Всероссийский конкурс исследовательских и проектных работ школьников «Высший пилотаж»

Сравнение статистических алгоритмов в задаче предсказания среднего балла школьника на основе его успеваемости.

Исследовательская работа

Направление «*Computer Science*»

Автор: Ильяшенко Корней Романович

Учащийся 11 класса, ОАНО «Новая Школа», Москва

2024 г.

Содержание

Аннотация

В исследовании анализируются статистические алгоритмы для предсказания средних оценок учеников: линейная регрессия, ARIMA (с автоматическим подбором параметров и без) и Prophet. Для оценки моделей использованы метрики MAE и время вычислений.

1 Введение

Практически в каждой школе основной метрикой успешности ученика является его средняя оценка за определенный период, например, за полугодие, год или триместр. Оценка, в свою очередь, является определенным числовым, либо категориальным значением. К каждой оценке могут относиться и прочие характеристики, такие как тип работы, вес оценки (коэффициент оценки при подсчете среднего взвешенного), дата проведения работы/выставления оценки и прочие. В исследовании рассмотрены основные метрики оценивания эффективности выбранных статистических алгоритмов в задаче предсказания средних оценок учеников.

Целью исследования будет являться **сопоставление алгоритмов предсказания временных рядов с точки зрения их эффективноти**.

В конкретной задаче рассмотрен список оценок одной параллели российской школы ОАНО «Новая школа». Оценки могут иметь различный вес, значение оценок варьируется от 0 до 100.

2 Методы

2.1 Выбор моделей для сравнения

Для начала необходимо было выбрать несколько моделей, используемых в задаче предсказания временных рядов. Из многочисленных вариантов было принято решение выбрать три модели:

1. **Linear Regression**
2. **ARIMA** (with and without automatic parameter selection) [1]
3. **Prophet** by facebook [2]

В списке модели приведены по мере увеличения их сложности (по кол-ву учитываемых характеристик временного ряда)

Были выбраны самые распространенные и не требующие высокого уровня погружения для освоения принципа работы модели. [3]

2.2 Выбор метрики оценивания эффективности моделей

Были выбраны две основные метрики для оценки эффективности модели

1. **Суммарное Время** обучения модели и генерации предсказаний на основе тренировочной выборки в *секундах*.
2. **Mean Absolute Error** - ошибка модели, вычисляемая как модуль разности действительной средней оценки ученика и предсказанной средней оценки. Эта метрика выбрана по причине ее хорошей интерпретируемости (средняя ошибка модели в баллах)

Энергозатратность не оценивалась по причине ее высокой корелляции с временем выполнения задачи (см. п. 1). Вся обработка данных и тестирование моделей были запущены на одном и том же устройстве *VivoBook ASUS M1603* последовательно при отсутствии прочей нагрузки ($< 5\%$ CPU; $< 10\%$ RAM).

2.3 Выбор используемого окружения для предварительного анализа, использования моделей и их оценивания.

Полный список использованных в решении задачи инструментов:

- Язык программирования *Python*
- Библиотека для работы с данными *Pandas*
- Библиотеки для построения визуализаций *Matplotlib* и *Seaborn*
- Библиотека математических инструментов *NumPy*
- Библиотека научных инструменов *scipy* (в задаче подсчета корреляции)
- Библиотека для использования Prophet *prophet*
- Редактор кода *Visual Studio Code*
- Операционная система *Arch Linux x86_6*

Все библиотеки находятся в открытом доступе и могут быть установлены через большинство пакетных менеджеров Python. *Подробнее*

Остальные библиотеки

2.4 Предварительный анализ данных.

2.4.1 Описание изначальных данных.

Изначально в распоряжении был архив файлов формата `.xls`. В названии каждой таблицы содержалась следующая информация:

1. **Учебный год** (например, 2022-2023)
2. **Класс**, для учеников которого указаны оценки (например, 5-2)
3. **Предмет**, за который указаны оценки (например, биология)
4. **Номер триместра**, за который выставлены оценки.

Все таблицы были загружены и конвертированы в удобный для анализа тип данных `pandas.DataFrame`, где каждая строка - информация об оценке.

