УДК 004.896

# Анализ эффективности квотербеков в Национальной футбольной лиги на основе данных с применением регрессионных моделей и важности признаков

*Акентьев Данила Денисович*
*Приамурский государственный университет имени Шолом-Алейхема*
*Студент*

**Аннотация**
Цель данной статьи заключается в проведении глубокого анализа эффективности выступления квотербеков в Национальной футбольной лиги с использованием методов анализа данных и машинного обучения. Использованы различные библиотеки Python: pandas для работы с данными и их анализа. numpy для математических вычислений. matplotlib и seaborn для визуализации данных. scikit-learn для построения и оценки моделей машинного обучения. Используется среда разработки Google Collab. В данной работе провели глубокий анализ, эффективности квотербеков в Национальной футбольной лиги с использованием методов анализа данных и машинного обучения.
**Ключевые слова:** Линейная Регрессия, Дерево Решений, Случайный Лес, k-Ближайшие Соседи, Национальная футбольная лига, Квотербеки, Регрессионные модели, Анализ данных, Машинное обучение в спорте, Важность признаков, Производительность квотербеков, Спортивная аналитика, Взаимосвязи в футбольной статистике, Эффективность передачи в Национальной футбольной лиги, Метрики производительности.

# Analysis of the effectiveness of quarterbacks in the National Football League based on data using regression models and the importance of features

*Akentev Danila Denisovich*
*Sholom-Aleichem Priamursky State University*
*Student*

**Abstract**
The purpose of this article is to conduct an in-depth analysis of the performance of quarterbacks in the National Football League using data analysis and machine learning methods. Various Python: pandas libraries are used to work with data and analyze it. numpy for mathematical calculations. matplotlib and seaborn for data visualization. scikit-learn for building and evaluating machine learning models. The Google Collab development environment is used. In this paper, we conducted an in-depth analysis of the effectiveness of quarterbacks in the National Football League using data analysis and machine learning methods.

**Keywords**: Linear Regression, Decision Tree, Random Forest, k-Nearest Neighbors, National Football League (NFL), Quarterbacks, Regression Models, Data analysis, Machine Learning in sports, Importance of attributes, Quarterback Performance, Sports Analytics, Relationships in football statistics, Transfer Efficiency in the NFL, Performance Metrics.

## 1 Введение

### 1.1 Актуальность

Тема "Анализ эффективности квотербеков в Национальной футбольной лиги на основе данных с применением регрессионных моделей и важности признаков" является актуальной по нескольким причинам:

1. Повышенный интерес к спортивной аналитике: Спортивная аналитика становится все более важной в профессиональных спортивных лигах. Команды и тренеры активно используют данные для принятия стратегических решений, и анализ эффективности квотербеков в Национальной футбольной лиги представляет собой ключевой аспект этой тенденции.

2. Развитие машинного обучения в спорте: Применение регрессионных моделей и методов машинного обучения в спорте становится все более распространенным. Анализ эффективности квотербеков предоставляет возможность показать, как эти методы могут быть успешно применены для прогнозирования и оптимизации результатов в футболе.

3. Заинтересованность болельщиков и профессионалов: Тема интересна не только специалистам в области аналитики и спортивных наук, но и широкой аудитории болельщиков Национальной футбольной лиги. Анализ производительности квотербеков может привлечь внимание исследователей, тренеров, болельщиков и журналистов.

4. Возможные практические применения: Результаты анализа могут быть использованы для повышения эффективности тренировок, принятия решений о составе команды и улучшения стратегий игры, что делает эту тему актуальной для профессиональных футбольных организаций.

Таким образом, данная тема сочетает в себе активный интерес к аналитике в спорте, применение современных методов машинного обучения и практическую значимость для футбольных команд и тренеров.

### 1.2 Обзор исследований

Р.Г. Галимов рассмотрел основы алгоритмов машинного обучения такие как линейная регрессия и дерево решений [1]. Собрали и проанализировали данные используя язык программирования python для выявления и уменьшения неравенства в обществе А. Ж. Серикпай, К. Давидов, О. Абдираманов [2]. Рассмотрели возможности применения машинного обучения для определения наиболее важных игровых атрибутов, определяющих рейтинги игроков после и по ходу матча, в соответствии с их более детальными позициями на поле В.И. Кияев и А.М. Макаров [3]. С.В.

Корнев проанализировал период игр национальной футбольной лиги с 1995-2016 г. [4].

