**Аналитический отчёт**

Романенков М.Г.

Содержание

[1. Анализ представленных данных (EDA) 2](#_Toc206352628)

[2. Задачи регрессии 8](#_Toc206352629)

[2.1. Задача регрессии IC50 9](#_Toc206352630)

[2.2. Задача регрессии CC50 10](#_Toc206352631)

[2.3. Задачи регрессии SI 11](#_Toc206352632)

[3. Задачи бинарной классификации 12](#_Toc206352633)

[3.1. Предсказание: IC50 > медианы 13](#_Toc206352634)

[3.2. Предсказание: CC50 > медианы 14](#_Toc206352635)

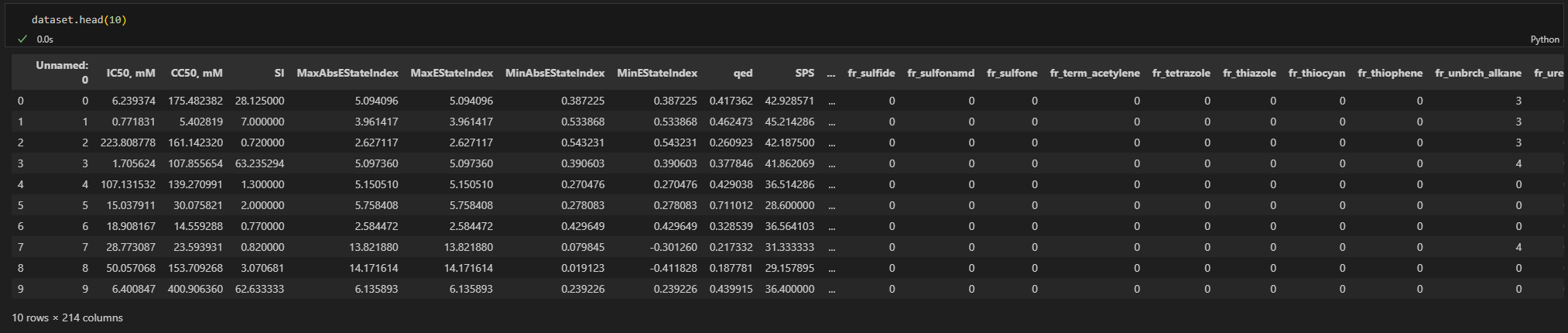
[3.3. Предсказание: SI > медианы 15](#_Toc206352636)

[3.4. Предсказание: SI > 8 16](#_Toc206352637)

[4. Вывод 17](#_Toc206352638)

# Анализ представленных данных (EDA)

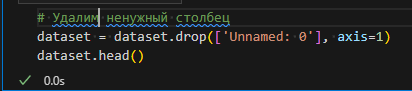
Выведем несколько верхних строк датасета, для ознакомления.



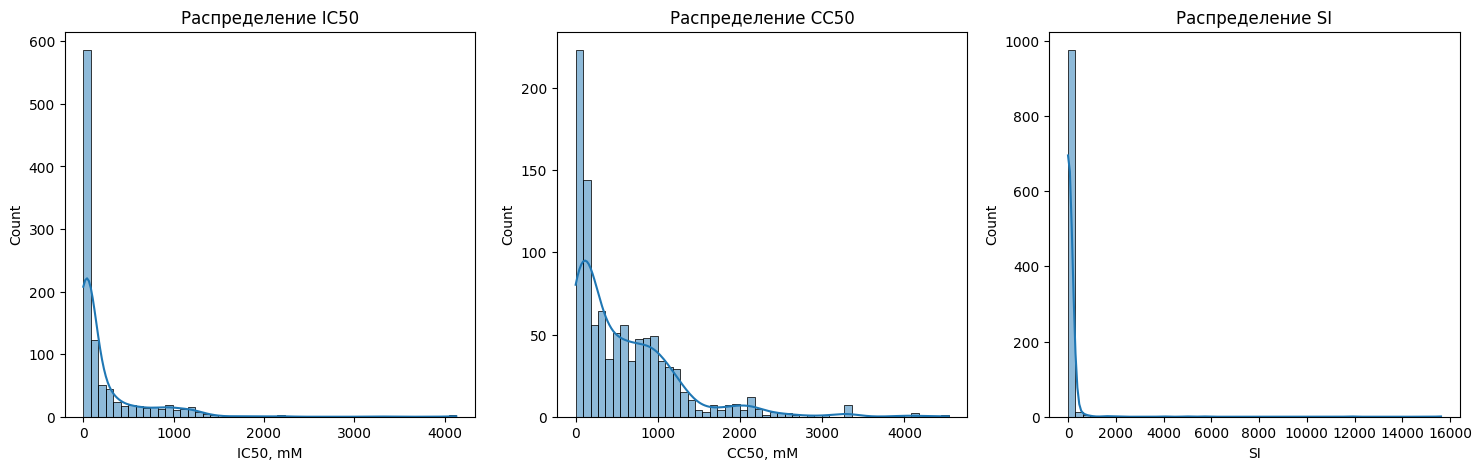
Обратим внимание на ключевые параметры

* IC50 – концентрация для 50% ингибирования
* CC50 – концентрация для 50% цитотоксичности
* SI – индекс селективности (CC50 / IC50)

Также заметим, что присутствует не информативный столбец 'Unnamed: 0' и удалим его.



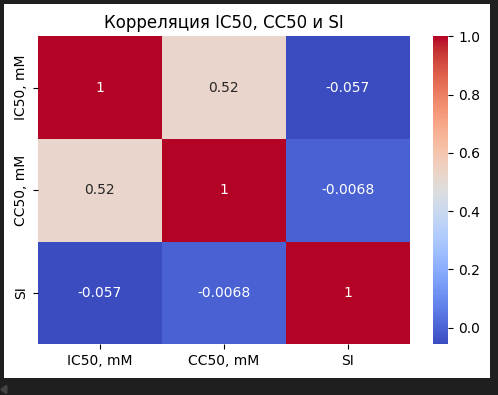
Визуализируем распределения ключевых параметров



Видим, что все распределения имеют сильный сдвиг вправо.