Информация о полученной таблице:

```
<class 'pandas.core.frame.DataFrame'>
Index: 10537 entries, 0 to 10587
Data columns (total 7 columns):
 #   Column        Count  Dtype
---  ------        --------------
 0   student_id    10537  object
 1   mark          10537  int64
 2   grade         10537  int64
 3   subject       10537  object
 4   trimester     10537  int64
 5   marks_weight  10537  object
 6   mark_date     10537  datetime64[ns]
dtypes: datetime64[ns](1), int64(3), object(3)
memory usage: 658.6+ KB
```

1. `student_id` - уникальный идентификатор ученика (например, Student_4430)

2. `mark` - оценка ученика от 0 до 100 (по правилам школы 0 не учитывается при подсчете средней оценки)
3. `grade` - класс обучения от 5 до 7 ученика
4. `subject` - предмет, за который получена оценка (например, `math`).
5. `trimester` - номер триместра, в котором выставлена оценка (например, `2`)
6. `marks_weight` - вес оценки (в проставлении средней оценки напрямую учитываются только катекории «К2», «»)

Всего в данных представлены оценки за полные 2022-2023, 2023-2024 года и данные первого триместра 2024 года.

Всего в таблице **10537** оценок.

2.4.2 Общее исследование данных

В Рис. 1 приведен общий предварительеный анализ по основным столбцам.

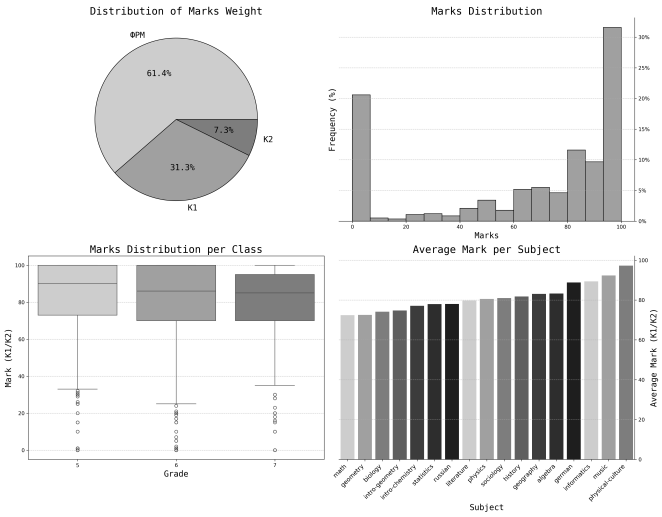

Рис. 1. Предватиельный анализ данных

Видно, что большинство оценок имеют формирующий вес, но по причине того, что они не влияют на итоговую оценку школьника, учитывать в анализе мы их не будем.

Более того, за большинство формирующих оценок равны очень маленькому значению (механизм: они не влияют на итоговую оценку, поэтому можно получать низкую оценку). Это видно из Рис. 2.

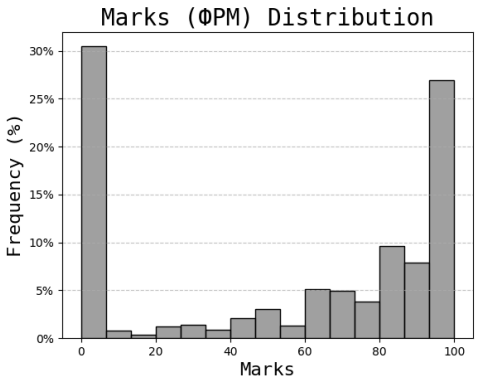

Рис. 2. Кол-во формирующих оценок с определенным баллом

2.4.3 Анализ дисперсии оценок

Для каждого ученика посчитаем его средний балл за весь период обучения. Отберем три ученика в соответствии с их уровнем (эмпирическая оценка):

1. `High level` ≈ 89 баллов (5 в пятибалльной системе)
2. `Medium level` ≈ 71 балл (4 в пятибалльной системе)
3. `Low Level` ≈ 60 баллов (3 в пятибалльной системе)

Также разделим оценки учеников на два учебных года для наглядности.

