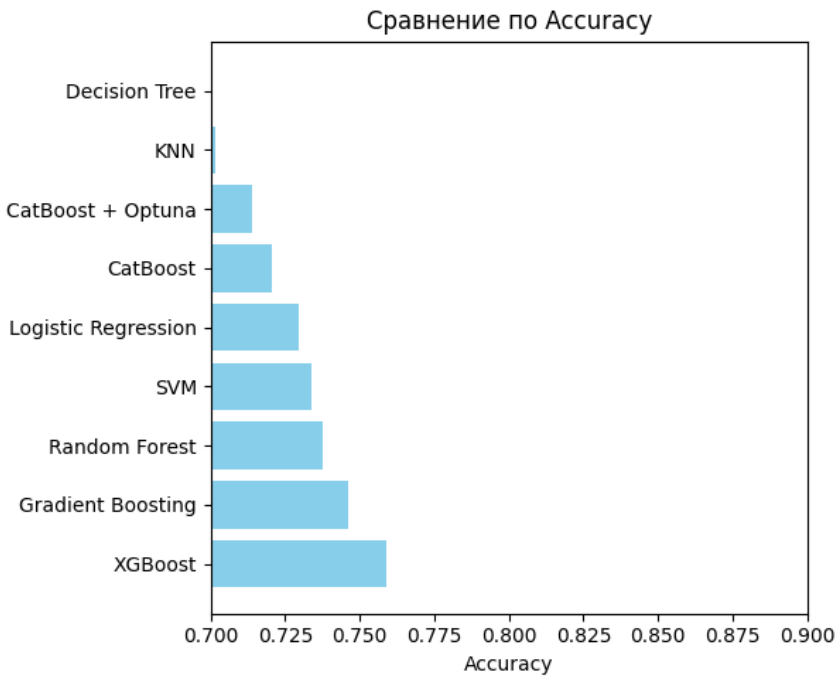
```
=== RandomForest ===
Accuracy: 0.825 ± 0.026
F1-score: 0.766 ± 0.034
ROC-AUC: 0.879 ± 0.021
```