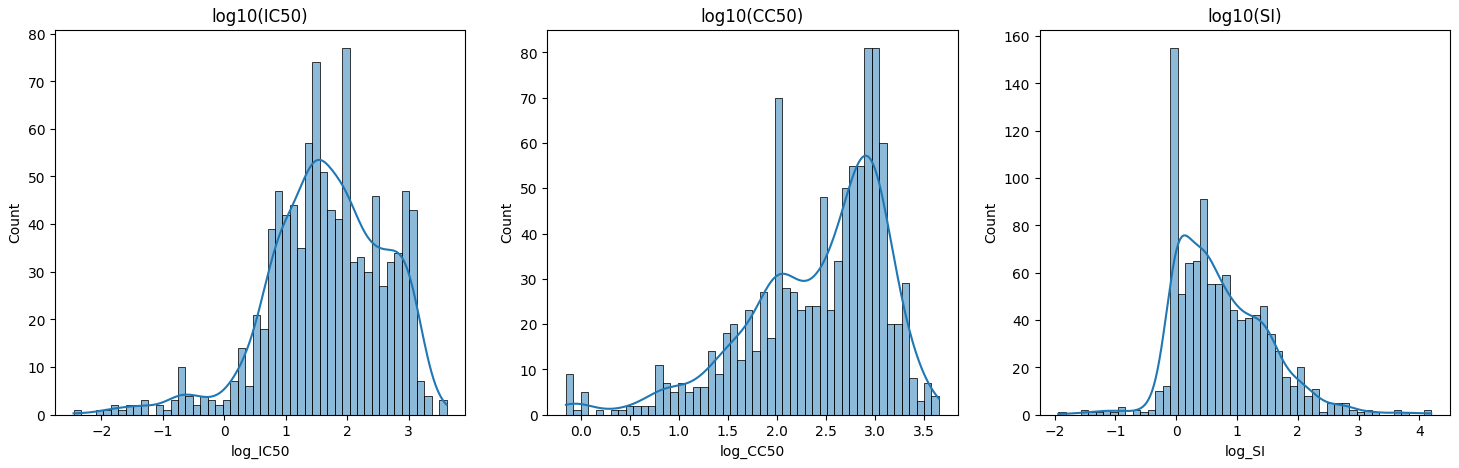
Также, в распределениях IC50 и SI присутствуют аномально большие значения.

Затем обратим внимание на корреляцию между ключевыми признаками



Видно, что между IC50 и CC50 корреляция низкая. Следовательно это независимые характеристики.

Чтобы уменьшить влияние выбросов и лучше увидеть структуру данных, применим логарифмирование.

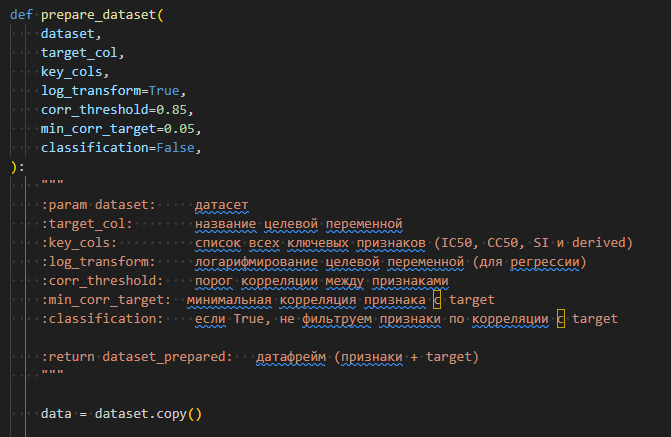


После логарифмирования распределения стали более похожи на нормальные. Это позволит нам использовать линейные модели и стандартные статистические тесты.

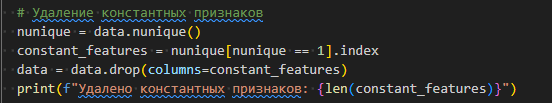
P.S. В дальнейшем логарифмированными столбцами я планирую пользоваться для задач регрессии.

Далее я реализовал метод ‘def prepare\_dataset()’ для подготовки отдельных датасетов для задач регрессии и классификации.

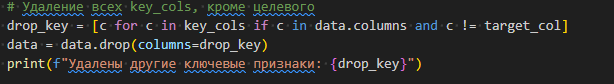
Сигнатура функции



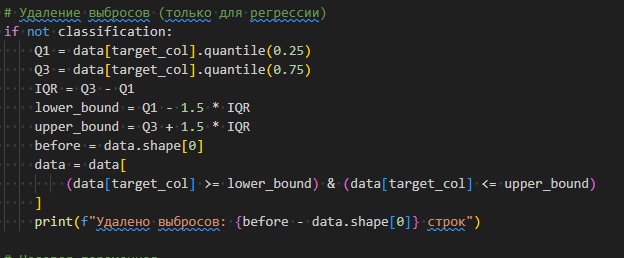
Тело функции



Удаляем все столбцы в которых значения неизменны



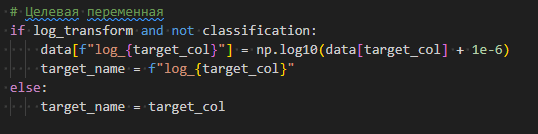
Оставляем только целевой признак, необходимый для конкретной задачи. Это сделано чтобы не возникало ситуаций как эта – (SI = CC50 / IC50, значит SI напрямую содержит информацию о целевой переменной.)



На этом этапе выбросы удаляются с помощью IQR(межквартильный размах**).**

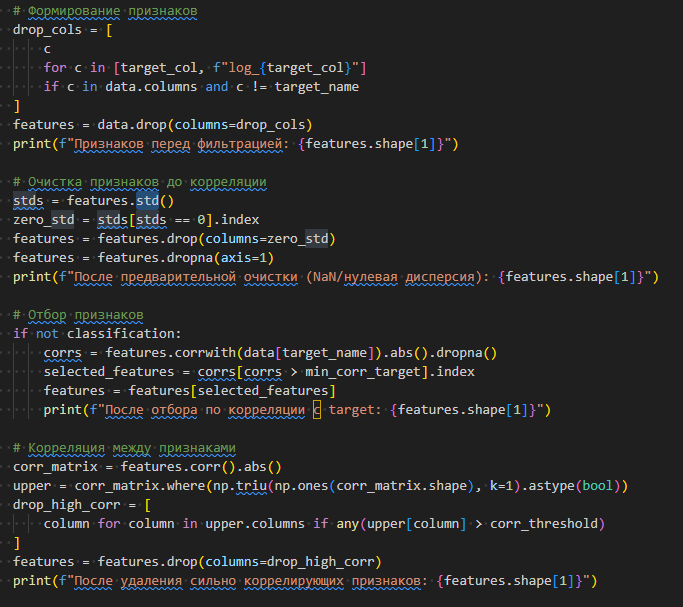
Для классификации выбросы не удаляются, т.к. даже если у соединения экстремальное значение ключевого параметра, это всё равно один из классов (0 или 1).