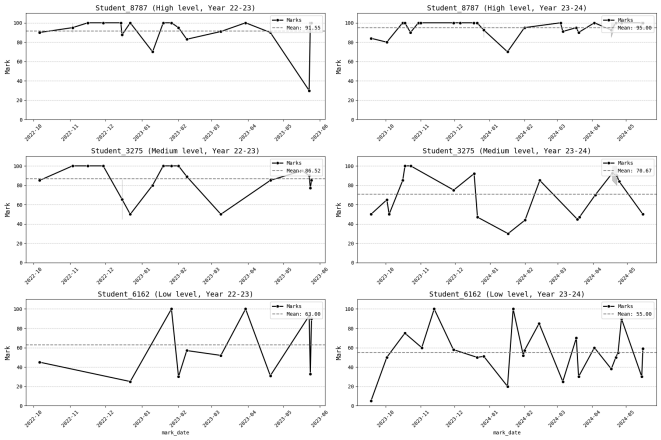

Рис. 3. Временной ряд оценок школьников разной успеваемости.

Из Рис. 3 видно, что у учеников меньшей успеваемости дисперсия оценок выше, чем у учеников высокой успеваемости. Проверим это на всем датасете.

Для первых 16 учеников, отсортированных по убыванию по количеству оценок, средней оценке по каждому из следующих предметов: `russian`, `math`, `biology` посчитаем их средние оценки и десперсию их оценок.

Построим соотвествующих график

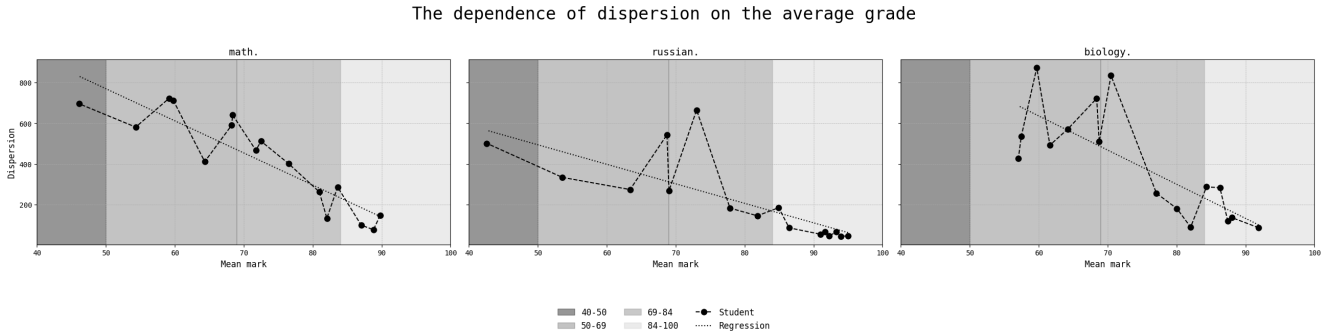

Рис. 4. Дисперсии оценок учеников разной успеваемости

В Рис. 4 по оси y указаны дисперсии оценок учеников, а по оси x - средний балл каждого ученика. Видно, что по мере увеличения среднего балла школьника разброс его оценок становится меньше $\Rightarrow$ становится близким к постоянному на определенном значении.

Подтвердим гипотезу наличия связи между средним баллом школьника и дисперсией его оценок, обратившись к **корреляционному тесту Пирсона**. [4]

Построим по каждому графику корелляцию и проверим ее статистическую значимость:

Subject	Correlation Coefficient	P-Value
math	-0.896996	2.5e-06
russian	-0.759978	0.00063
biology	-0.769346	0.00049

Таблица 1. Анализ дисперсий оценок

Из Таблица 1 видно, что при выбранном граничном значении p-value $= 0.05$ полученные значения корелляции являются статистически значимыми, более того, все они ниже -0.7, что говорит о **высокой отрицательной корелляции среднего балла учеников и дисперсии их оценок**. Данное явление можно экстраполировать и на оставшиеся предметы и на оставшихся учеников из-за идентичной образовательной системы и системы получения оценок учениками.

В случае, если дисперсия данных высокая и в данных отсутствует влияние сезонности, праздников, тренда, то предсказание средней оценки с помощью выбранных алгоритмов будет бессмысленным из-за высокой ошибки и случайности данных.

