Исследование методов классификации

Фролова М.В. ИСТмд-11

Цель: исследовать различные классификаторы посредством их применения на одном наборе данных.

Исходные данные:

1. Датасет: файл ‘Coffee_sales.csv’, размер 3547x11.
2. Целевая переменная: ‘coffee_name’.
3. Признаки: ‘hour_of_day’, ‘money’, ‘Weekdaysort’, ‘Monthsort’.
4. Размер тестовой выборки - 30%.
5. Исследуемые классификаторы: RandomForestClassifier(), KNeighborsClassifier(), DecisionTreeClassifier(), AdaBoostClassifier(), GaussianNB().
6. Используемые библиотеки:

- import pandas as pd
- import matplotlib.pyplot as plt
- import sklearn
(модули .preprocessing, .model_selection, .metrics, .neighbors, .tree, .ensemble, .naive_bayes)

Описание используемых классификаторов

1. RandomForestClassifier() -> Метод «Случайный лес»

Суть «Random Forest»: алгоритм создает деревья решений, каждое из которых строится на случайном подмножестве обучающих данных и случайном подмножестве признаков. Затем он получает прогноз от каждого дерева и выбирает наилучшее решение посредством голосования.

2. KNeighborsClassifier() -> Метод «k-ближайших соседей»

Суть «K-Nearest Neighbors» (kNN): для каждого объекта из тестовой выборки находится k ближайших соседей из обучающей выборки, и алгоритм классифицирует объект на основе классов его соседей. Класс, который наиболее часто встречается среди соседей, и будет являться классом, к которому относится исходный объект.

3. DecisionTreeClassifier() -> Метод «Дерево решений»

Суть «DecisionTree»: алгоритм строит бинарное дерево, в котором каждый внутренний узел представляет собой условие о признаках, а листья - конечный результат его работы, то есть принадлежность к определённому классу.

Описание используемых классификаторов

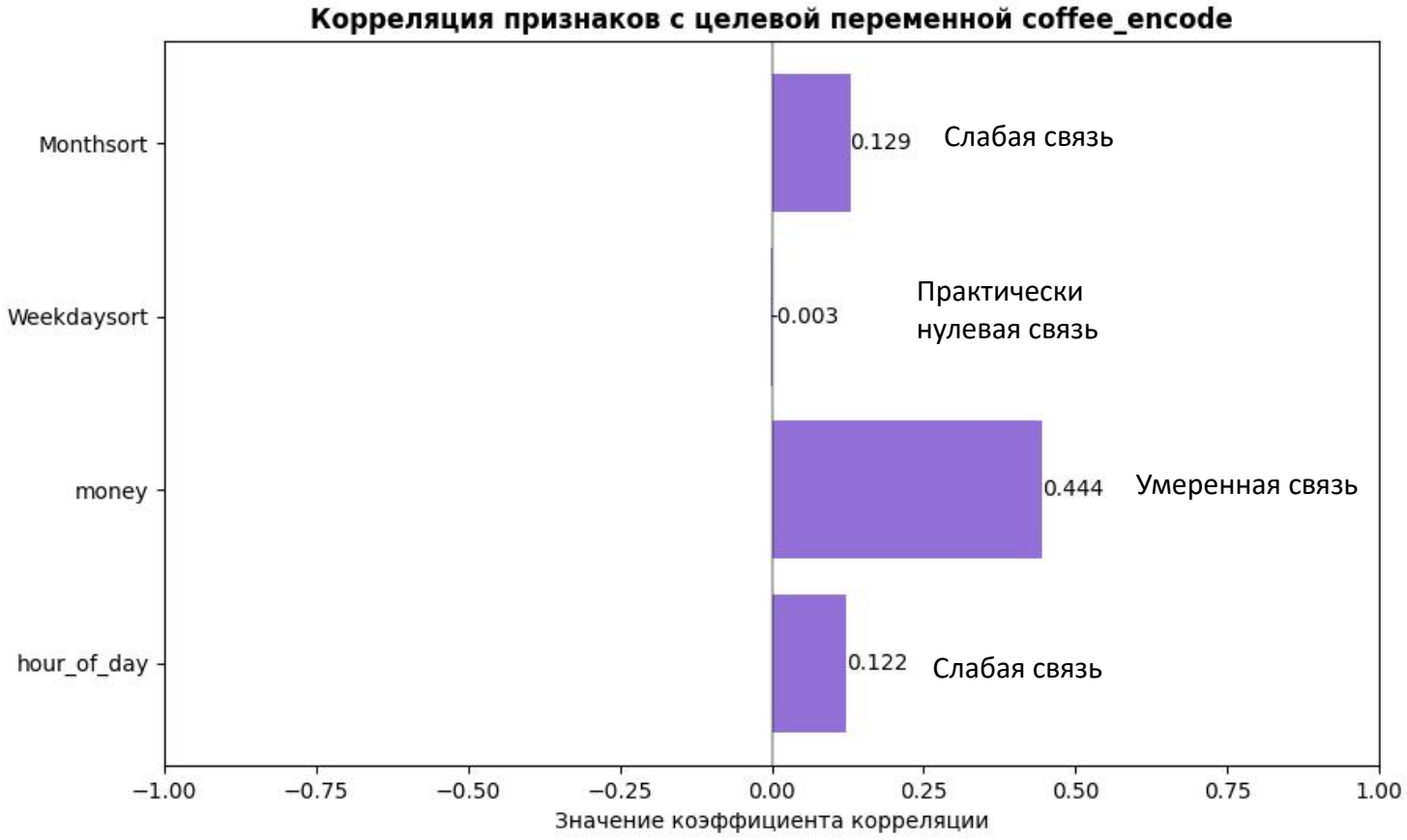
4. AdaBoostClassifier() -> Метод «Адаптивный бустинг»