Если мы удалим такие строки, мы искусственно исказим баланс классов.

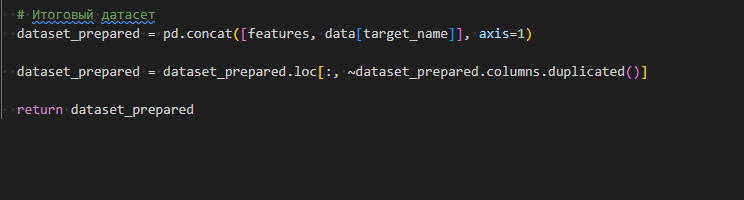


Логарифмируем целевую переменную только для задач регрессии (причины описаны ранее).

Теперь уберём признаки, которые слабо коррелируют с ключевым признаком (< min\_corr\_target) и признаки, которые имеют высокую корреляцию между собой (> corr\_threshold).

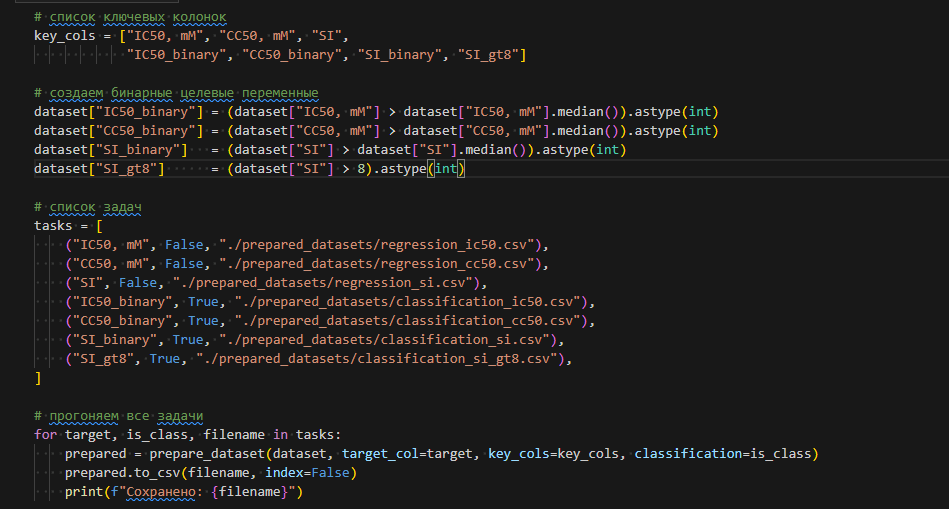


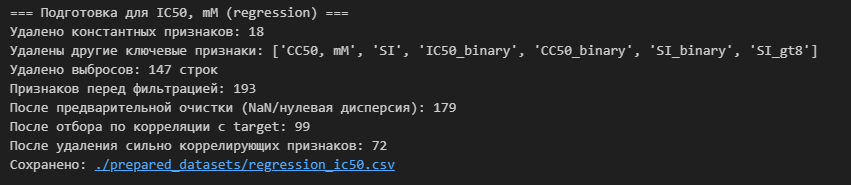
В завершении вернём датасет, готовый для выполнения дальнейших задач.



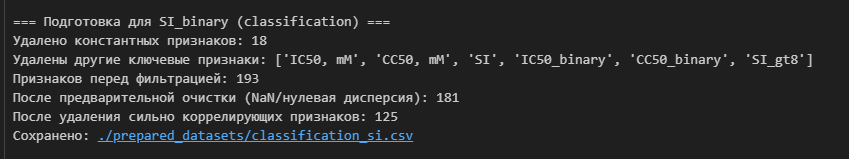
С помощью ранее описаного метода подготовим датасеты для дальнейшей работы.

Все они будут храниться в папке ‘prepared\_datasets’ в формате ‘.csv’.



Пример подготовки датасета для задачи регресии

Пример подготовки датасета для задачи классификации



# Задачи регрессии

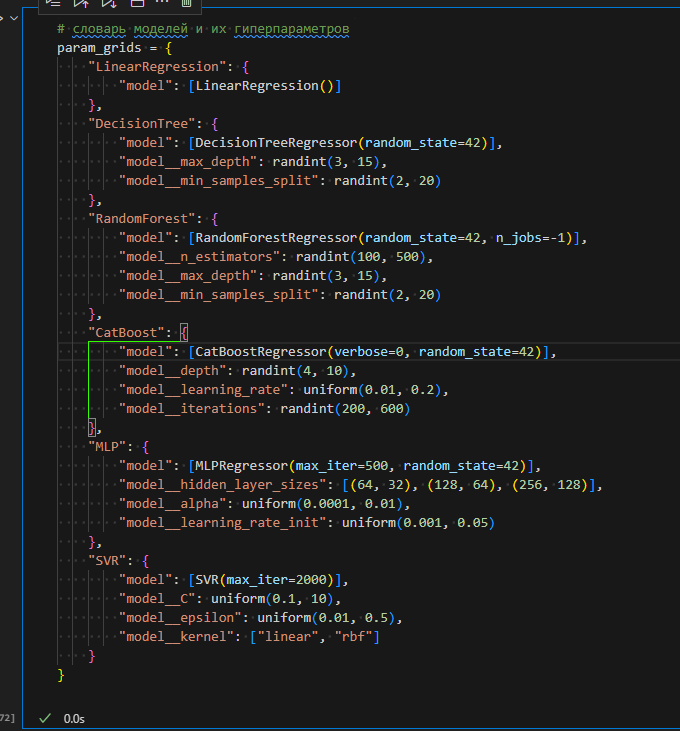
Для решения задач регресии будут использоваться такие модели:

* **Линейная регрессия** (LinearRegressio**n)**
* **Дерево решений** (DecisionTreeRegressor)
* **Случайный лес** (RandomForestRegressor)
* **Градиентный бустинг деревьев решений (**CatBoostRegressor)
* **Многослойный перцептрон (**MLPRegressor**)**
* **Метод опорных векторов (**Support Vector Regressor**)**

Сравнивать данные модели я буду по следующим метрикам:

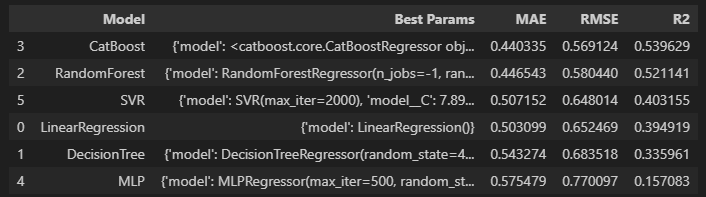
* **MAE** (Mean Absolute Error) - показывает, насколько в среднем модель ошибается в тех же единицах, что и целевая переменная.
* **RMSE** (Root Mean Squared Error) - более чувствителен к крупным ошибкам, т.к. ошибки возводятся в квадрат.
* **R2** (коэффициент детерминации) - насколько хорошо модель объясняет вариацию данных.