### 1.3 Цель исследования

Цель данной статьи заключается в проведении глубокого анализа эффективности выступления квотербеков в Национальной футбольной лиги с использованием методов анализа данных и машинного обучения.

## 2 Материалы и методы

Для выполнения работы использован csv-файл "nflpass.csv" [5], содержащий статистические данные о квотербеках в Национальной футбольной лиги. Использованы различные библиотеки Python: pandas для работы с данными и их анализа. numpy для математических вычислений. matplotlib и seaborn для визуализации данных. scikit-learn для построения и оценки моделей машинного обучения. Используется среда разработки Google Collab.

## 3 Результаты и обсуждения

Выполнение анализа происходит на сайте Google Collab. Перед началом работы требуется установить и импортировать все библиотеки, которые потребуются в дальнейшем.

```
# Импорт библиотек
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.neighbors import KNeighborsRegressor
```

Считываем csv-файл, вывод первых 5 строк для визуального просмотра дата сета, предоставление информации о данном файле.

```
# Чтения данных из файла CSV
data = pd.read_csv("nflpass.csv")
# Выводит первые 5 строк
print(data.head())
# Статистическое описание числовых данных в DataFrame
print(data.describe())

Passing_Attempts  Passing_Completions  Passing_Yards  \
0              2429                 1546          19869
1              5391                 3409          40551
2              6049                 3604          45173
```

```
3              3942                  2397          29527
4              2958                  1685          22700

   Touchdowns_by_Passing  Interceptions  Rating           Name
0                    140             68    96.8     Steve_Young
1                    273            139    92.3     Joe_Montana
2                    328            185    88.2      Dan_Marino
3                    201            143    85.8       Jim_Kelly
4                    153            109    83.4  Roger_Staubach
       Passing_Attempts  Passing_Completions  Passing_Yards  \
count         26.000000            26.000000      26.000000
mean        3622.076923          2122.730769   26901.615385
std         1357.835275           803.607444   10185.972959
min         1505.000000           864.000000   10412.000000
25%         2707.000000          1580.750000   22037.250000
50%         3421.500000          2003.000000   24906.000000
75%         4365.250000          2531.500000   32196.500000
max         6467.000000          3686.000000   47003.000000

       Touchdowns_by_Passing  Interceptions     Rating
count              26.000000      26.000000  26.000000
mean              176.230769     130.846154  82.916923
std                76.668276      57.517088   4.117484
min                54.000000      38.000000  79.000000
25%               130.000000      84.000000  80.372500
50%               159.500000     136.500000  81.800000
75%               226.750000     164.500000  82.925000
max               342.000000     266.000000  96.800000
```