2.4.4 Проверка данных на наличие тренда

Возьмем первых 10 студентов по кол-ву оценок по предметам (всего студентов $\approx 30 \Rightarrow$ выборка репрезентативна)

Для поиска тренда используем библиотеку `statmodels` и метод `Ordinary Least Squares` для построения линейной регрессии на данных. Если будет найдена линейная регрессия с коэффициентом при зависимой переменной (временем) неблизким к 0, высокой корелляцией и низким p-value < 0.05, будем считать, что в данных присутствуетн тренд, который может объяснить высокую дисперсию оценок учеников с низким средним баллом (механизм: родители заставляют ученика трудиться больше), при этом высокая дисперсия не будет означать бесполезность выбранных моделей для таких учеников.

Student ID	Subject	P-Value	Trend Absent
1047	math	0.27	True
8499	math	0.063	True
8787	math	0.46	True
8008	russian	0.0017	False
7905	russian	0.93	True
1093	russian	0.98	True
6300	literature	0.95	True
3435	history	0.77	True
2940	history	0.0001	False
3275	intro-geometry	0.53	True

Таблица 2. Анализ трендов

Из Таблица 2 видно, что в некоторых случаях наличие тренда подтверждается (`not Trend Absent`)

Таким образом, мы не можем утверждать, что каждая из выбранных моделей будет гарантированно ошибаться на оценках учеников с низкой успеваемостью.

2.4.5 Проверка данных на наличие периода

Проведем те же манипуляции с данными, что и в предыдущем праграфе.

Далее воспользуемся методом `Lomb-Scargle` библиотеки `astropy`

Он позволяет найти статистически значимые периоды в данных, которые влияют на них. Используется во временных рядах.

Подтвердим наличие периода, только если он больше *14 дней* (эмпирическое значение. Если период меньше, возможный механизм такого явления сложен и маловероятен.), если сила периода больше 0.3 и p-value < 0.05

Student ID	Subject	Confirmed Seasonality
1047	math	False
8499	math	True
8787	math	False
8008	russian	False
7905	russian	False
1093	russian	False
6300	literature	False
3435	history	False
2940	history	False
3275	intro-geometry	False

Таблица 3. Результаты подтверждения сезонности

Построим периодограмму для студента с подтвержденным периодом (id 8499)

