**Московский государственный технический   
университет им. Н. Э. Баумана**

Курс «Технологии машинного обучения»

Отчёт по рубежному контролю №2

«Методы построения моделей машинного обучения.»

Вариант № 7

Выполнил: Проверил:  
Горенков А.А. Гапанюк Ю.Е.  
группа ИУ5-63Б

Дата: 15.04.25 Дата:

Подпись: Подпись:

Москва, 2025 г.

**Задание:**

Номер варианта: **7**

Номер набора данных, указанного в задаче: **7**

<https://www.kaggle.com/mohansacharya/graduate-admissions>

Метод №1: **Дерево решений**

Метод №2: **Случайный лес**

**Ход выполнения:**

|  |
| --- |
| *# Анализ данных о поступлении в магистратуру*  *# Построение моделей: Дерево решений и Случайный лес*  **import** pandas **as** pd **import** numpy **as** np **from** sklearn.model\_selection **import** train\_test\_split **from** sklearn.tree **import** DecisionTreeRegressor **from** sklearn.ensemble **import** RandomForestRegressor **from** sklearn.metrics **import** mean\_squared\_error, mean\_absolute\_error, r2\_scor **from** sklearn.preprocessing **import** StandardScaler **import** matplotlib.pyplot **as** plt **import** seaborn **as** sns  *# Загрузка данных*  *# Предполагается, что файл CSV уже доступен* df **=** pd**.**read\_csv('Admission\_Predict\_Ver1.1.csv')  *# Первичный анализ данных* print("Размер датасета:", df**.**shape) print("\nИнформация о данных:") print(df**.**info()) print("\nПервые 5 строк:") print(df**.**head()) print("\nСтатистическое описание:") print(df**.**describe())  *# Проверка на пропущенные значения* print("\nПропущенные значения:") print(df**.**isnull()**.**sum())  *# Анализ целевой переменной*  print("\nРаспределение целевой переменной 'Chance of Admit':") print(df['Chance of Admit ']**.**describe())  *# Визуализация данных* plt**.**figure(figsize**=**(15, 10))  *# Распределение целевой переменной* plt**.**subplot(2, 3, 1) plt**.**hist(df['Chance of Admit '], bins**=**30, edgecolor**=**'black') plt**.**title('Распределение вероятности поступления') plt**.**xlabel('Вероятность поступления') plt**.**ylabel('Частота')  *# Корреляционная матрица* plt**.**subplot(2, 3, 2)  *# Удаляем столбец Serial No. для корреляции* corr\_data **=** df**.**drop('Serial No.', axis**=**1) corr\_matrix **=** corr\_data**.**corr() sns**.**heatmap(corr\_matrix, annot**=True**, cmap**=**'coolwarm', center**=**0) plt**.**title('Корреляционная матрица')  *# Boxplot для некоторых признаков* plt**.**subplot(2, 3, 3) |

]