Для начала используется код для построения матрицы диаграмм рассеяния для всех числовых переменных DataFrame (рис.1). Также строим гистограмму переменной ‘Rating’ (рис.2).

```
# Построение матрицы диаграмм рассеяния для всех числовых переменных в
DataFrame
sns.pairplot(data)
plt.show()

# Построение гистограммы распределения переменной 'Rating'
plt.figure(figsize=(6, 4))
sns.histplot(data['Rating'], bins=15, kde=True)
plt.title('Распределение рейтингов')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```

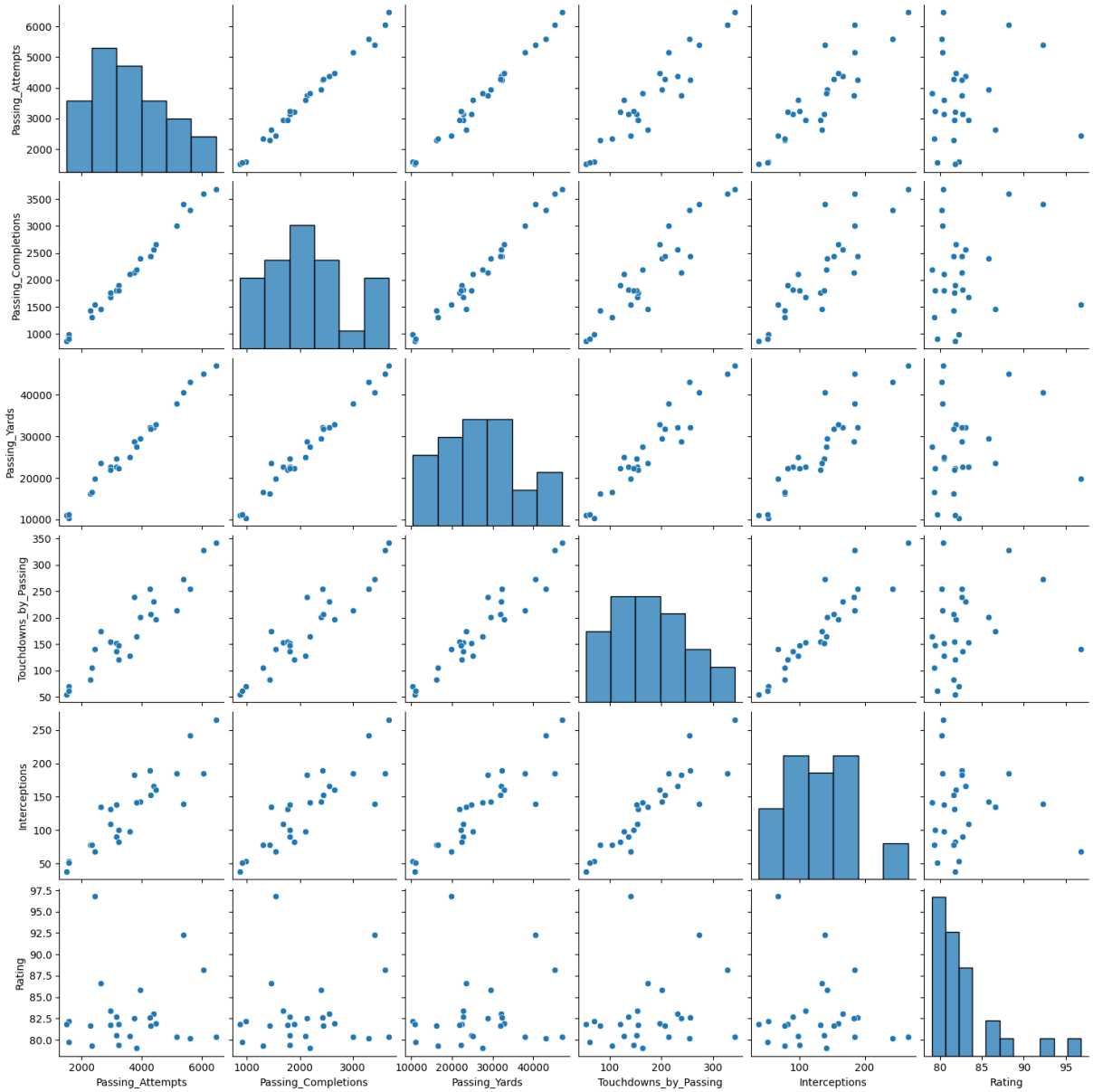


Рисунок 1 – Матрицы диаграмм рассеяния

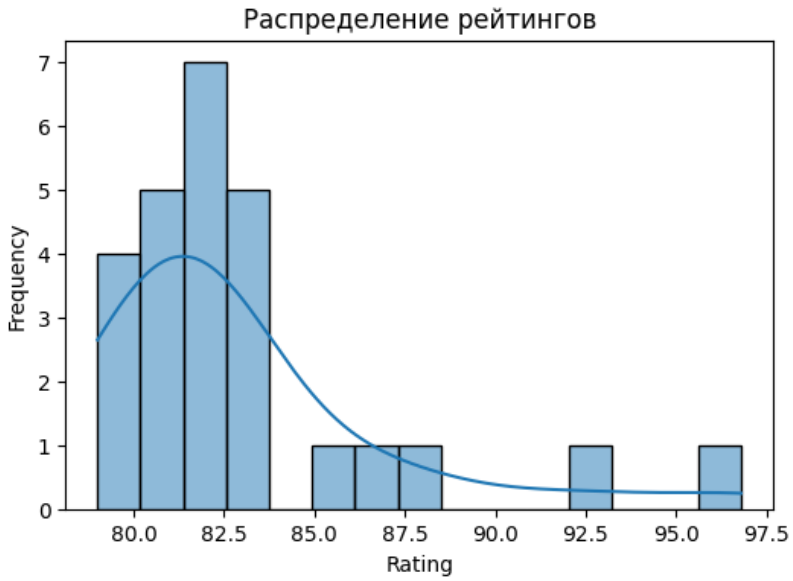


Рисунок 2 – Гистограмма распределения переменной ‘Rating’

Далее оценка взаимосвязи между ключевыми параметрами и рейтингом квотербеков, а также анализ мультиколлинеарности для выявления возможных проблем в моделировании. Вычисление и визуализация корреляции: Первоначально была построена матрица корреляции (рис.3), выявляющая степень взаимосвязи между параметрами, такими как количество попыток передачи, успешные передачи, ярды, количество Touchdowns и количество перехватов, а также рейтингом квотербеков. Эта матрица была визуализирована с использованием тепловой карты для лучшего понимания силы и направления корреляций. Оценка степени корреляции: Степень влияния каждого параметра на рейтинг была оценена с использованием коэффициентов корреляции Пирсона. Это позволяет определить, какие параметры имеют наибольшее влияние на рейтинг квотербеков. Исследование мультиколлинеарности: Для оценки мультиколлинеарности был использован метод Variance Inflation Factor (VIF). Этот метод позволяет выявить проблемы, связанные с высокой корреляцией между предикторами, что может повлиять на точность модели. Исследование включает в себя расчет VIF для каждого признака и анализ их значений. Результаты: Анализ корреляции и мультиколлинеарности предоставляет важную информацию для дальнейшего моделирования производительности квотербеков в NFL. Он выявляет ключевые параметры, которые могут оказаться наиболее значимыми при прогнозировании рейтинга и предупреждает о возможных проблемах мультиколлинеарности, которые могут потребовать дополнительной обработки данных.

