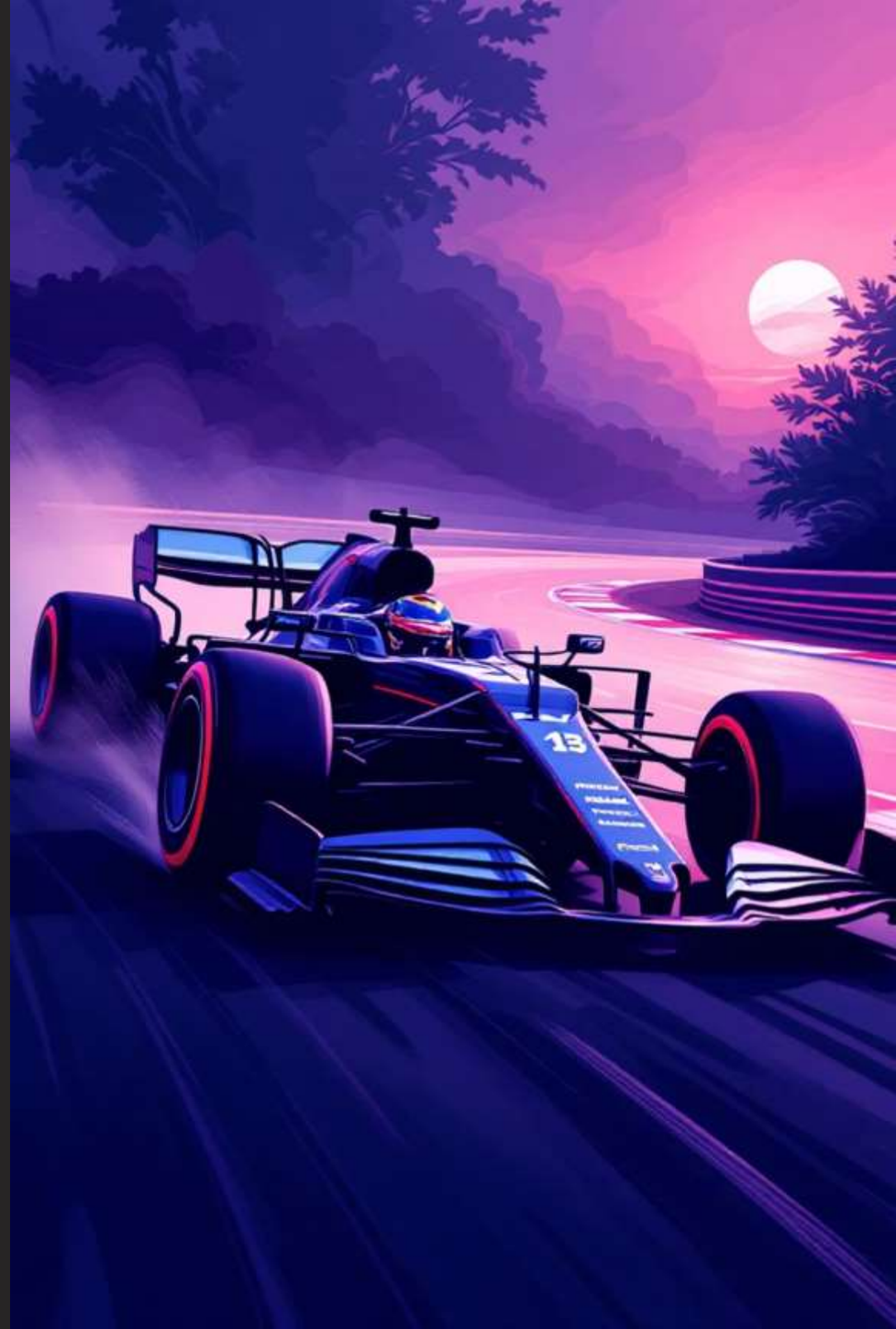
Подбор гиперпараметров и интерпретация модели ML на на датасете Формулы 1

Классификация на попадание в топ-3 подиума

Шароченкова Софья Максимовна

309 группа



Цели и задачи лабораторной работы

Выбор и сравнение моделей

Определить оптимальную модель (Decision Tree, Random Forest, SVM, KNN, Boosting) для задачи, с акцентом на Random Forest.

Настройка гиперпараметров

Подбор оптимальных гиперпараметров с использованием GridSearchCV, RandomizedSearchCV и Optuna, а также их сравнительный анализ.

Подготовка данных Formula 1

Объединение и предобработка данных из различных таблиц (results, races, drivers, constructors) для классификации.

Интерпретация модели

Реализация методов локальной (LIME) и глобальной (SHAP) интерпретации для понимания работы модели.