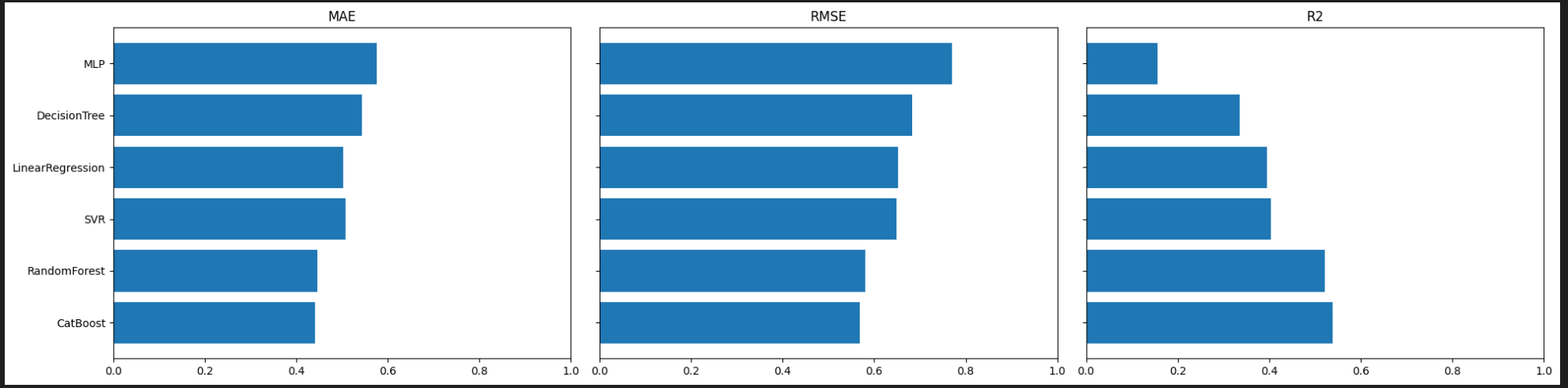
Для подбора лучших гиперпараметров будет использоваться следующий словарь



## Задача регрессии IC50

Метрики для каждой модели с наиболее удачными наборами гиперпараметров.

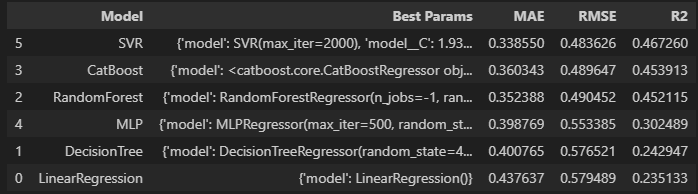


Графики сравнения моделей по метрикам  


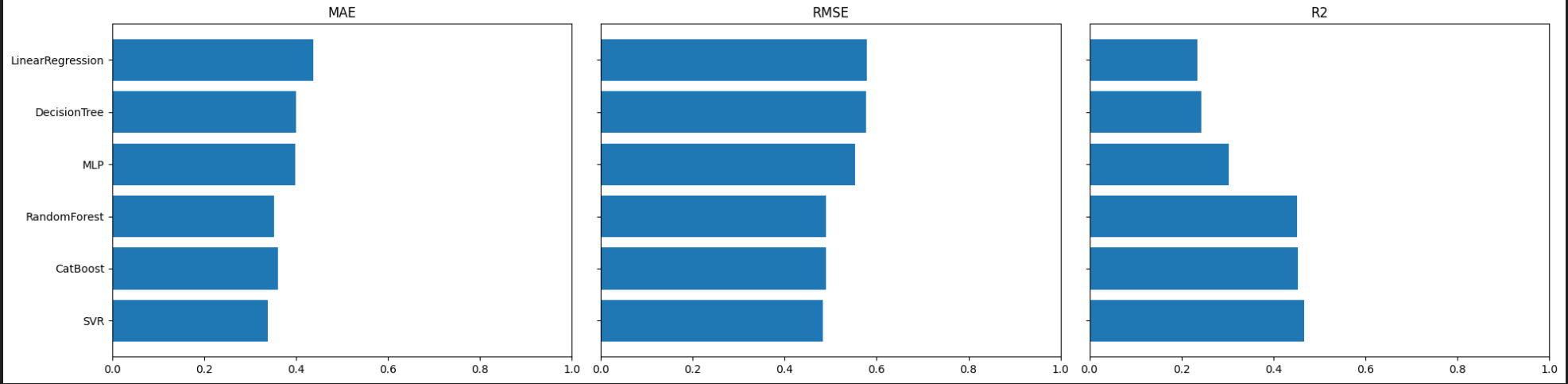
Графики предсказанных против истинных значений 3х лучших моделей  