```
# Вычисление коэффициентов корреляции
correlation_matrix = data[['Passing_Attempts', 'Passing_Completions',
'Passing_Yards', 'Touchdowns_by_Passing', 'Interceptions',
'Rating']].corr()

# Визуализация матрицы корреляции
plt.figure(figsize=(8, 6))
```

```
sns.heatmap(correlation_matrix, cmap='coolwarm', annot=True, fmt=".2f",
linewidths=.5)
plt.title('Матрица корреляции')
plt.show()

# Оценка степени корреляции
pearson_corr =
correlation_matrix['Rating'].abs().sort_values(ascending=False)
print("Коэффициенты корреляции с Rating:")
print(pearson_corr)

# Исследование мультиколлинеарности с использованием VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Выделение признаков
X = data[['Passing_Attempts', 'Passing_Completions', 'Passing_Yards',
'Touchdowns_by_Passing', 'Interceptions']]
X['Intercept'] = 1  # Добавление Intercept для расчета VIF

# Вычисление VIF для каждого признака
vif_data = pd.DataFrame()
vif_data["Feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in
range(X.shape[1])]

print("VIF для каждого признака:")
print(vif_data)
```

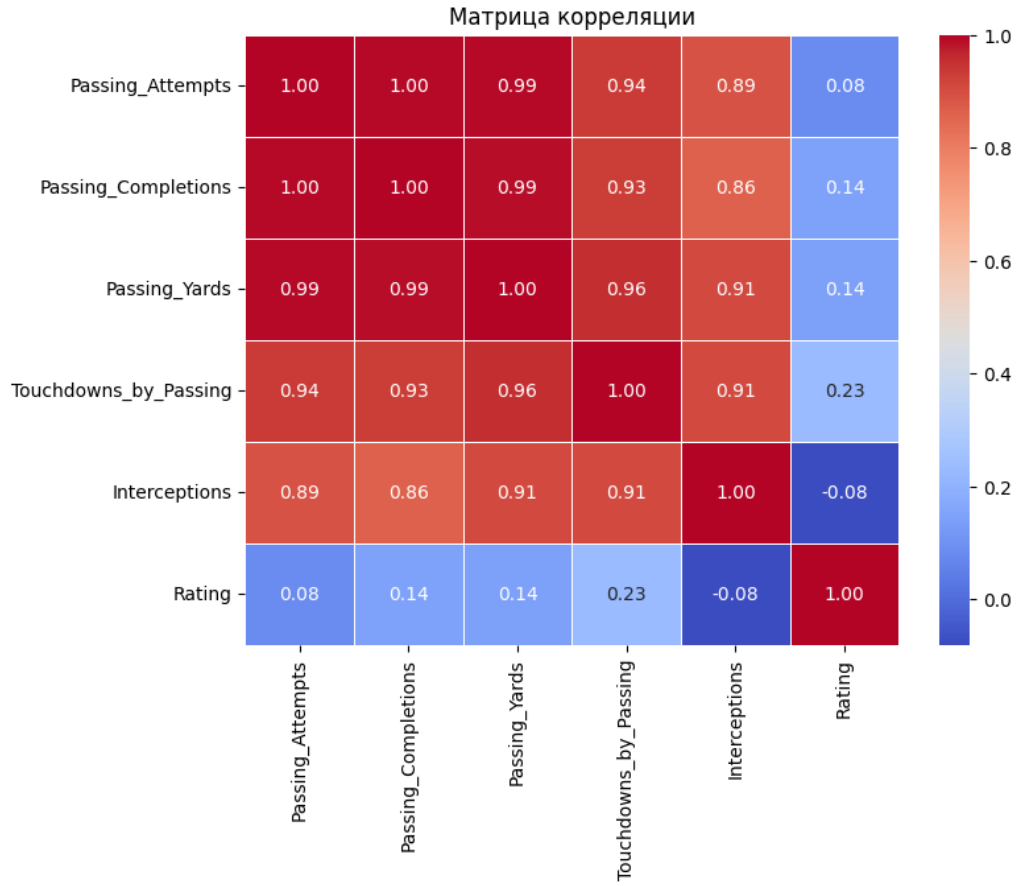


Рисунок 3 – Матрица корреляции с высокой мультиколлинеарностью