Суть «AdaBoost»: в качестве базовой модели обычно используется пень решений - дерево с глубиной 1, которому присваивается вектор весов размера N, каждое значение которого соответствует определённому обучающему примеру выборки и изначально равно 1 / N, где N - количество образцов в обучающей выборке. Каждый следующий пень обучается с учётом весов, рассчитанных на основе ошибок предыдущей модели. Также для каждого обученного пня отдельно рассчитывается вес, используемый для оценки важности в итоговом прогнозе.

5. GaussianNB() -> Метод «Наивный Байес»

Суть «Gaussian Naive Bayes»: алгоритм основан на теореме Байеса с “наивным” предположением об условной независимости между каждой парой признаков при заданном значении переменной класса, что упрощает расчёт вероятностей для каждой гипотезы.

Связь признаков с целевой переменной



Таким образом, по диаграмме видим, что самым значимым признаком является money (цена), самым незначимым - weekdaysort (день недели).

Проведём троекратное тестирование для оценки классификаторов. Диаграмма сравнения точности (accuracy) классификаторов №1

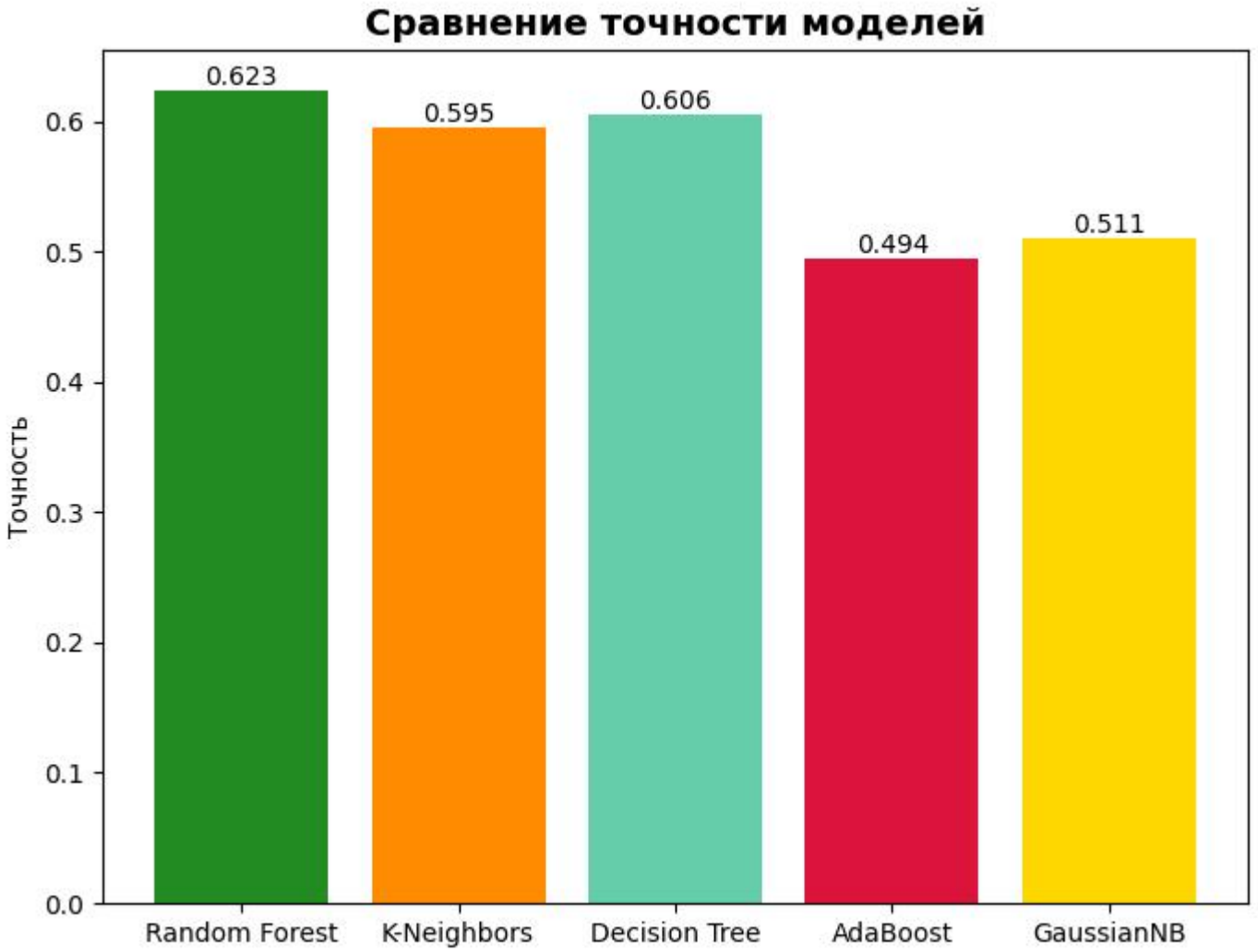


Диаграмма сравнения точности классификаторов №2

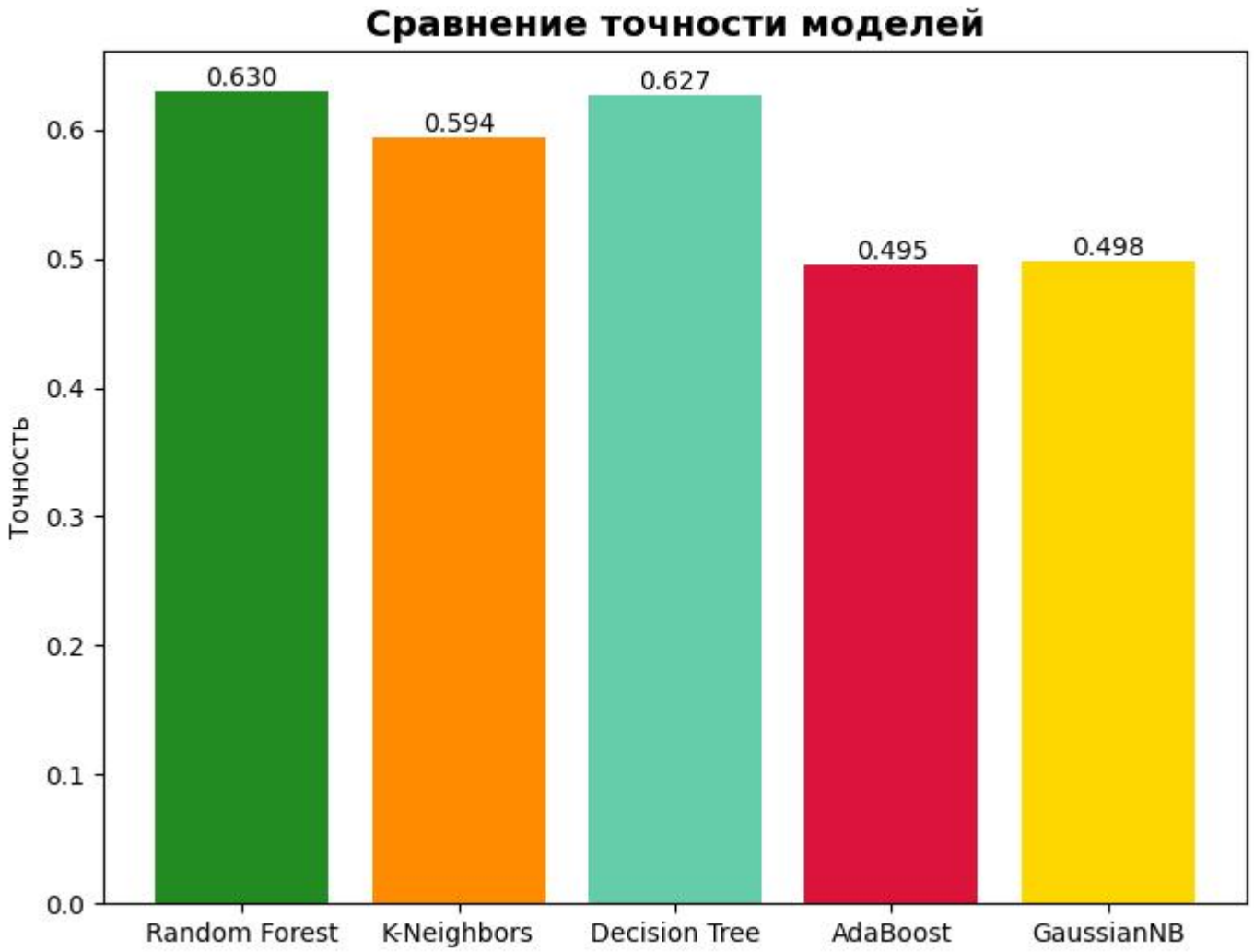
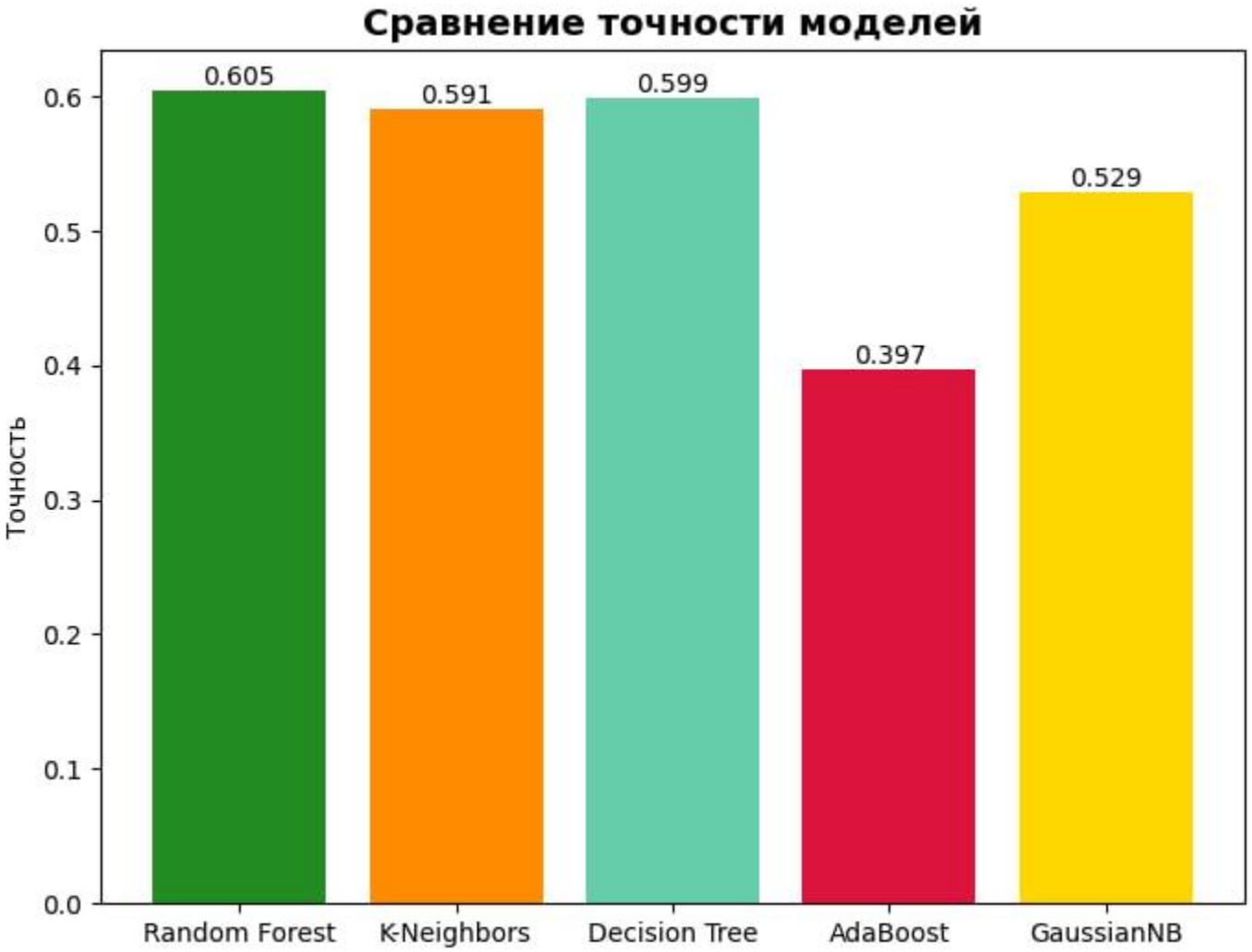