|  |
| --- |
| plt**.**boxplot([df['GRE Score'], df['TOEFL Score']**/**10, df['CGPA']], labels**=**['GRE', 'TOEFL/10', 'CGPA']) plt**.**title('Boxplot основных признаков')  *# Scatter plot: GRE vs Chance of Admit* plt**.**subplot(2, 3, 4) plt**.**scatter(df['GRE Score'], df['Chance of Admit '], alpha**=**0.6) plt**.**xlabel('GRE Score') plt**.**ylabel('Chance of Admit')  plt**.**title('GRE Score vs Chance of Admit')  *# Scatter plot: CGPA vs Chance of Admit* plt**.**subplot(2, 3, 5) plt**.**scatter(df['CGPA'], df['Chance of Admit '], alpha**=**0.6) plt**.**xlabel('CGPA') plt**.**ylabel('Chance of Admit') plt**.**title('CGPA vs Chance of Admit')  *# Research влияние* plt**.**subplot(2, 3, 6) research\_groups **=** df**.**groupby('Research')['Chance of Admit ']**.**mean() plt**.**bar(['No Research', 'With Research'], research\_groups**.**values) plt**.**title('Влияние исследовательского опыта') plt**.**ylabel('Средняя вероятность поступления')  plt**.**tight\_layout() plt**.**show()  *# Подготовка данных для моделирования*  *# Удаляем Serial No. как он не несет информационной нагрузки* X **=** df**.**drop(['Serial No.', 'Chance of Admit '], axis**=**1) y **=** df['Chance of Admit ']  print("\nПризнаки для модели:") print(X**.**columns**.**tolist()) print("\nРазмер признаков:", X**.**shape) print("Размер целевой переменной:", y**.**shape)  *# Проверим, нужна ли стандартизация* print("\nСтатистика признаков:") print(X**.**describe())  *# Разделение данных на обучающую и тестовую выборки*  X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2, ran  print(f"\nРазмер обучающей выборки: {X\_train**.**shape}") print(f"Размер тестовой выборки: {X\_test**.**shape}")  *# МОДЕЛЬ 1: ДЕРЕВО РЕШЕНИЙ* print("\n" **+** "="**\***50) print("МОДЕЛЬ 1: ДЕРЕВО РЕШЕНИЙ") print("="**\***50)  *# Создание и обучение модели дерева решений* dt\_model **=** DecisionTreeRegressor( max\_depth**=**10, *# Ограничиваем глубину для избежания переобучени* |

|  |
| --- |
| min\_samples\_split**=**5, *# Минимальное количество образцов для разделения*  min\_samples\_leaf**=**3, *# Минимальное количество образцов в листе*  random\_state**=**42  ) dt\_model**.**fit(X\_train, y\_train)  *# Предсказания*  y\_train\_pred\_dt **=** dt\_model**.**predict(X\_train) y\_test\_pred\_dt **=** dt\_model**.**predict(X\_test)  *# Метрики качества для дерева решений*  dt\_train\_mse **=** mean\_squared\_error(y\_train, y\_train\_pred\_dt) dt\_test\_mse **=** mean\_squared\_error(y\_test, y\_test\_pred\_dt) dt\_train\_mae **=** mean\_absolute\_error(y\_train, y\_train\_pred\_dt) dt\_test\_mae **=** mean\_absolute\_error(y\_test, y\_test\_pred\_dt) dt\_train\_r2 **=** r2\_score(y\_train, y\_train\_pred\_dt) dt\_test\_r2 **=** r2\_score(y\_test, y\_test\_pred\_dt)  print("Метрики качества дерева решений:") print(f"Обучающая выборка - MSE: {dt\_train\_mse:.6f}, MAE: {dt\_train\_mae:.6f print(f"Тестовая выборка - MSE: {dt\_test\_mse:.6f}, MAE: {dt\_test\_mae:.6f}, R  *# МОДЕЛЬ 2: СЛУЧАЙНЫЙ ЛЕС* print("\n" **+** "="**\***50) print("МОДЕЛЬ 2: СЛУЧАЙНЫЙ ЛЕС") print("="**\***50)  *# Создание и обучение модели случайного леса* rf\_model **=** RandomForestRegressor( n\_estimators**=**100, *# Количество деревьев*  max\_depth**=**15, *# Максимальная глубина деревьев*  min\_samples\_split**=**5, *# Минимальное количество образцов для разделения*  min\_samples\_leaf**=**2, *# Минимальное количество образцов в листе*  random\_state**=**42, n\_jobs**=-**1 *# Использовать все доступные процессоры*  ) rf\_model**.**fit(X\_train, y\_train)  *# Предсказания*  y\_train\_pred\_rf **=** rf\_model**.**predict(X\_train) y\_test\_pred\_rf **=** rf\_model**.**predict(X\_test)  *# Метрики качества для случайного леса*  rf\_train\_mse **=** mean\_squared\_error(y\_train, y\_train\_pred\_rf) rf\_test\_mse **=** mean\_squared\_error(y\_test, y\_test\_pred\_rf) rf\_train\_mae **=** mean\_absolute\_error(y\_train, y\_train\_pred\_rf) rf\_test\_mae **=** mean\_absolute\_error(y\_test, y\_test\_pred\_rf) rf\_train\_r2 **=** r2\_score(y\_train, y\_train\_pred\_rf) rf\_test\_r2 **=** r2\_score(y\_test, y\_test\_pred\_rf)  print("Метрики качества случайного леса:") print(f"Обучающая выборка - MSE: {rf\_train\_mse:.6f}, MAE: {rf\_train\_mae:.6f print(f"Тестовая выборка - MSE: {rf\_test\_mse:.6f}, MAE: {rf\_test\_mae:.6f}, R |