```
Коэффициенты корреляции с Rating:
Rating                    1.000000
Touchdowns_by_Passing     0.229714
Passing_Completions       0.144324
Passing_Yards             0.141585
Interceptions             0.080961
Passing_Attempts          0.080405
Name: Rating, dtype: float64
VIF для каждого признака:
                  Feature         VIF
0       Passing_Attempts  198.730008
1    Passing_Completions  183.727910
2          Passing_Yards  126.632780
3  Touchdowns_by_Passing   13.834347
4          Interceptions   11.100553
5              Intercept    9.020317
```

Анализ мультиколлинеарности с использованием метода Variance Inflation Factor (VIF). Результат выявили высокий уровень мультиколлинеарности среди параметров Passing_Attempts, Passing_Completions и Passing_Yards, что может негативно сказаться на стабильности модели. Для улучшения модели и предотвращения проблем, связанных с мультиколлинеарностью, удалим одного или нескольких параметров с высокими значениями VIF. Это может улучшить стабильность и интерпретируемость модели. Выводим матрицу корреляции (рис.4).

```
# Удаление признаков с высокой мультиколлинеарностью
data_filtered = data[['Passing_Yards', 'Touchdowns_by_Passing',
'Interceptions', 'Rating']]

# Вычисление коэффициентов корреляции для нового набора данных
correlation_matrix_filtered = data_filtered.corr()

# Визуализация матрицы корреляции
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix_filtered, cmap='coolwarm', annot=True,
fmt=".2f", linewidths=.5)
plt.title('Матрица корреляции (без высокой мультиколлинеарности)')
plt.show()

# Оценка степени корреляции
pearson_corr_filtered =
correlation_matrix_filtered['Rating'].abs().sort_values(ascending=False)
print("Коэффициенты корреляции с Rating после удаления признаков с высокой
мультиколлинеарностью:")
print(pearson_corr_filtered)

# Исследование мультиколлинеарности с использованием VIF для нового набора
данных
X_filtered = data_filtered[['Passing_Yards', 'Touchdowns_by_Passing',
'Interceptions']]
X_filtered['Intercept'] = 1

vif_data_filtered = pd.DataFrame()
vif_data_filtered["Feature"] = X_filtered.columns
vif_data_filtered["VIF"] = [variance_inflation_factor(X_filtered.values,
i) for i in range(X_filtered.shape[1])]
```