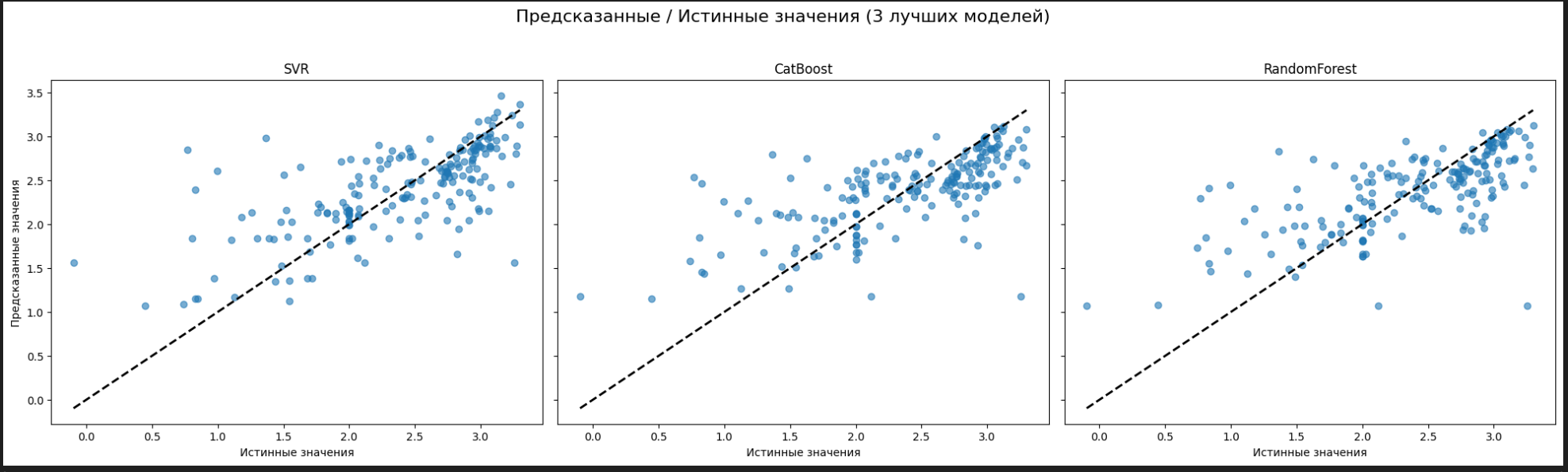

## Задача регрессии CC50

Метрики для каждой модели с наиболее удачными наборами гиперпараметров.

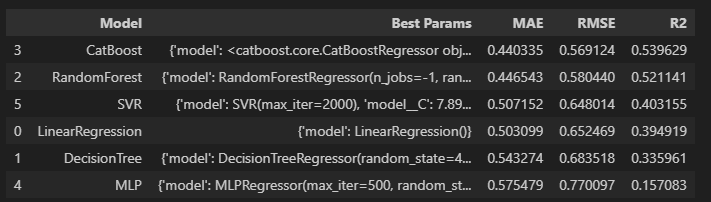


Графики сравнения моделей по метрикам

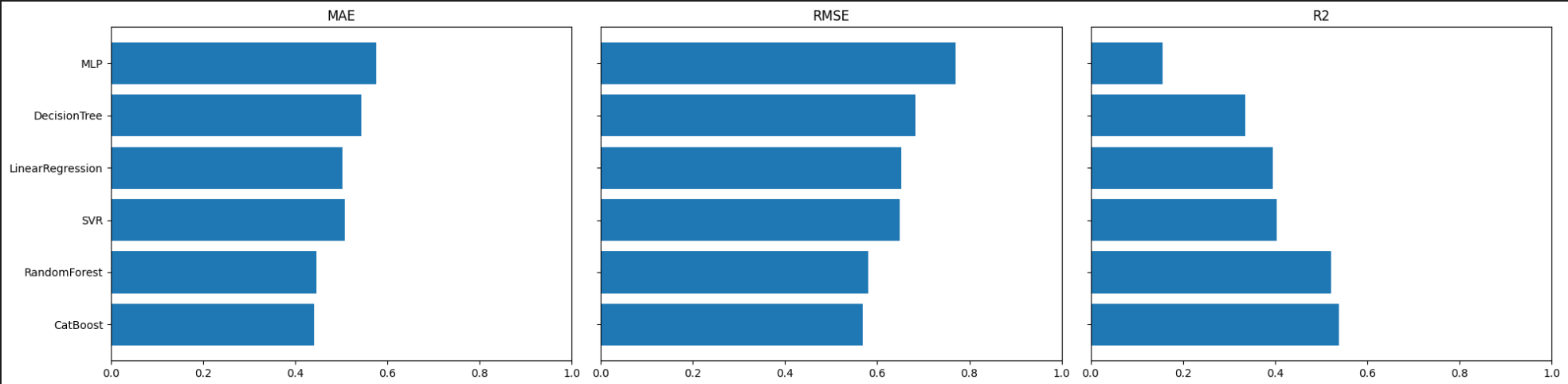


Графики предсказанных против истинных значений 3х лучших моделей  


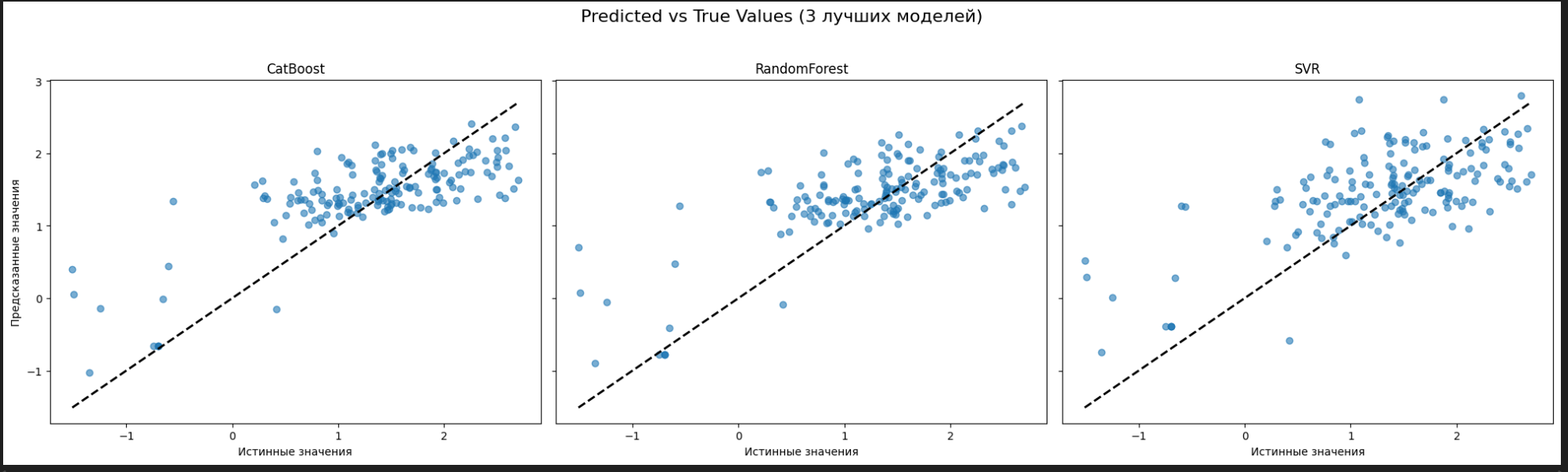
## Задачи регрессии SI

Метрики для каждой модели с наиболее удачными наборами гиперпараметров. 