}

|  |
| --- |
| *# СРАВНЕНИЕ МОДЕЛЕЙ* print("\n" **+** "="**\***50) print("СРАВНЕНИЕ МОДЕЛЕЙ") print("="**\***50)  *# Создание таблицы сравнения* comparison\_data **=** {  'Модель': ['Дерево решений', 'Случайный лес'],  'MSE (тест)': [dt\_test\_mse, rf\_test\_mse],  'MAE (тест)': [dt\_test\_mae, rf\_test\_mae],  'R² (тест)': [dt\_test\_r2, rf\_test\_r2],  'Переобучение (R²)': [dt\_train\_r2 **-** dt\_test\_r2, rf\_train\_r2 **-** rf\_test\_r2  }  comparison\_df **=** pd**.**DataFrame(comparison\_data)  print(comparison\_df**.**to\_string(index**=False**, float\_format**=**'%.6f'))  *# Важность признаков* print("\n" **+** "="**\***30) print("ВАЖНОСТЬ ПРИЗНАКОВ") print("="**\***30)  *# Для дерева решений*  dt\_feature\_importance **=** pd**.**DataFrame({  'Признак': X**.**columns,  'Важность': dt\_model**.**feature\_importances\_ })**.**sort\_values('Важность', ascending**=False**)  print("\nВажность признаков (Дерево решений):")  print(dt\_feature\_importance**.**to\_string(index**=False**, float\_format**=**'%.4f'))  *# Для случайного леса*  rf\_feature\_importance **=** pd**.**DataFrame({  'Признак': X**.**columns,  'Важность': rf\_model**.**feature\_importances\_ })**.**sort\_values('Важность', ascending**=False**)  print("\nВажность признаков (Случайный лес):")  print(rf\_feature\_importance**.**to\_string(index**=False**, float\_format**=**'%.4f'))  *# ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ* plt**.**figure(figsize**=**(15, 12))  *# График 1: Сравнение предсказаний с истинными значениями* plt**.**subplot(2, 3, 1) plt**.**scatter(y\_test, y\_test\_pred\_dt, alpha**=**0.6, label**=**'Дерево решений') plt**.**plot([y\_test**.**min(), y\_test**.**max()], [y\_test**.**min(), y\_test**.**max()], 'r--', plt**.**xlabel('Истинные значения') plt**.**ylabel('Предсказанные значения') plt**.**title('Дерево решений: Предсказания vs Истинные значения') plt**.**legend()  plt**.**subplot(2, 3, 2) plt**.**scatter(y\_test, y\_test\_pred\_rf, alpha**=**0.6, label**=**'Случайный лес', color plt**.**plot([y\_test**.**min(), y\_test**.**max()], [y\_test**.**min(), y\_test**.**max()], 'r--', plt**.**xlabel('Истинные значения') |
| plt**.**ylabel('Предсказанные значения') plt**.**title('Случайный лес: Предсказания vs Истинные значения') plt**.**legend()  *# График 2: Распределение остатков* plt**.**subplot(2, 3, 3) residuals\_dt **=** y\_test **-** y\_test\_pred\_dt residuals\_rf **=** y\_test **-** y\_test\_pred\_rf plt**.**hist(residuals\_dt, alpha**=**0.7, label**=**'Дерево решений', bins**=**20) plt**.**hist(residuals\_rf, alpha**=**0.7, label**=**'Случайный лес', bins**=**20) plt**.**xlabel('Остатки') plt**.**ylabel('Частота') plt**.**title('Распределение остатков') plt**.**legend()  *# График 3: Важность признаков для случайного леса* plt**.**subplot(2, 3, 4) plt**.**barh(rf\_feature\_importance['Признак'], rf\_feature\_importance['Важность' plt**.**xlabel('Важность') plt**.**title('Важность признаков (Случайный лес)') plt**.**gca()**.**invert\_yaxis()  *# График 4: Сравнение метрик* plt**.**subplot(2, 3, 5) models **=** ['Дерево решений', 'Случайный лес'] mse\_values **=** [dt\_test\_mse, rf\_test\_mse] mae\_values **=** [dt\_test\_mae, rf\_test\_mae]  x **=** np**.**arange(len(models)) width **=** 0.35  plt**.**bar(x **-** width**/**2, mse\_values, width, label**=**'MSE', alpha**=**0.8) plt**.**bar(x **+** width**/**2, mae\_values, width, label**=**'MAE', alpha**=**0.8) plt**.**xlabel('Модели') plt**.**ylabel('Значение метрики') plt**.**title('Сравнение MSE и MAE') plt**.**xticks(x, models) plt**.**legend()  *# График 5: R² score* plt**.**subplot(2, 3, 6) r2\_values **=** [dt\_test\_r2, rf\_test\_r2] plt**.**bar(models, r2\_values, color**=**['blue', 'green'], alpha**=**0.7) plt**.**ylabel('R² Score') plt**.**title('Сравнение R² Score') plt**.**ylim(0, 1)  *# Добавляем значения на столбцы* **for** i, v **in** enumerate(r2\_values): plt**.**text(i, v **+** 0.01, f'{v:.3f}', ha**=**'center', va**=**'bottom')  plt**.**tight\_layout() plt**.**show()  *# ВЫВОДЫ И АНАЛИЗ* print("\n" **+** "="**\***60) |