Разработка "калькулятора"

Создание интерактивного инструмента для предсказаний с наглядными объяснениями.

Наша работа сосредоточена на создании robust-модели для предсказания попадания в топ-3 подиума Формулы 1, используя реальные гоночные данные.

Выбор модели: Random Forest

Для решения задачи классификации была выбрана модель **Random Forest**. Эта модель демонстрирует превосходный баланс между точностью предсказаний и устойчивостью к переобучению, а также предоставляет возможности для интерпретации.

Альтернативные модели:

- **Decision Tree:** Простая в понимании, но склонна к переобучению на сложных данных.
- **SVM:** Эффективна для линейно разделимых данных, но может быть вычислительно затратной.
- **KNN:** Легко реализуется, но чувствительна к масштабу признаков и объему данных.
- **Boosting (например, XGBoost):** Часто показывает очень высокую точность, но сложнее в настройке и интерпретации.

Гиперпараметры Random Forest

Параметр	Описание	Диапазон/Значения
n_estimators	Количество деревьев в лесу	10-200
max_depth	Максимальная глубина каждого дерева	None или 1-50
min_samples_split	Минимальное количество образцов для разделения узла	2-10
min_samples_leaf	Минимальное количество образцов в листе	1-5
bootstrap	Использование бутстреп-выборки при построении деревьев	True/False

Правильный подбор этих гиперпараметров критически важен для предотвращения переобучения и повышения обобщающей способности модели. Для Decision Tree можно использовать аналогичные параметры, исключая `n_estimators`.

Датасет: Формула 1 и подготовка данных

Источники данных:

- **results.csv, races.csv, drivers.csv, constructors.csv:** Объединены по соответствующим ID для создания комплексного датасета.

Целевая переменная:

- **podium:** Бинарная переменная (1, если гонщик попал в топ-3, 0 в противном случае).

Основные признаки:

- **grid:** Стартовая позиция гонщика.
- **year:** Год проведения гонки.
- **circuitId:** Идентификатор трассы.
- **constructor_name, driverId:** Имена конструкторов и идентификаторы гонщиков.
- И многие другие, отражающие динамику гонок.

Предварительная обработка:

- Обработка пропущенных значений.
- **OneHotEncoder:** Для категориальных признаков.
- **StandardScaler:** Для масштабирования числовых признаков.
- **Train/test split (80/20):** Разделение данных для обучения и валидации.

Пайплайн: Предобработка и обучение

ColumnTransformer

Объединяет `StandardScaler` для числовых и `OneHotEncoder` для категориальных признаков, обеспечивая корректную предобработку.

Pipeline

Создает последовательность действий: предобработка данных, затем применение `RandomForestClassifier`. Это предотвращает утечки данных и упрощает процесс.

Обучение модели

Пайплайн обучается на тренировочном наборе данных (`fit`) и делает предсказания на тестовом (`predict`).

Метрика Accuracy

Используется для оценки производительности модели, отражая долю правильных предсказаний.

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.ensemble import RandomForestClassifier

preprocessor = ColumnTransformer(
 transformers=[
 ('num', StandardScaler(), num_features),
 ('cat', OneHotEncoder(), cat_features)
 ])