Графики сравнения моделей по метрикам



Графики предсказанных против истинных значений 3х лучших моделей

****

# Задачи бинарной классификации

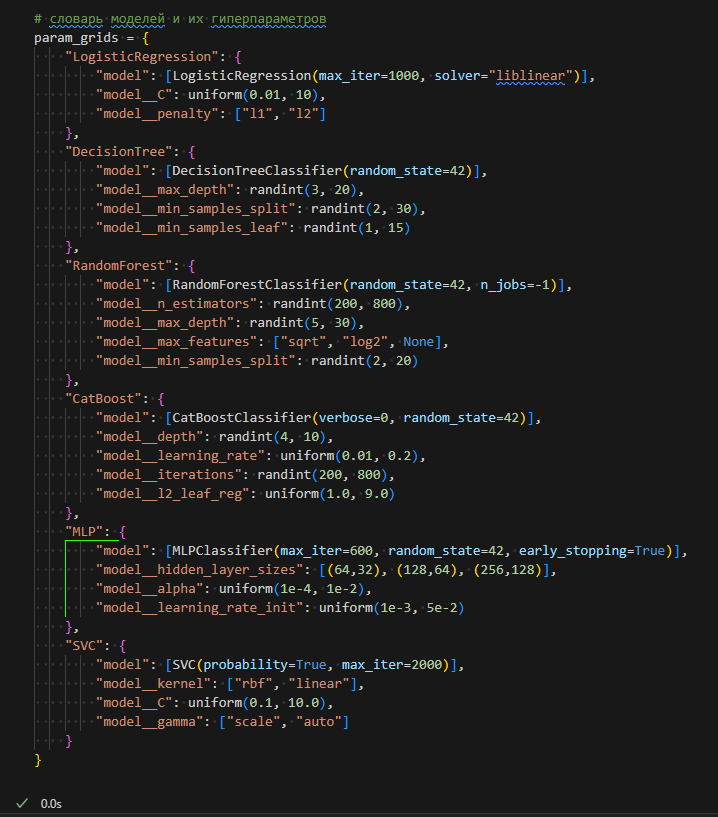
Для решения задач бинарной классификации будут использоваться такие модели:

* **Логистическая регрессия** (Logistic Regression)
* **Дерево решений** (Decision Tree Classifier)
* **Случайный лес** (Random Forest Classifier)
* **Градиентный бустинг деревьев решений (**CatBoost Classifier)
* **Многослойный перцептрон (**MLP Classifier**)**
* **Метод опорных векторов** (SVC / Support Vector Classifier)

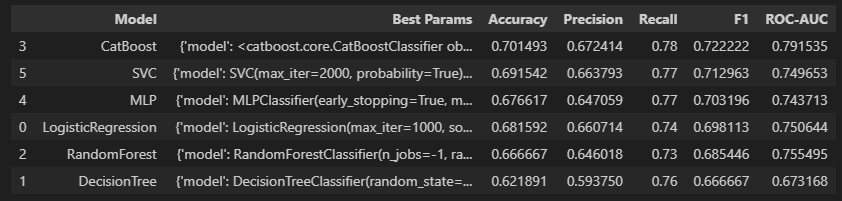
Сравнивать данные модели я буду по следующим метрикам:

* **Accuracy** - простая метрика, % правильных ответов. Но может быть обманчива, если классы не сбалансированы.
* **Precision / Recall / F1** - стандарт для несбалансированных данных.
* **ROC-AUC** - полезно для бинарной классификации, показывает, насколько модель хорошо ранжирует вероятности.

Для подбора лучших гиперпараметров будет использоваться следующий словарь



## Предсказание: IC50 > медианы

Метрики для каждой модели с наиболее удачными наборами гиперпараметров. 

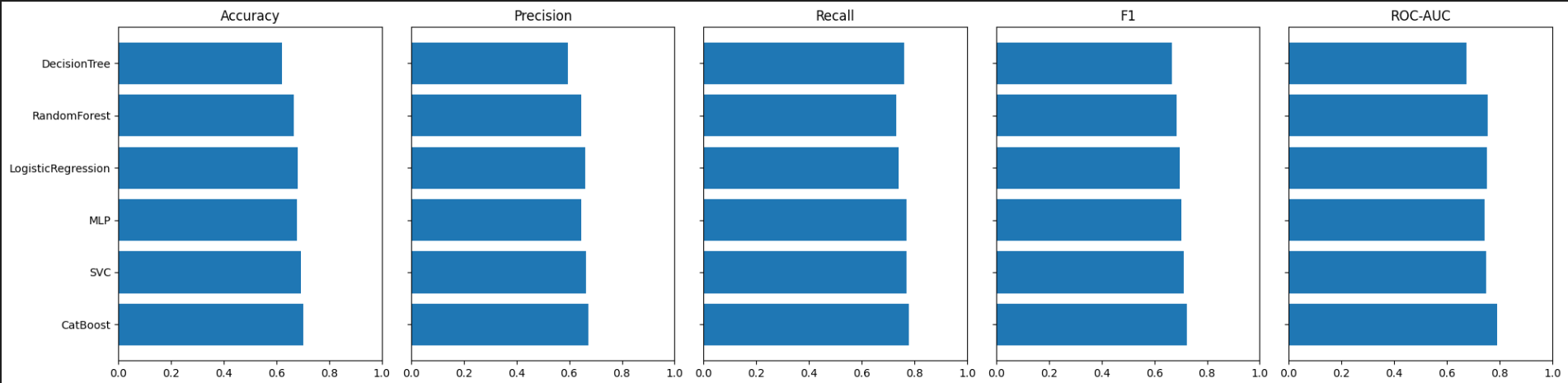
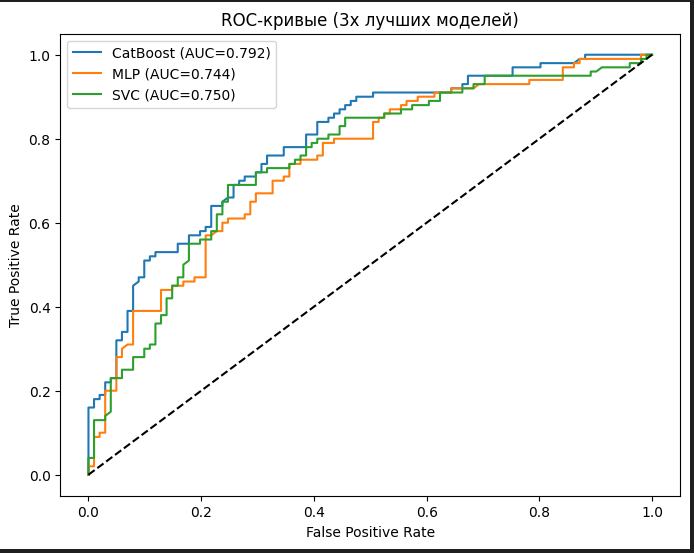
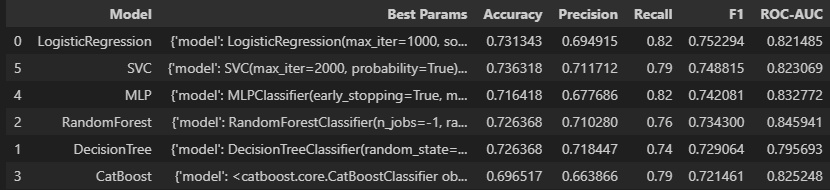
Графики сравнения моделей по метрикам  


График ROC кривых



## Предсказание: CC50 > медианы

Метрики для каждой модели с наиболее удачными наборами гиперпараметров. 