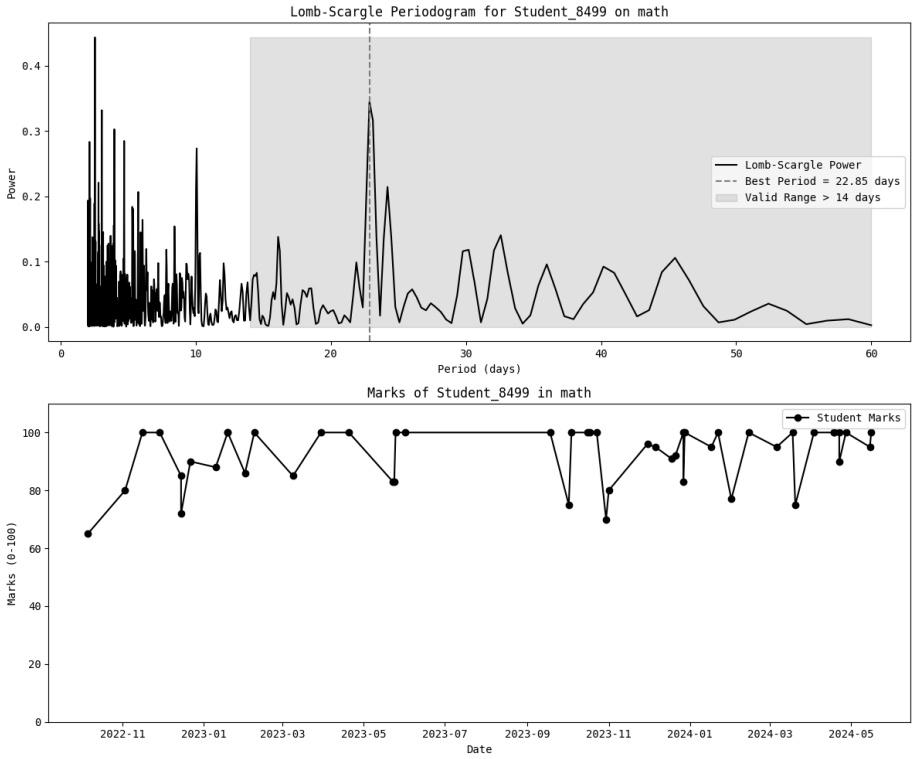

Рис. 5. Периодограмма студента с подтвержденным периодом в данных

Действительно, из Рис. 5 видно, что в данных для студента 8499 присутствует слабый период в оценках по математике. Такой период могут учесть такие сложные модели, как `Prophet` и дисперсия в данных может быть объяснена именно им.

2.4.6 Результаты предварительного анализа

- Нет смысла рассматривать прогнозирование на основе формирующих оценок или оценок, равных 0.
- Наибольший потенциал для точного прогнозирования оценок наблюдается у студентов, которые преуспевают в учебе, у тех, чьи оценки улучшаются или снижаются (указывая на тренд), а также у тех, чьи оценки демонстрируют периодичность, например, закономерности, вызванные ежемесячными тестами. В случаях периодичности использование этих повторяющихся паттернов может значительно повысить точность предсказательных моделей.
- Если в данных отсутствует периодичность или тренд, то прогнозирование становится почти бессмысленным. Например, если у студента низкие оценки, можно провести статистические тесты на наличие периодичности и тренда перед созданием модели. Если ни одного из них нет, использование дополнительных моделей для прогнозов также окажется бесполезным.
- Вариация оценок напрямую зависит от успеваемости студента.

2.5 Оценивание моделей на данных

В каждом сформированном датасете будет ровно **84** строк, каждая из которых включает ID ученика, предмет, список его оценок, соотв. ему список дат этих оценок, его общий средний балл.

2.5.1 Линейная регрессия

Построим линейную регрессию на временном ряде оценок для каждого ученика по различным предметам и для трех групп оценивания (обрезая результаты на 0 и 100):

1. Средний балл от 85 до 100 (соответсвует оценке 5 в большинстве государственных школ)
2. Средний балл от 70 до 84 (соответсвует оценке 4 в большинстве государственных школ)
3. Средний балл от 51 до 69 (соответсвует оценке 3 в большинстве государственных школ)

Учеников с более низким средним баллом мы учитывать не будем из-за их небольшого количества.

Получим следующий результат (пример)