|  |
| --- |
| print("ВЫВОДЫ И АНАЛИЗ РЕЗУЛЬТАТОВ") print("="**\***60)  print("\n1. ВЫБОР МЕТРИК КАЧЕСТВА:") print(" • MSE (Mean Squared Error) - штрафует большие ошибки сильнее") print(" • MAE (Mean Absolute Error) - более устойчива к выбросам") print(" • R² (коэффициент детерминации) - показывает долю объясненной дисп  print(f"\n2. КАЧЕСТВО МОДЕЛЕЙ:") print(f" Дерево решений:") print(f" • MSE: {dt\_test\_mse:.6f}") print(f" • MAE: {dt\_test\_mae:.6f}") print(f" • R²: {dt\_test\_r2:.6f}")  print(f" • Переобучение: {dt\_train\_r2 **-** dt\_test\_r2:.6f}")  print(f"\n Случайный лес:") print(f" • MSE: {rf\_test\_mse:.6f}") print(f" • MAE: {rf\_test\_mae:.6f}") print(f" • R²: {rf\_test\_r2:.6f}")  print(f" • Переобучение: {rf\_train\_r2 **-** rf\_test\_r2:.6f}")  print(f"\n3. СРАВНЕНИЕ И ВЫВОДЫ:") **if** rf\_test\_r2 **>** dt\_test\_r2:  print(" • Случайный лес показывает лучше качество на тестовых данных" **else**: print(" • Дерево решений показывает лучше качество на тестовых данных"  **if** abs(rf\_train\_r2 **-** rf\_test\_r2) **<** abs(dt\_train\_r2 **-** dt\_test\_r2):  print(" • Случайный лес менее склонен к переобучению") **else**: print(" • Дерево решений менее склонно к переобучению")  print(f"\n4. ВАЖНОСТЬ ПРИЗНАКОВ:") print(" Наиболее важные признаки согласно случайному лесу:") **for** i, row **in** rf\_feature\_importance**.**head(3)**.**iterrows(): print(f" • {row['Признак']}: {row['Важность']:.4f}")  print(f"\n5. РЕКОМЕНДАЦИИ:") **if** rf\_test\_r2 **>** dt\_test\_r2:  print(" • Для данной задачи рекомендуется использовать случайный лес" print(" • Случайный лес обеспечивает лучшую обобщающую способность") **else**:  print(" • Для данной задачи можно использовать дерево решений") print(" • Дерево решений более интерпретируемо")  print(f" • Средняя абсолютная ошибка составляет ~{min(dt\_test\_mae, rf\_test print(f" • Модель объясняет ~{max(dt\_test\_r2, rf\_test\_r2)**\***100:.1f}% диспер  *# Дополнительный анализ*  print(f"\n6. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ:") print(f" • Диапазон целевой переменной: [{y**.**min():.3f}, {y**.**max():.3f}]") print(f" • Среднее значение: {y**.**mean():.3f}") print(f" • Стандартное отклонение: {y**.**std():.3f}")  *# Проверка адекватности модели*  baseline\_mse **=** mean\_squared\_error(y\_test, [y\_train**.**mean()] **\*** len(y\_test)) |