```
print("VIF для каждого признака после удаления признаков с высокой
мультиколлинеарностью:")
print(vif_data_filtered)
```

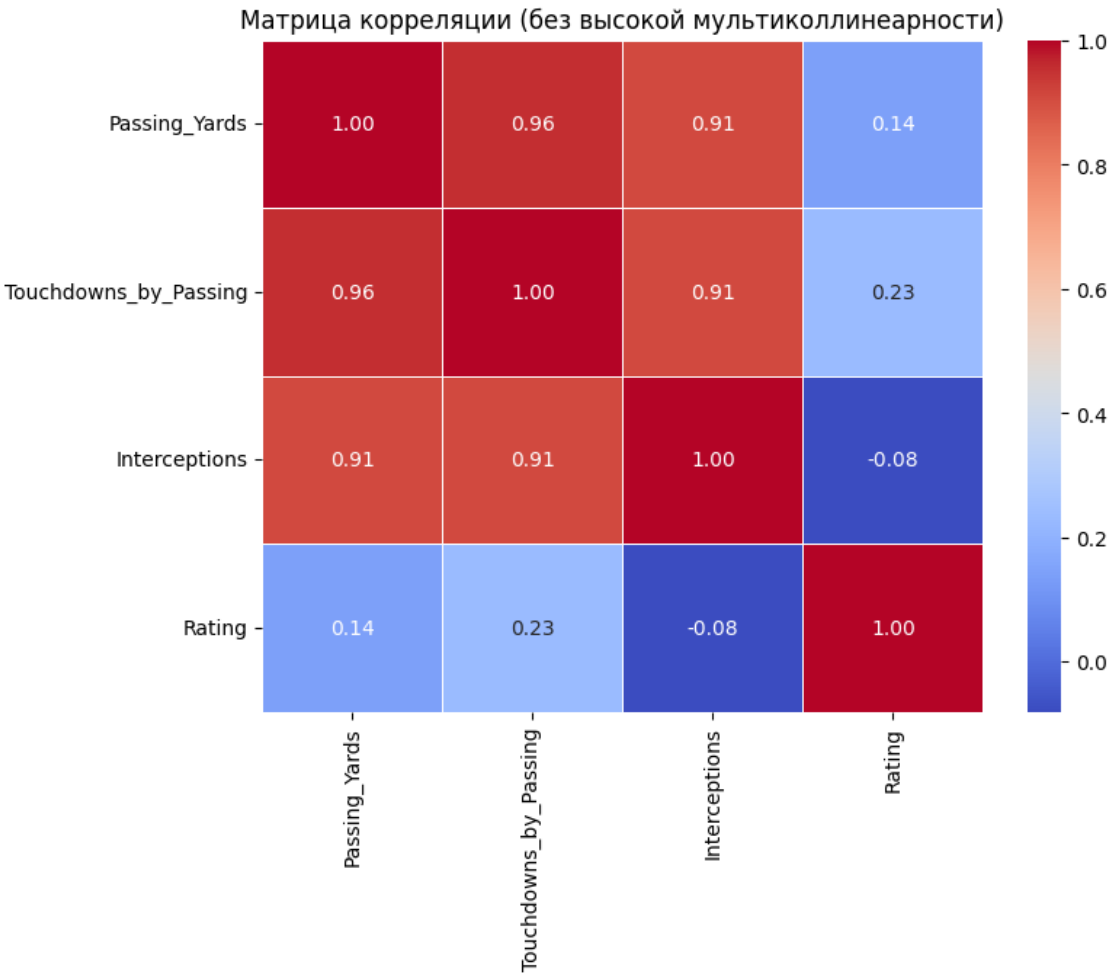


Рисунок 4 – Матрица корреляции без высокой мультиколлинеарности

```
Коэффициенты корреляции с Rating после удаления признаков с высокой
мультиколлинеарностью:
Rating                    1.000000
Touchdowns_by_Passing     0.229714
Passing_Yards             0.141585
Interceptions             0.080961
Name: Rating, dtype: float64
VIF для каждого признака после удаления признаков с высокой
мультиколлинеарностью:
                 Feature        VIF
0          Passing_Yards  12.885165
1  Touchdowns_by_Passing  12.902699
2          Interceptions   6.559155
3              Intercept   8.860535
```

Теперь предстоит, выбор ключевых признаков, которые являются важными для предсказания рейтинга. Эти признаки включают Passing_Yards, Touchdowns_by_Passing и Interceptions. Выбор этих параметров обусловлен

их тесной связью с производительностью квотербеков в атаке. Используем функцию train_test_split для разбиения данных на тренировочные и тестовые наборы. Этот шаг необходим для обучения модели на одном наборе данных и последующей проверки ее производительности на независимом тестовом наборе. Процесс нормализации данных с использованием стандартного скалирования (StandardScaler) обеспечивает однородность масштабов признаков. Это важно для моделей, основанных на расстояниях, так как гарантирует, что каждый признак вносит сопоставимый вклад в обучение модели.