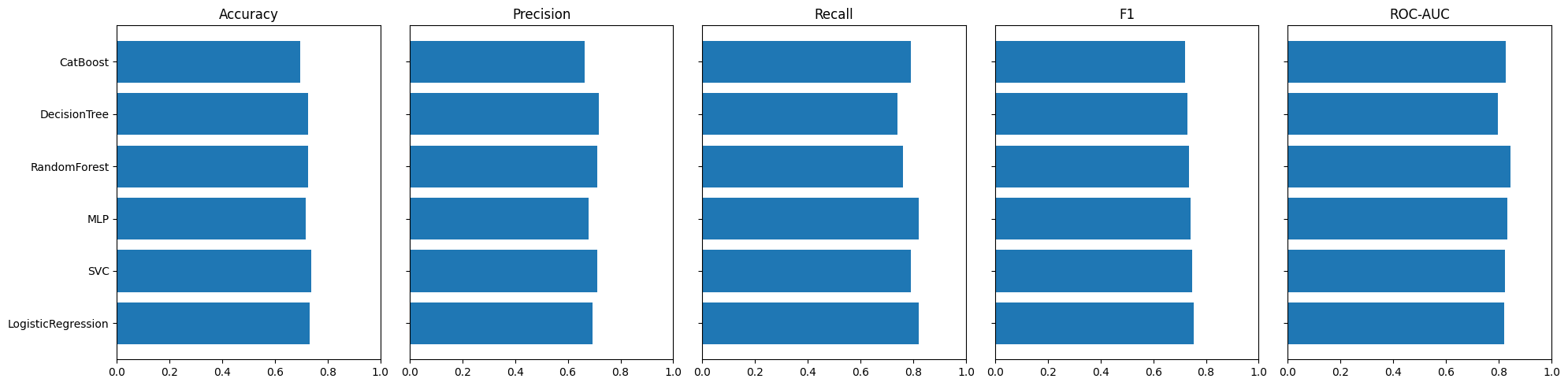
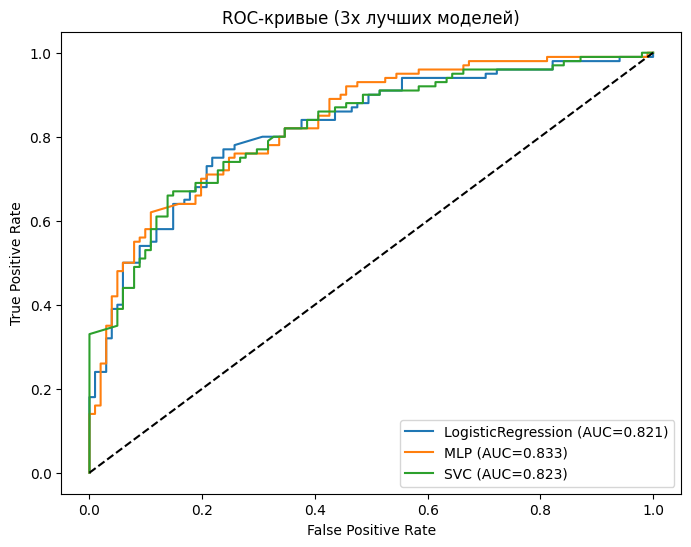
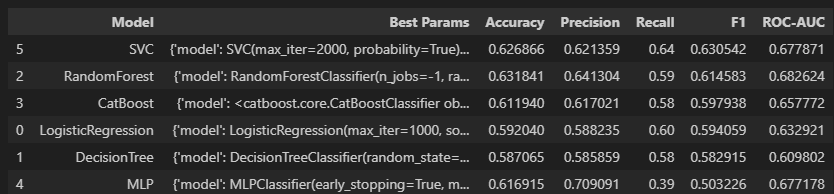
Графики сравнения моделей по метрикам  


График ROC кривых



## Предсказание: SI > медианы

Метрики для каждой модели с наиболее удачными наборами гиперпараметров. 