)

)

|  |
| --- |
| print(f" • Baseline MSE (среднее): {baseline\_mse:.6f}")  print(f" • Улучшение относительно baseline: {((baseline\_mse **-** min(dt\_test\_ |

Размер датасета: (500, 9)

Информация о данных:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 500 entries, 0 to 499 Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

1. Serial No. 500 non-null int64
2. GRE Score 500 non-null int64
3. TOEFL Score 500 non-null int64
4. University Rating 500 non-null int64
5. SOP 500 non-null float64
6. LOR 500 non-null float64
7. CGPA 500 non-null float64
8. Research 500 non-null int64
9. Chance of Admit 500 non-null float64 dtypes: float64(4), int64(5) memory usage: 35.3 KB None

Первые 5 строк:

Serial No. GRE Score TOEFL Score University Rating SOP LOR CGPA \ 0 1 337 118 4 4.5 4.5 9.65

1. 2 324 107 4 4.0 4.5 8.87
2. 3 316 104 3 3.0 3.5 8.00
3. 4 322 110 3 3.5 2.5 8.67
4. 5 314 103 2 2.0 3.0 8.21

Research Chance of Admit 0 1 0.92

1. 1 0.76
2. 1 0.72
3. 1 0.80
4. 0 0.65

Статистическое описание:

Serial No. GRE Score TOEFL Score University Rating SOP \ count 500.000000 500.000000 500.000000 500.000000 500.000000 mean 250.500000 316.472000 107.192000 3.114000 3.374000 std 144.481833 11.295148 6.081868 1.143512 0.991004 min 1.000000 290.000000 92.000000 1.000000 1.000000 25% 125.750000 308.000000 103.000000 2.000000 2.500000

50% 250.500000 317.000000 107.000000 3.000000 3.500000 75% 375.250000 325.000000 112.000000 4.000000 4.000000 max 500.000000 340.000000 120.000000 5.000000 5.000000

LOR CGPA Research Chance of Admit count 500.00000 500.000000 500.000000 500.00000 mean 3.48400 8.576440 0.560000 0.72174 std 0.92545 0.604813 0.496884 0.14114 min 1.00000 6.800000 0.000000 0.34000 25% 3.00000 8.127500 0.000000 0.63000

50% 3.50000 8.560000 1.000000 0.72000 75% 4.00000 9.040000 1.000000 0.82000 max 5.00000 9.920000 1.000000 0.97000 Пропущенные значения:

Serial No. 0

GRE Score 0

TOEFL Score 0

University Rating 0

SOP 0

LOR 0

CGPA 0

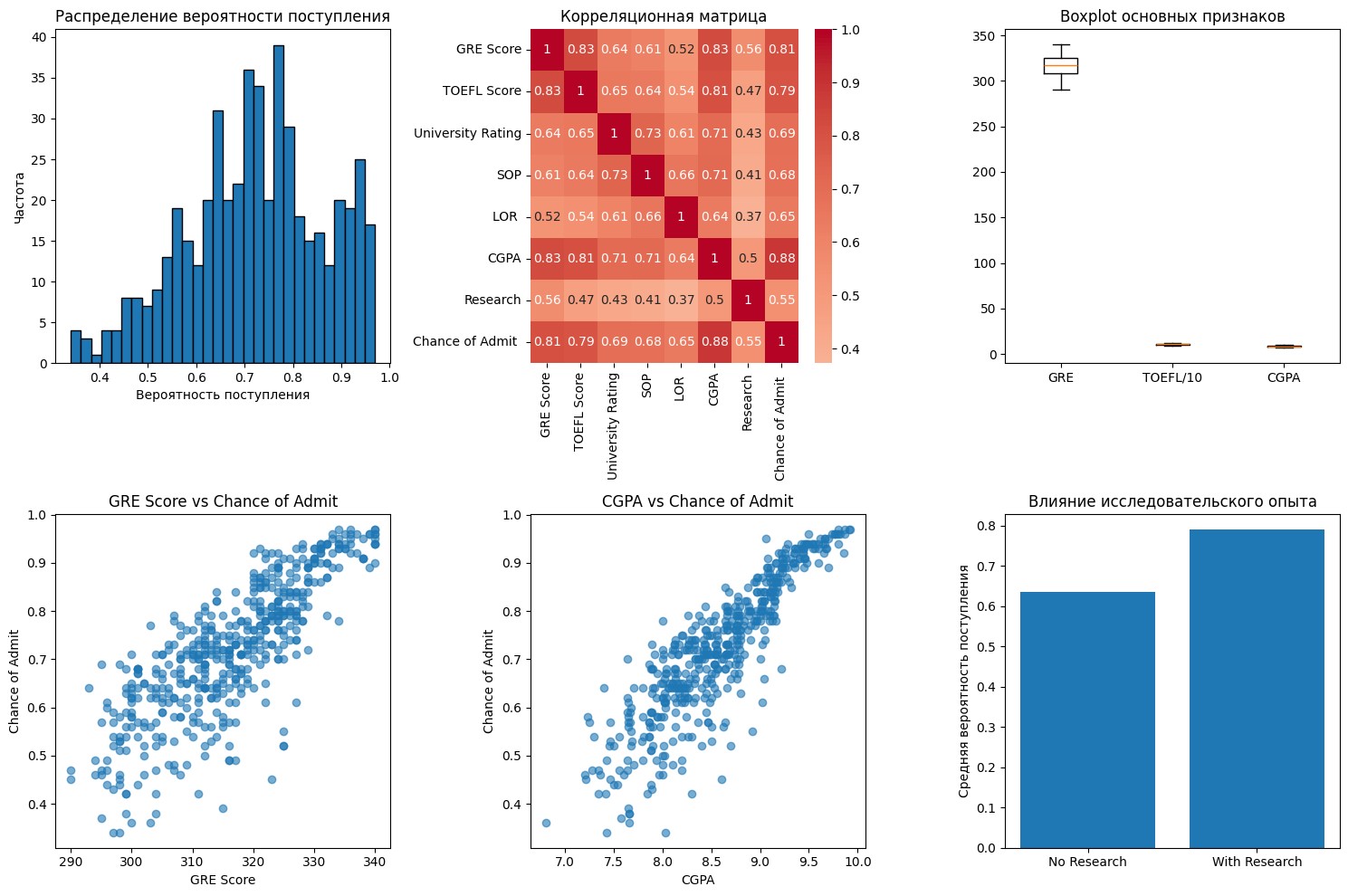
Research 0 Chance of Admit 0 dtype: int64

Распределение целевой переменной 'Chance of Admit':

count 500.00000 mean 0.72174 std 0.14114 min 0.34000 25% 0.63000

50% 0.72000 75% 0.82000 max 0.97000

Name: Chance of Admit , dtype: float64



Признаки для модели:

['GRE Score', 'TOEFL Score', 'University Rating', 'SOP', 'LOR ', 'CGPA', 'Re search']

Размер признаков: (500, 7)

Размер целевой переменной: (500,)

Статистика признаков:

GRE Score TOEFL Score University Rating SOP LOR \ count 500.000000 500.000000 500.000000 500.000000 500.00000 mean 316.472000 107.192000 3.114000 3.374000 3.48400 std 11.295148 6.081868 1.143512 0.991004 0.92545 min 290.000000 92.000000 1.000000 1.000000 1.00000 25% 308.000000 103.000000 2.000000 2.500000 3.00000

50% 317.000000 107.000000 3.000000 3.500000 3.50000 75% 325.000000 112.000000 4.000000 4.000000 4.00000 max 340.000000 120.000000 5.000000 5.000000 5.00000

CGPA Research count 500.000000 500.000000 mean 8.576440 0.560000 std 0.604813 0.496884 min 6.800000 0.000000 25% 8.127500 0.000000

50% 8.560000 1.000000 75% 9.040000 1.000000 max 9.920000 1.000000

Размер обучающей выборки: (400, 7)

Размер тестовой выборки: (100, 7)

==================================================

МОДЕЛЬ 1: ДЕРЕВО РЕШЕНИЙ

================================================== Метрики качества дерева решений:

Обучающая выборка - MSE: 0.001325, MAE: 0.026323, R²: 0.932784

Тестовая выборка - MSE: 0.007606, MAE: 0.063702, R²: 0.628090

==================================================

МОДЕЛЬ 2: СЛУЧАЙНЫЙ ЛЕС

================================================== Метрики качества случайного леса:

Обучающая выборка - MSE: 0.001247, MAE: 0.024212, R²: 0.936706

Тестовая выборка - MSE: 0.004130, MAE: 0.042913, R²: 0.798032

==================================================

СРАВНЕНИЕ МОДЕЛЕЙ

==================================================

Модель MSE (тест) MAE (тест) R² (тест) Переобучение (R²)

Дерево решений 0.007606 0.063702 0.628090 0.304695

Случайный лес 0.004130 0.042913 0.798032 0.138674

==============================

ВАЖНОСТЬ ПРИЗНАКОВ

==============================

Важность признаков (Дерево решений): Признак Важность CGPA 0.8463 GRE Score 0.0776 LOR 0.0236 Research 0.0167