Диаграмма сравнения точности классификаторов №3



Сопоставив диаграммы, можно сделать вывод, что получаемая точность моделей относительно стабильна, за исключением AdaBoost. Это связано с последовательной природой алгоритма.

Самая высокая точность моделей у классификатора RandomForestClassifier(), затем следуют DecisionTreeClassifier() и KNeighborsClassifier(). Точность GaussianNB() заметно ниже.
Самая низкая точность у моделей AdaBoostClassifier().

Метрики для оценки моделей

```
Результаты:

Random Forest
Accuracy: 0.62723
F1-Score: 0.62695
Precision: 0.62799
Recall: 0.62723

K-Neighbors
Accuracy: 0.60939
F1-Score: 0.61276
Precision: 0.61903
Recall: 0.60939

Decision Tree
Accuracy: 0.61221
F1-Score: 0.61467
Precision: 0.62110
Recall: 0.61221

AdaBoost
Accuracy: 0.43850
F1-Score: 0.38091
Precision: 0.38824
Recall: 0.43850

GaussianNB
Accuracy: 0.54178
F1-Score: 0.48919
Precision: 0.47194
Recall: 0.54178
```

Accuracy - точность, представляющая собой долю правильных предсказаний среди всех предсказаний модели.

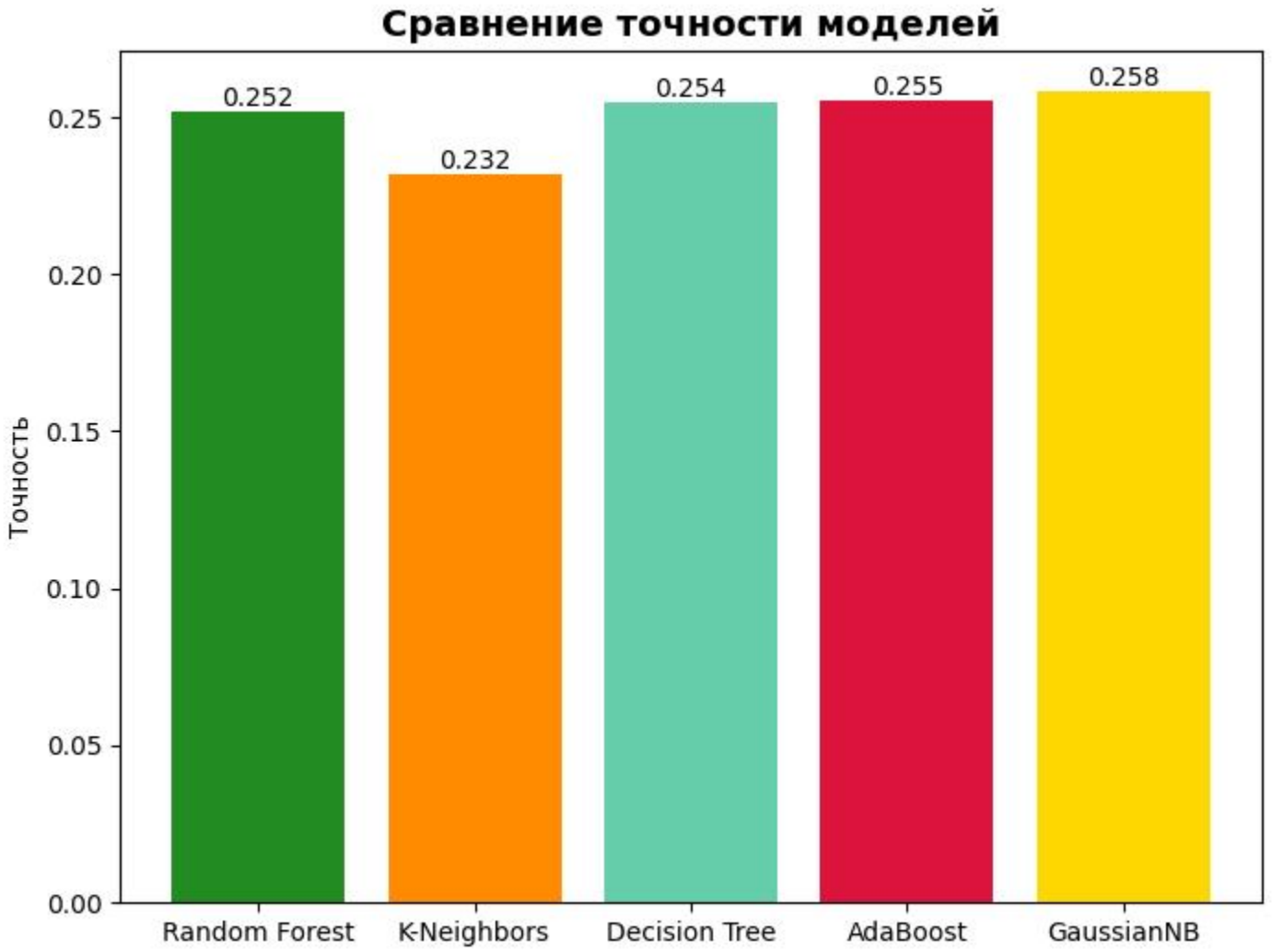
Precision - точность положительных предсказаний, то есть доля объектов, которые действительно принадлежат предсказанному моделью классу.