```
# Подготовка данных для моделей
X = data[['Passing_Yards', 'Touchdowns_by_Passing', 'Interceptions']]
y = data['Rating']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Нормализация данных
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Для анализа производительности моделей машинного обучения были выбраны следующие алгоритмы: линейная регрессия, дерево решений, случайный лес, k-ближайших соседей и XGBoost. Каждая модель была обучена на тренировочных данных и оценена на тестовом наборе с использованием двух ключевых метрик: среднеквадратичной ошибки (MSE) и коэффициента детерминации (R^2).

```
model_lr = LinearRegression()
model_tree = DecisionTreeRegressor(random_state=42)
model_rf = RandomForestRegressor(n_estimators=100, random_state=42)
model_knn = KNeighborsRegressor(n_neighbors=5)
model_xgb = XGBRegressor(random_state=42)

# Обучение моделей
model_lr.fit(X_train_scaled, y_train)
y_pred_lr = model_lr.predict(X_test_scaled)
model_tree.fit(X_train_scaled, y_train)
y_pred_tree = model_tree.predict(X_test_scaled)
model_rf.fit(X_train_scaled, y_train)
y_pred_rf = model_rf.predict(X_test_scaled)
model_knn.fit(X_train_scaled, y_train)
y_pred_knn = model_knn.predict(X_test_scaled)
model_xgb.fit(X_train_scaled, y_train)
y_pred_xgb = model_xgb.predict(X_test_scaled)

# Оценка моделей
print("Линейная регрессия:")
print(f"MSE: {mean_squared_error(y_test, y_pred_lr):.4f}")
print(f"R^2: {r2_score(y_test, y_pred_lr):.4f}\n")
print("Дерево решений:")
print(f"MSE: {mean_squared_error(y_test, y_pred_tree):.4f}")
print(f"R^2: {r2_score(y_test, y_pred_tree):.4f}\n")
print("Случайный лес:")
print(f"MSE: {mean_squared_error(y_test, y_pred_rf):.4f}")
print(f"R^2: {r2_score(y_test, y_pred_rf):.4f}\n")
```

```
print("k-Ближайшие соседи:")
print(f"MSE: {mean_squared_error(y_test, y_pred_knn):.4f}")
print(f"R^2: {r2_score(y_test, y_pred_knn):.4f}\n")
print("XGBoost:")
print(f"MSE: {mean_squared_error(y_test, y_pred_xgb):.4f}")
print(f"R^2: {r2_score(y_test, y_pred_xgb):.4f}\n")
```

Для оценки важности каждого признака в линейной регрессии были рассчитаны коэффициенты влияния (веса). Результаты представлены в следующем порядке:

Визуализация важности признаков: Построена круговая диаграмма, отражающая важность каждого признака в модели линейной регрессии (рис.5). Цветовая гамма выбрана для наглядности распределения влияний.

Таблица коэффициентов линейной регрессии: Представлены значения коэффициентов каждого признака с указанием абсолютной величины. Также приведены в порядке убывания важности.

Эти результаты могут помочь в понимании того, какие признаки оказывают более существенное влияние на предсказание рейтинга в модели линейной регрессии. Подобный анализ может быть полезен для выявления ключевых факторов, влияющих на целевую переменную в данном контексте

```
# Важность признаков для линейной регрессии
coefficients = model_lr.coef_
feature_importance_lr_df = pd.DataFrame({'Feature': X.columns,
'Coefficient': abs(coefficients)})
feature_importance_lr_df =
feature_importance_lr_df.sort_values(by='Coefficient', ascending=False)

plt.figure(figsize=(8, 8))
plt.pie(feature_importance_lr_df['Coefficient'],
labels=feature_importance_lr_df['Feature'], autopct='%1.1f%%',
startangle=90, colors=sns.color_palette('viridis'))
plt.title('Важность признаков из линейной регрессии')
plt.show()

print("Коэффициенты линейной регрессии:")
for index, row in feature_importance_lr_df.iterrows():
    print(f"{row['Feature']}: {row['Coefficient']:.4f}")
```