TOEFL Score 0.0153

University Rating 0.0102

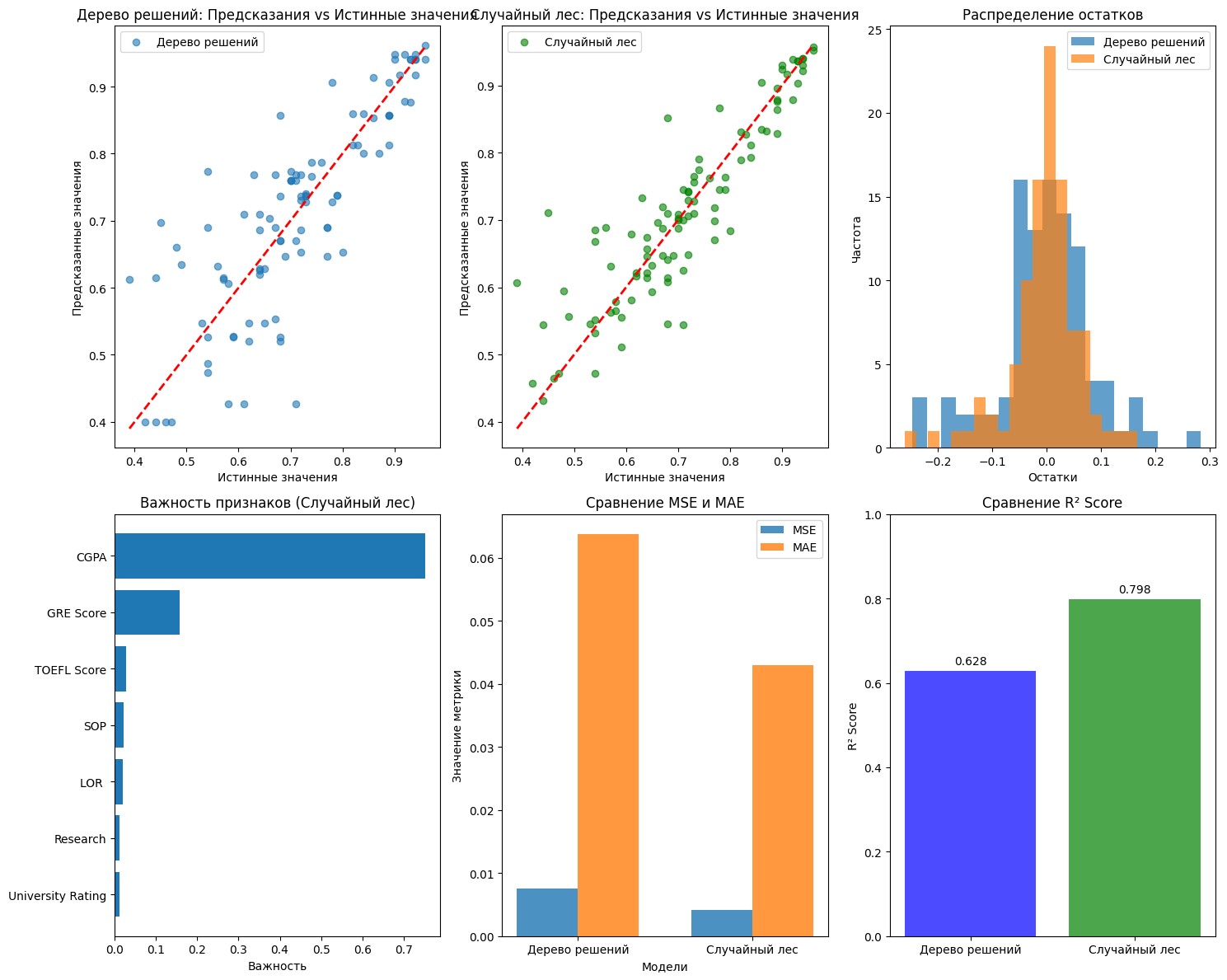
SOP 0.0102

Важность признаков (Случайный лес): Признак Важность CGPA 0.7513 GRE Score 0.1572

TOEFL Score 0.0279 SOP 0.0209

LOR 0.0205

Research 0.0114 University Rating 0.0108



============================================================

ВЫВОДЫ И АНАЛИЗ РЕЗУЛЬТАТОВ

============================================================

1. ВЫБОР МЕТРИК КАЧЕСТВА:
   * MSE (Mean Squared Error) - штрафует большие ошибки сильнее
   * MAE (Mean Absolute Error) - более устойчива к выбросам
   * R² (коэффициент детерминации) - показывает долю объясненной дисперсии
2. КАЧЕСТВО МОДЕЛЕЙ: Дерево решений:
   * MSE: 0.007606
   * MAE: 0.063702
   * R²: 0.628090
   * Переобучение: 0.304695

Случайный лес:

* + MSE: 0.004130
  + MAE: 0.042913
  + R²: 0.798032 • Переобучение: 0.138674

1. СРАВНЕНИЕ И ВЫВОДЫ:
   * Случайный лес показывает лучше качество на тестовых данных
   * Случайный лес менее склонен к переобучению
2. ВАЖНОСТЬ ПРИЗНАКОВ:

Наиболее важные признаки согласно случайному лесу:

* + CGPA: 0.7513
  + GRE Score: 0.1572
  + TOEFL Score: 0.0279

1. РЕКОМЕНДАЦИИ:
   * Для данной задачи рекомендуется использовать случайный лес
   * Случайный лес обеспечивает лучшую обобщающую способность
   * Средняя абсолютная ошибка составляет ~0.043
   * Модель объясняет ~79.8% дисперсии целевой переменной
2. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ:
   * Диапазон целевой переменной: [0.340, 0.970]
   * Среднее значение: 0.722
   * Стандартное отклонение: 0.141
   * Baseline MSE (среднее): 0.020598
   * Улучшение относительно baseline: 79.9%