Recall - полнота, представляющая собой долю объектов класса, которые модель смогла правильно отнести к этому классу.

F1-Score - гармоническое среднее между Precision и Recall, позволяющее сбалансировать их значения.

По результатам вычисления метрик, можно заметить, что метрики у моделей Random Forest, K-Neighbors, Decision Tree практически совпадают, а у AdaBoost и GaussianNB отличаются - метрика Accuracy больше, чем F1-Score, то есть модели в основном предсказывают самые частые классы.

Уберём самый значимый признак «money» и сравним результаты



Результаты:

Random Forest
Accuracy: 0.25164
F1-Score: 0.25104
Precision: 0.25130
Recall: 0.25164

K-Neighbors
Accuracy: 0.23192
F1-Score: 0.23268
Precision: 0.23623
Recall: 0.23192

Decision Tree
Accuracy: 0.25446
F1-Score: 0.24594
Precision: 0.24917
Recall: 0.25446

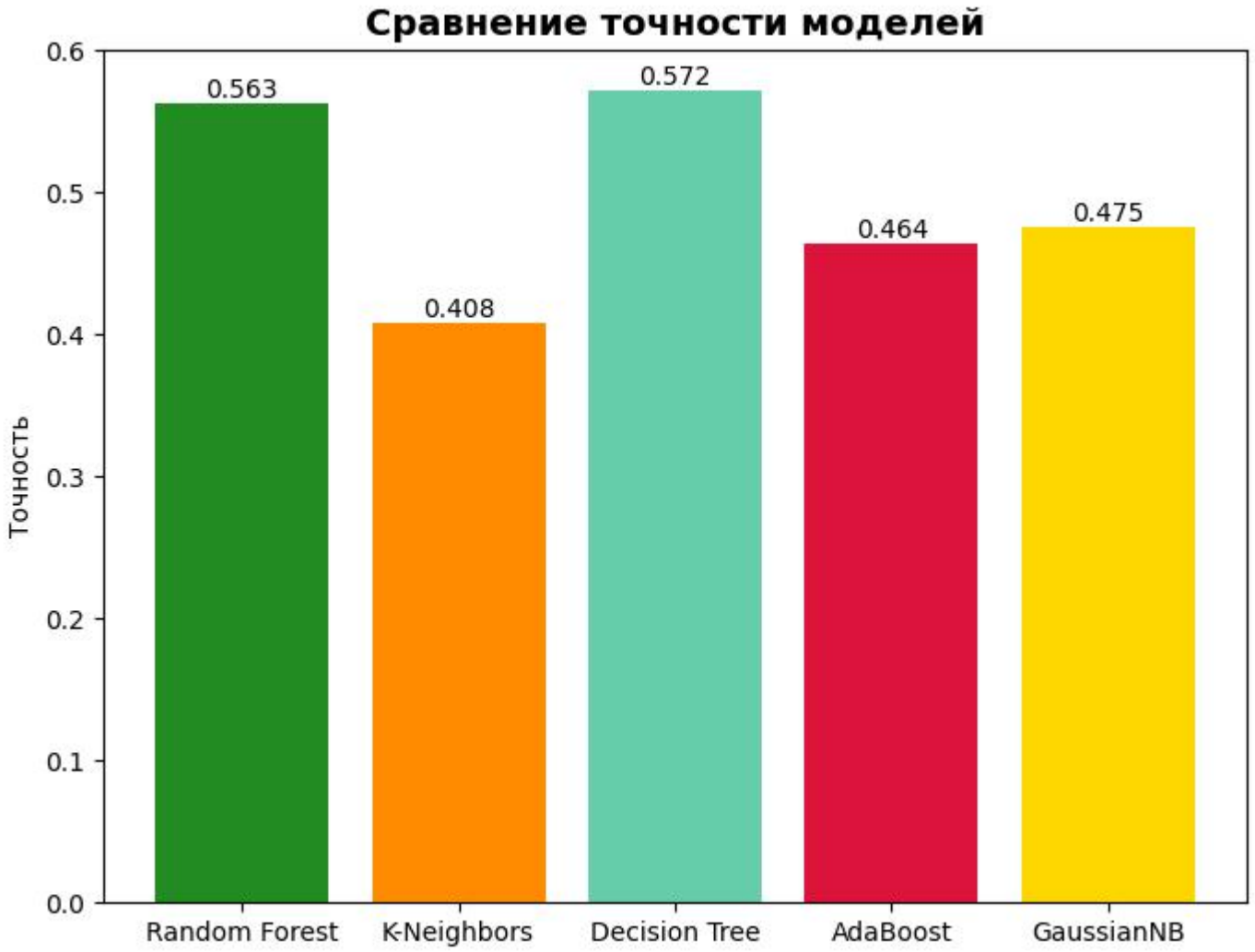
AdaBoost
Accuracy: 0.25540
F1-Score: 0.18652
Precision: 0.18235
Recall: 0.25540