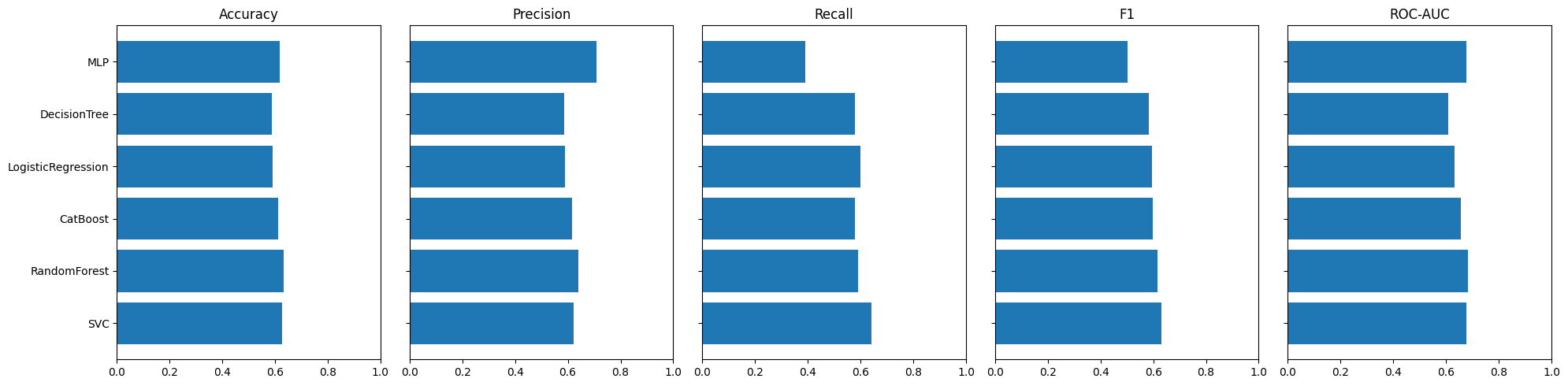
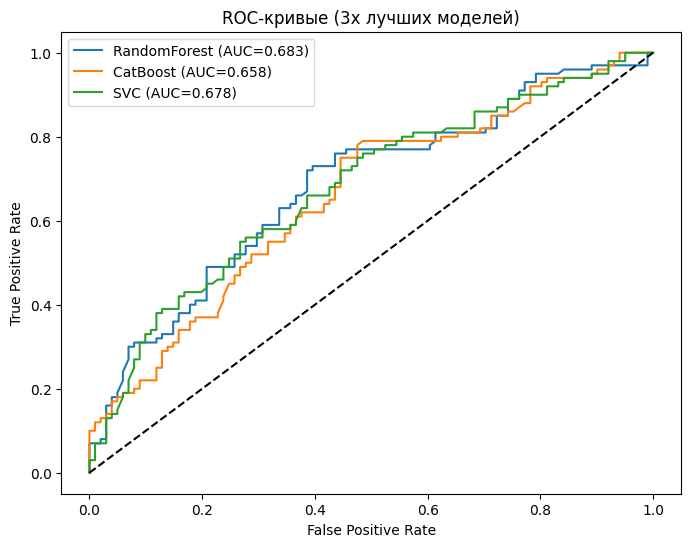
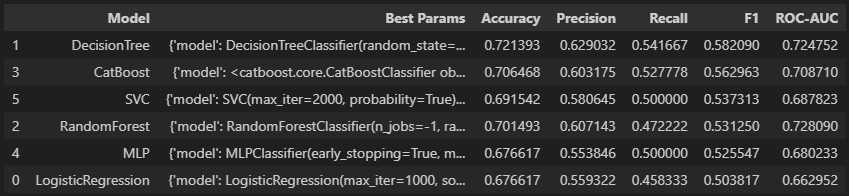
Графики сравнения моделей по метрикам  


График ROC кривых



## Предсказание: SI > 8

Метрики для каждой модели с наиболее удачными наборами гиперпараметров.

Графики сравнения моделей по метрикам

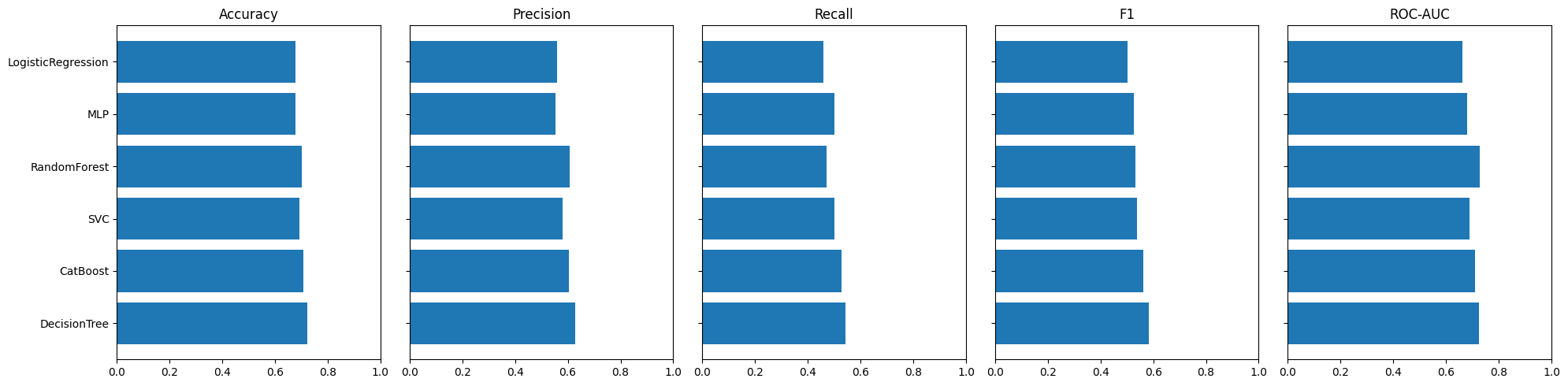
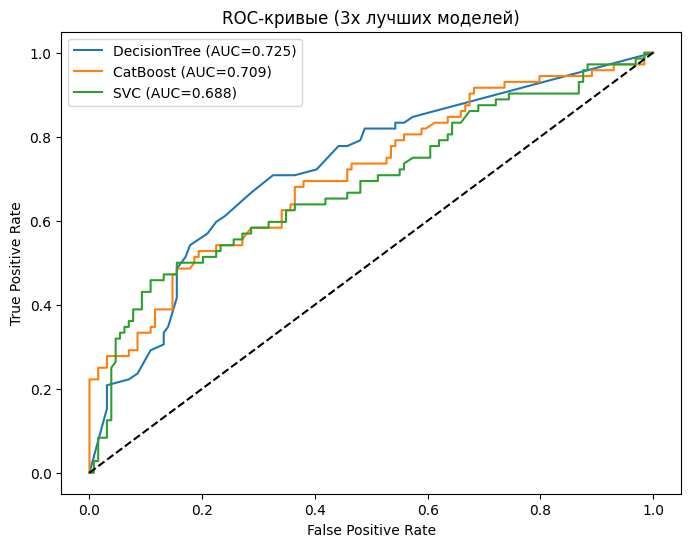


График ROC кривых



# Вывод

Как для задач классификации, так и для задач регрессии CatBoost показывает хорошие результаты по большинству метрик (часто лучшие).

Для дальнейшего улучшения можно выполнить следующие шаги:

* Расширить гиперпараметрический поиск для моделей с лучшими результатами по каждой конкретной задаче.
* Попробовать использовать ансамблирование.
* протестировать дополнительные бустинговые алгоритмы (например, LightGBM, XGBoost) для сравнения.
* Побробовать построить новые производные признаки, чтобы усилить сигнал в данных.

На данном этапе, сложно выявить однозначно лучшую модель.

Однако можно подчеркнуть, что простые модели, такие как линейная регрессия (LinearRegression) и дерево решений (DecisionTree) показали неспособность выявить сложные зависимости в данных. Это касается задач регрессии.

Для задач бинарной классификации простые модели (DecisionTree и LogisticRegression) зачастую уступают более сложным методам.

Также, для задач классификации стоит попробовать ансамблирование SVC и CatBoost, т.к. они показывают хорошие результаты.