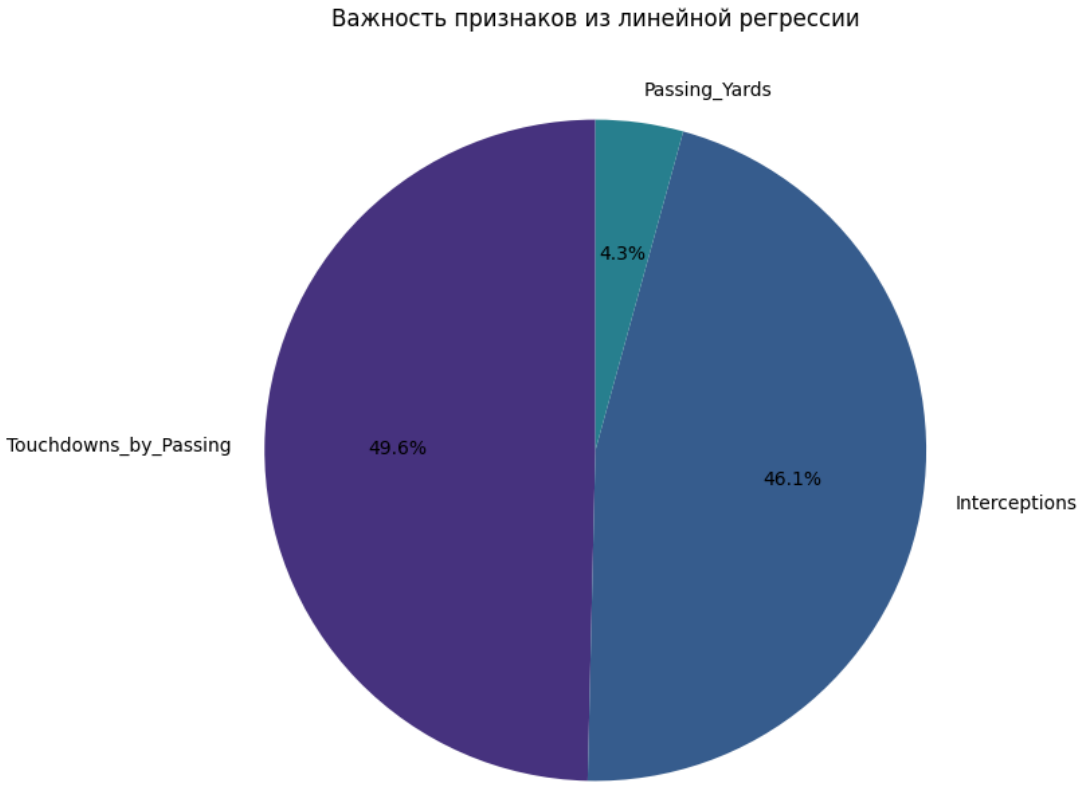


Рисунок 5 – Важность признаков из линейной регрессии

```
Коэффициенты линейной регрессии:
Touchdowns_by_Passing: 5.6788
Interceptions: 5.2745
Passing_Yards: 0.4893
```

В проведенном исследовании коэффициентов линейной регрессии на основе модели предсказания рейтинга в NFL, выявлены следующие ключевые факторы, оказывающие значительное влияние на предсказываемую переменную:

Touchdowns_by_Passing: Коэффициент: 5.6788. Видимо, количество Touchdowns, сделанных при помощи передачи, имеет существенное положительное влияние на рейтинг игрока. Это может указывать на важность эффективности атаки команды и успешного завершения передач.

Interceptions: Коэффициент: 5.2745. Присутствие данного признака в модели говорит о том, что количество перехватов также сильно влияет на рейтинг. Вероятно, защитные действия, предотвращающие успешные передачи соперника, имеют важное значение для успешной игры и, следовательно, для рейтинга.

Passing_Yards: Коэффициент: 0.4893. Хотя коэффициент этого признака ниже, чем у двух предыдущих, он все равно оказывает влияние на рейтинг. Вероятно, количество ярдов, пройденных при успешных передачах, также имеет своеобразное воздействие на оценку производительности игрока.

Такие выводы могут быть полезны для тренеров, аналитиков и фанатов в понимании того, какие аспекты игры игрока существенны для формирования высокого рейтинга в Национальной футбольной лиги. Подобный анализ коэффициентов линейной регрессии способствует более глубокому пониманию влияния различных факторов на успех игрока в данном виде спорта

**4 Выводы**

В данной работе провели глубокий анализ, эффективности квотербеков в Национальной футбольной лиги с использованием методов анализа данных и машинного обучения.

**Библиографический список**

1. Галимов Р.Г. Основы алгоритмов машинного обучения - обучение с учителем // Аллея науки. 2017. Т. 1. №. 14. С. 810-817
2. Серікпай Ә.Ж., Давидов К., Абдираманов О. Анализ данных в области спорта и применение машинного обучения // Central asian scientific journal. 2024. Т. 1. №. 20. С. 127-129.
3. Кияев В.И., Макаров А.М. Применение новых методов спортивной аналитики в построении прогнозных моделей в игровых видах спорта // Гипотеза. 2020. Т. 1. №. 10. С. 45-51.
4. Корнев С.В. Аналитика провального периода для квотербеков в национальной футбольной лиги // E-SCIO. 2020. Т. 5. №. 44. С. 553-558.
5. URL: http://lib.stat.cmu.edu/datasets/nflpass