GaussianNB
Accuracy: 0.25822
F1-Score: 0.20020
Precision: 0.18240
Recall: 0.25822

По значениям метрик и диаграмме видно, что точность моделей уменьшилась более, чем в 2 раза и стала примерно одинаковой у всех моделей.

Кроме того, у модели K-Neighbors оказалась самая низкая доля правильных предсказаний. Это связано с тем, что после исключения «money» оставшиеся признаки недостаточно хорошо разделяют классы, что необходимо для KNeighborsClassifier().

Теперь оставим только самый значимый признак - «money» и посмотрим на результаты




```
Результаты:

Random Forest
Accuracy: 0.53897
F1-Score: 0.43030
Precision: 0.38613
Recall: 0.53897

K-Neighbors
Accuracy: 0.44695
F1-Score: 0.39776
Precision: 0.48538
Recall: 0.44695

Decision Tree
Accuracy: 0.53897
F1-Score: 0.43030
Precision: 0.38613
Recall: 0.53897

AdaBoost
Accuracy: 0.44977
F1-Score: 0.32894
Precision: 0.34647
Recall: 0.44977

GaussianNB
Accuracy: 0.46573
F1-Score: 0.35806
Precision: 0.30052
Recall: 0.46573
```

По значениям метрик и диаграмме видно, что точность моделей несколько уменьшилась. Также стоит отметить, что значения accurancy и f1-score довольно отличаются у всех моделей.

У Random Forest и Decision Tree сильно различаются метрики Precision и Recall. Это говорит о том, несмотря на правильные предсказания, модели также делают много ложных предсказаний.

У K-Neighbors самая низкая доля правильных предсказаний
<- один признак, однако Precision, напротив, больше чем Recall.

Выводы:

- Для датасета ‘Coffee_sales.csv’ лучшие результаты показал классификатор RandomForestClassifier(), использующий метод случайного леса. Худшие результаты оказались у AdaBoostClassifier(), использующего метод адаптивного бустинга, и GaussianNB(), использующий метод наивного Байеса.

- Наиболее восприимчивым к количеству признаков является классификатор KNeighborsClassifier().

- У моделей, обученных классификаторами GaussianNB() и AdaBoostClassifier(), в ходе тестирования метрика Recall всегда оказывалась больше, чем Precision, что говорит о наличии большого количества ложных предсказаний.

- Самые нестабильные результаты точности предсказаний показывал классификатор AdaBoostClassifier().

- Самые стабильные результаты точности предсказаний показывали классификаторы RandomForestClassifier() и DecisionTreeClassifier(), использующие деревья решений.

- У моделей, обученных классификатором KNeighborsClassifier(), во всех рассмотренных в ходе исследования случаях Precision > Recall, то есть модели реже ошибаются в предсказаниях, но много пропускают.