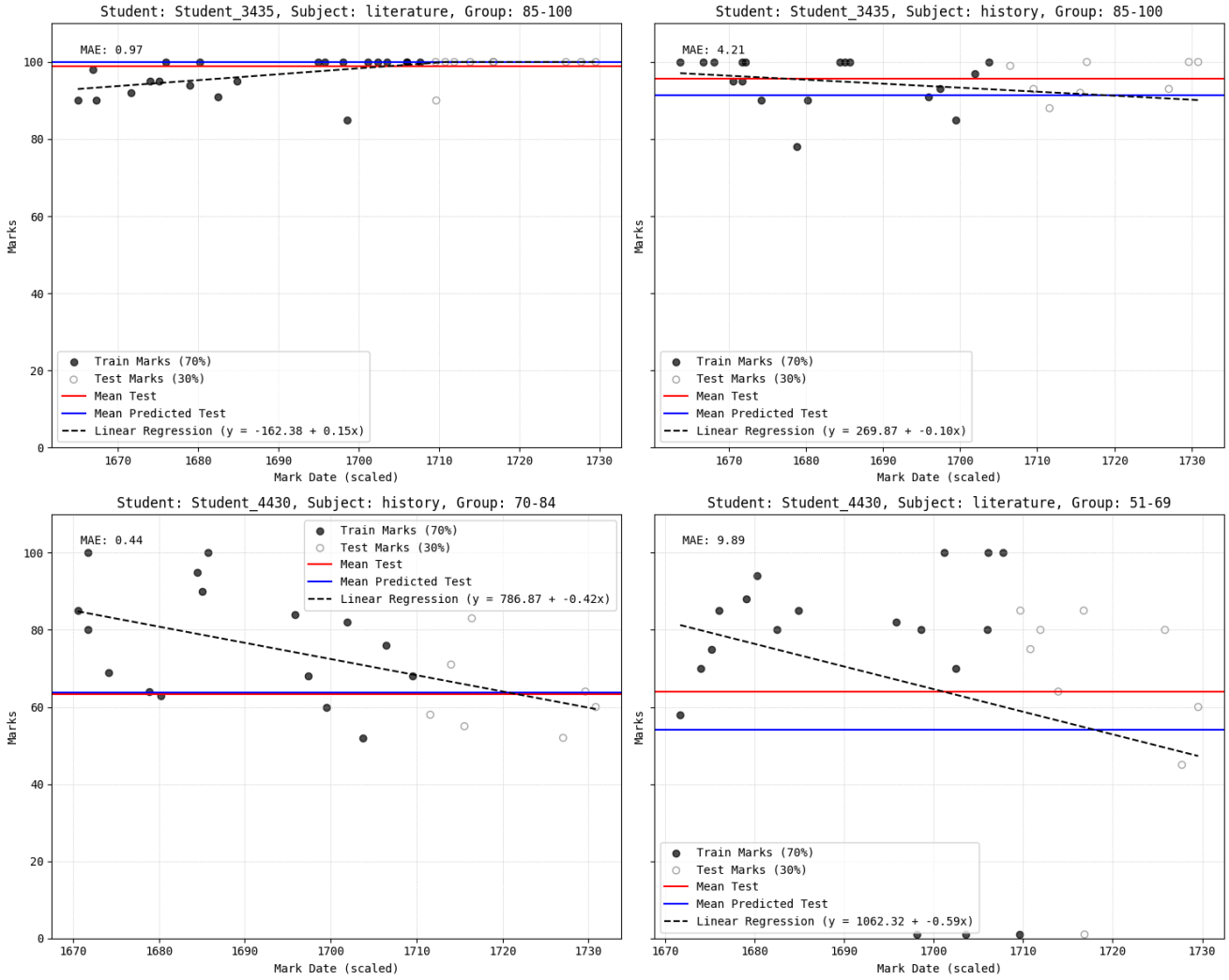

Рис. 6. Линейная регрессия на оценках учеников разных групп

Посчитаем среднюю ошибку для всех временных рядов.

Mark	MAE
51-69	26.159391
70-84	13.083765
85-100	7.314302

Таблица 4. MAE линейной регрессии на оценках учеников.

Время обучения и предсказания средних оценок на всех составленных рядах: **0.02 sec**

2.5.2 ARIMA

Разобьем датасет на такие же выборки как и с линейной регрессией. Сначала протестируем на ней модель *ARIMA* с параметрами[1] (1, 0, 0), обрезая при этом предсказания в 0 и 100.

2.5.2.1 ARIMA без автоматического подбора параметров

Получим следующие ошибки

Mark	MAE
51-69	14.075952
70-84	11.708339
85-100	5.768409

Таблица 5. Mean Absolute Error by Mean Mark. ARIMA without automatic parameter selection.

Время обучения и предсказания средних оценок на всех составленных рядах: **0.86 sec**

2.5.2.2 ARIMA с автоматическим подбором параметров

Для автоматического подбора параметров для модели **ARIMA** будем использовать функционал библиотеки `pmdarima` и метода `auto_arima`.

Этот метод позволяет подобрать параметры, предварительно проверив данные на стационарность с помощью теста *Квятковского–Филлипса–Шмидта–Шина* (параметр ***d***), осуществляется перебор возможных комбинаций p и q в заданных диапазонах. Для каждой комбинации параметров вычисляется значение информационного критерия, и выбирается модель с минимальным значением. [5]

Mark	MAE
51-69	6.723553
70-84	7.295925
85-100	3.511832

Таблица 6. MAE Auto Arima на оценках учеников.

[1]Параметры в виде (p, d, q), где $\boldsymbol{p}$ - порядок авторегрессии (AR, AutoRegressive), $\boldsymbol{d}$ — порядок интеграции (I, Integrated), $\boldsymbol{q}$ - порядок скользящего среднего (MA, Moving Average).