pipeline = Pipeline(steps=[('preprocessor', preprocessor),
 ('classifier', RandomForestClassifier())])
```

Подбор гиперпараметров: GridSearchCV

Метод GridSearchCV:

GridSearchCV — это метод исчерпывающего поиска, который систематически перебирает все возможные комбинации заданных гиперпараметров для нахождения наилучшей.

- **Пространство поиска:**
  - `n_estimators`: [50, 100, 150]
  - `max_depth`: [10, 20, None]
  - `min_samples_split`: [2, 5, 10]
- **Кросс-валидация:** 5-fold для устойчивой оценки.
- **Лучшие параметры (пример):** `n_estimators=100`, `max_depth=20`.
- **Accuracy на тестовой выборке (пример):** 0.85.

Подбор гиперпараметров: RandomizedSearchCV и Optuna

RandomizedSearchCV

Использует случайный поиск по заданным распределениям гиперпараметров.
Пространство поиска: Распределения (например, равномерное для max_depth).
Итерации: 50.
Лучшие параметры (пример): `n_estimators=120`, `max_depth=25`.
Accuracy (пример): 0.86.
Преимущества: Быстрее, чем GridSearchCV, часто находит хорошие решения.

Optuna

Реализует байесовскую оптимизацию, которая "учится" на предыдущих итерациях.
Метод: Байесовская оптимизация, 50 trials.
Лучшие параметры (пример): `n_estimators=150`, `max_depth=30`.
Accuracy (пример): 0.87.
Преимущества: Самый эффективный метод, часто находит наилучшие параметры за меньшее время.

Optuna показала наилучшие результаты, найдя более точную модель за меньшее время благодаря своей "умной" стратегии поиска.

Сравнение методов и выбор лучшей модели

Метод	Лучшие параметры	Accuracy (пример)	Время поиска (пример)
GridSearchCV	n_estimators=100, max_depth=20	0.85	10 мин
RandomizedSearchCV	n_estimators=120, max_depth=25	0.86	5 мин
Optuna	n_estimators=150, max_depth=30	0.87	3 мин

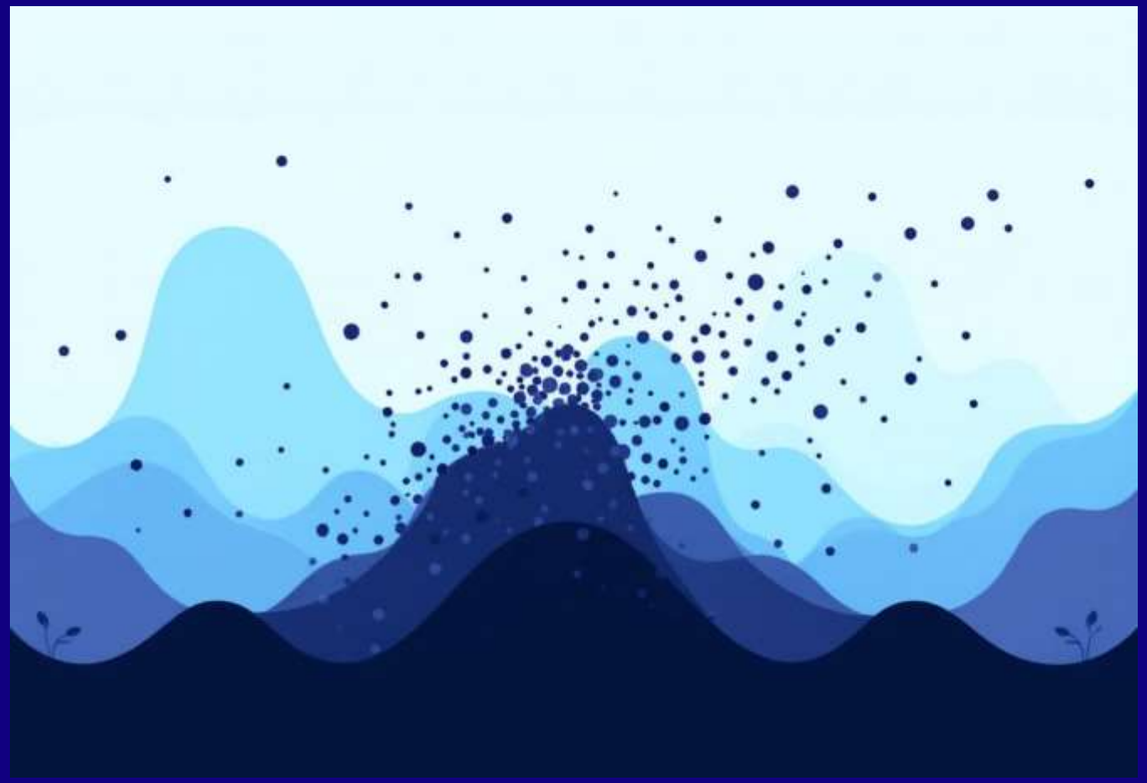
На основе проведенного сравнения, модель, оптимизированная с помощью **Optuna**, показала наилучшую комбинацию высокой точности (0.87) и минимального времени поиска. Эта модель была сохранена в формате `.joblib` для дальнейшего использования.

Интерпретация модели: SHAP и LIME

Глобальная интерпретация (SHAP)

SHAP (SHapley Additive exPlanations) позволяет понять, какие признаки оказывают наибольшее влияние на предсказания модели в целом.

- **Summary plot:** Визуализирует важность признаков и их влияние на весь датасет. Например, `grid` (стартовая позиция) часто является самым важным признаком.



Локальная интерпретация (LIME)

LIME (Local Interpretable Model-agnostic Explanations) объясняет предсказания отдельного экземпляра, создавая локальную, интерпретируемую модель вокруг этого предсказания.

- **LimeTabularExplainer:** Показывает вклад каждого признака в конкретное предсказание. Например, для гонщика, стартующего с 1-й позиции, LIME покажет, что `grid +0.5` значительно увеличивает вероятность попадания на подиум.