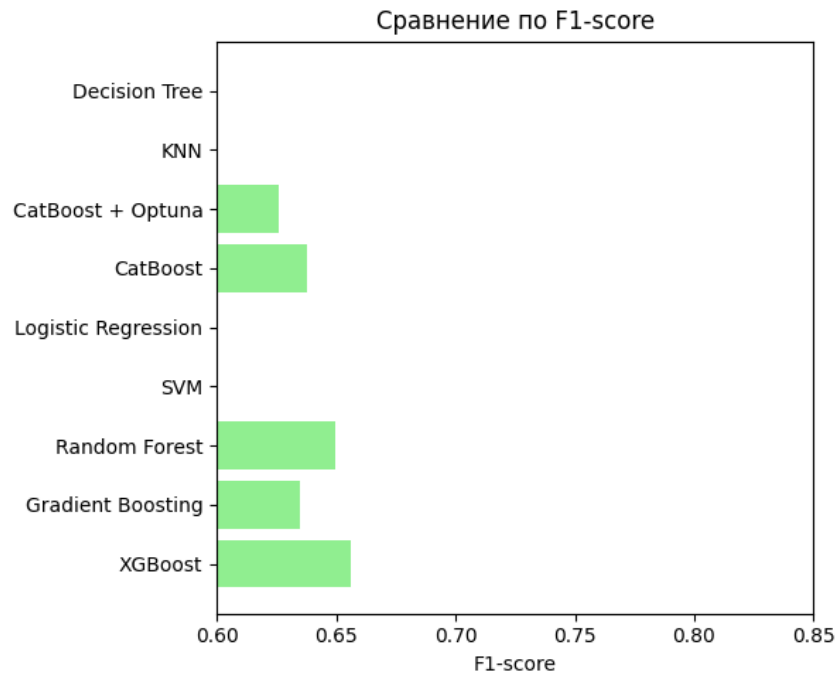
CatBoost

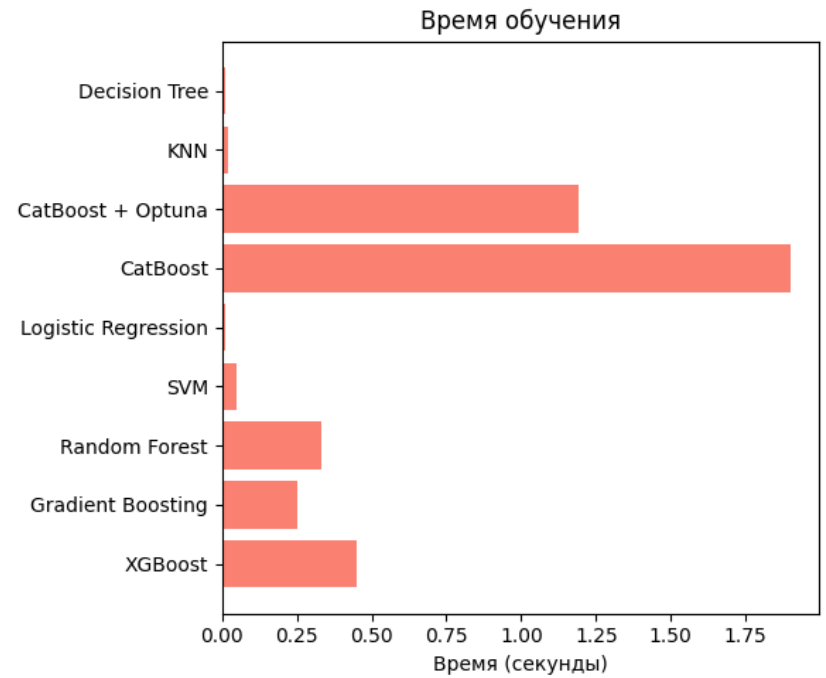
```
=== CatBoost ===
Accuracy: 0.818 ± 0.015
F1-score: 0.757 ± 0.022
ROC-AUC: 0.875 ± 0.020
```

```
=== CatBoost + Optuna ===
Accuracy: 0.828 ± 0.018
F1-score: 0.765 ± 0.026
ROC-AUC: 0.875 ± 0.020
```


Топ-15 самых важных признаков (CatBoost)
Sex
Pclass
Age
Fare
CabinClass
FamilySize
SibSp
Parch
Embarked_S
Embarked_Q
IsAlone
0
10
20
30
40
50
Важность признака


Сравнение по Accuracy
Decision Tree
KNN
CatBoost + Optuna
CatBoost
Logistic Regression
SVM
Random Forest
Gradient Boosting
XGBoost
0.700
0.725
0.750
0.775
0.800
0.825
0.850
0.875
0.900
Accuracy


Сравнение по F1-score
Decision Tree
KNN
CatBoost + Optuna
CatBoost
Logistic Regression
SVM
Random Forest
Gradient Boosting
XGBoost
0.60
0.65
0.70
0.75
0.80
0.85
F1-score


Время обучения
Decision Tree
KNN
CatBoost + Optuna
CatBoost
Logistic Regression
SVM
Random Forest
Gradient Boosting
XGBoost
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Время (секунды)

Выводы

1. Самыми важными признаками в датасете оказались: пол пассажира, его возраст, класс билета и его стоимость.

2. CatBoost, хоть и не самая точная модель (уступает XGBoost), показала себя достойно и на уровне других бустингов, тем самым сильно опережая обычные деревья и линейные модели. Эта модель годится для продуктовых задач, требующих высокой точности.

3. Optuna действительно помогла в обучении CatBoost, ускорив время обучения модели и повысив показатели, хоть и для первоначального подбора гиперпараметров нужно побольше времени, чем тренировка обычной модели.