Время обучения и предсказания средних оценок на всех составленных рядах: **39.24 sec**

2.5.3 Prophet by facebook

Prophet — это библиотека с открытым исходным кодом, разработанная командой Core Data Science компании Meta (Facebook) для прогнозирования временных рядов. Она реализована на языках R и Python и предназначена для моделирования данных с выраженной сезонностью и трендами. Модель основывается на аддитивной регрессии, комбинируя нелинейные тренды с ежегодной, еженедельной и ежедневной сезонностью, а также учитывая эффекты праздников.

Составим для каждой группы учеников и для каждого ученика в этой группе список строк, где колонка `ds` - дата оценки, а `y` - значение оценки

Построим предсказания модели (среднее всех предсказанных значений в данной задаче) **Prophet**

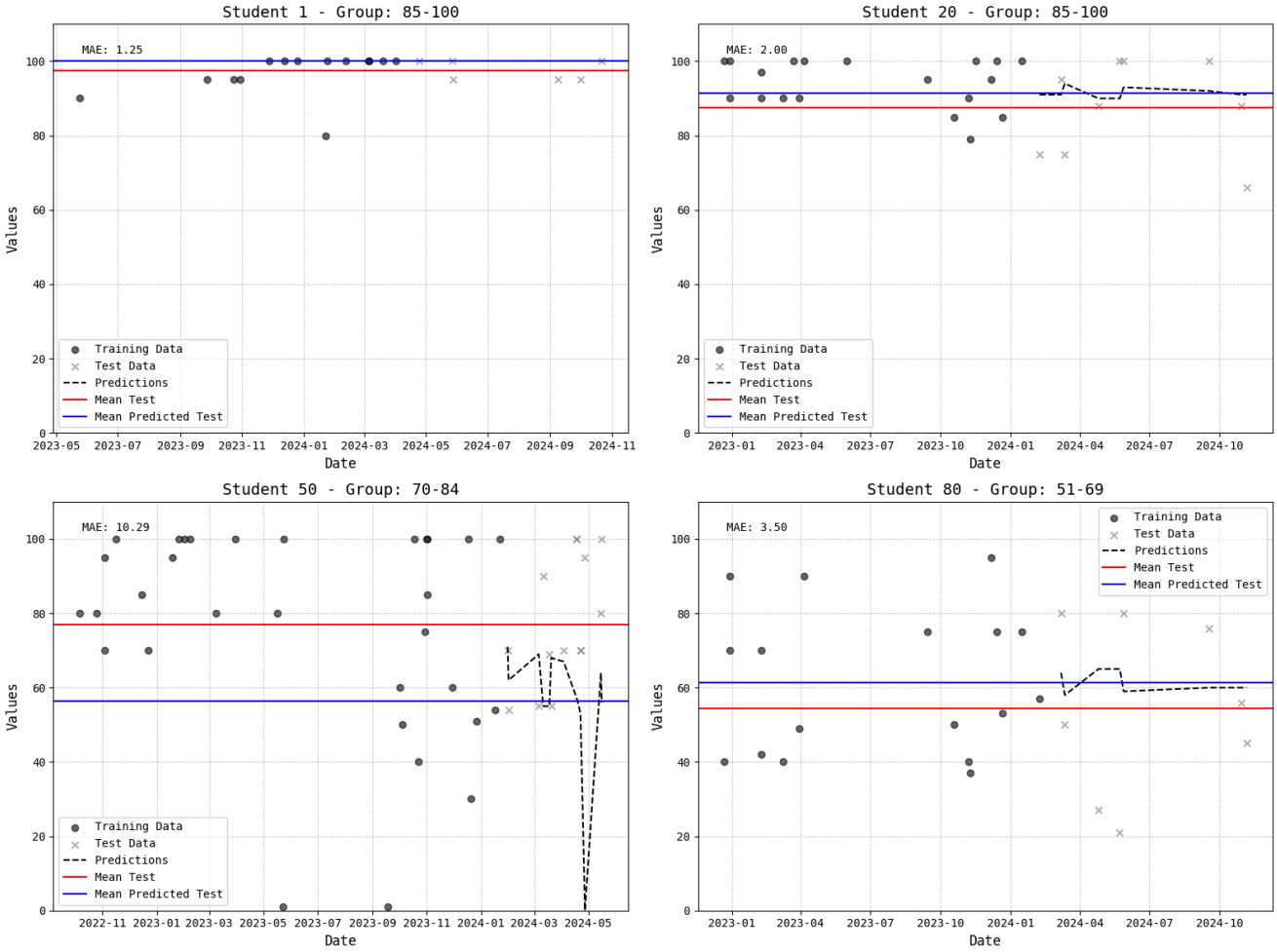

Рис. 7. Prophet на оценках учеников разных групп

Из Рис. 7 видно, что **prophet** переобучается и пытается учесть паттерны, которые в действительности являются случайными, поэтому часто ошибочно предсказывает данные (например, 1)

Mark	MAE
51-69	25.139706
70-84	11.618982
85-100	6.794221

Таблица 7. MAE prophet на оценках учеников.

Время обучения и предсказания средних оценок на всех составленных рядах: **11.8 sec**

3 Результаты и обсуждение

Составим финальную таблицу, отражающая полученные результаты для каждой из моделей. Зеленым обозначены лучшее значение в каждой метрике, красным - худшее.

Method	time (sec)	MAE (51-69)	MAE (70-84)	MAE (85-100)
Linear Regression	0.02	26.159391	13.083765	7.314302
AIRMA	0.86	14.075952	11.708339	5.768409
Auto-ARIMA	39.24	6.723553	7.295925	3.511832
Prophet	11.8	25.139706	11.618982	6.794221

Таблица 8. Итоговые метрики

В результате анализа (Таблица 8) мы подтвердили заранее предсказанную закономерность: будущий средний балл учеников с более низким текущим средним баллом сложнее предсказать используй методы, рассмотренные в статье из-за высокой дисперсии их оценок.

В качестве дополнения к `Prophet` часто используют такие алгоритмы, как *LightGBM* или *Optuna*, что может значительно повысить точность *Prophet* и снизить влияние выбросов на данные.

4 Актуальность

Результаты исследования являются актуальными и важными для следующих групп лиц:

1. **Разработчики** электронных дневников. Благодаря этому исследованию можно понять, какие алгоритмы лучше подойдут в задаче предсказания среднего балла ученика прямо в приложении.
2. **Ученики**. Основываясь на прошлом опыте, модель может предсказать будущую оценку ученика. Это полезно, например, при необходимости повысить успеваемость.

5 Заключение

Из-за низкой абсолютной ошибки у `Auto ARIMA` есть высокий потенциал для точного предсказания итоговой оценки независимо от дисперсии оценок. `Auto ARIMA` может учесть общие паттерны в данных (например, тренд), но не будет переобучаться, как `Prophet` без дополнительных модификаций ⇒ если на обработку оценок ученика можно выделить достаточное количество времени, отталкиваясь от вычислительных мощностей

устройства/сервера, то этот вариант подойдет для решения такой задачи с относительно высокой точностью.

6. Список литературы

1.Asteriou, D. ARIMA models and the Box–Jenkins methodology / D. Asteriou, S. G. Hall // Applied Econometrics. – 2011. – Т. 2. – ARIMA models and the Box–Jenkins methodology. – № 2. – С. 265-286

2.Sean J. Taylor, B. L. Forecasting at scale. / B. L. Sean J. Taylor // The American Statistician. – Forecasting at scale.

3.Kolambe, M. Forecasting the Future: A Comprehensive Review of Time Series Prediction Techniques / M. Kolambe // Journal of Electrical Systems. – 2024. – Т. 20. – Forecasting the Future: A Comprehensive Review of Time Series Prediction Techniques. – С. 575-586

4.Freedman, D. Statistics (international student edition) / D. Freedman, R. Pisani, R. Purves // Pisani, R. Purves, 4th edn. WW Norton & Company, New York. – 2007. – Statistics (international student edition)

5.Kwiatkowski, D. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? / D. Kwiatkowski, P. C. Phillips, P. Schmidt, Y. Shin // Journal of Econometrics. – 1992. – Т. 54. – Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?. – № 1. – С. 159-178