

Принцип мета-обучения для отбора моделей при прогнозировании временных рядов

Зехов Матвей

Факультет экономических наук НИУ ВШЭ

Learn more:



Что за страшные слова?

Задача прогнозирования временных рядов – весьма специфическая и требующая немало навыков. Даже при ограниченном пуле моделей может потребоваться значительное время для оценки всех моделей на конкретном ряде и выбора наилучшей. А если рядов много и выбрать модель необходимо в кратчайшие сроки? На помощь придет мета-обучение!

Не стоит пугаться этого страшного слова, за ним кроется довольно простая идея. Вместо обучения каких-либо моделей непосредственно на самих рядах, предлагается следующее:

1. Сформируем некоторое множество рядов, которое будет нашей выборкой.
2. Сгенерируем на основе выборки некоторое количество симуляционных рядов для расширения выборки.
3. Каждый ряд новой выборки разделим на тренировочную и тестовую части.
4. На основе тренировочной части вычислим заранее определенный список характеристик. Это и будут наши мета-данные.
5. На основе тренировочной части обучим каждую модель из заранее определенного пула моделей. На основе каждой модели построим прогноз соответственно для линии тестовой части ряда и вычислим ошибки. На основе наименьшей ошибки присвоим ряду лейбл в виде названия модели.
6. Получив на основе разметки рядов матрицу характеристик и вектор лейблов, получим набор данных каноничного вида, пригодный для обработки любого мультиклассового классификатора.
7. Обучаем классификатор. Например, неплохо подойдет случайный лес.
8. Для построения прогноза на новых рядах необходимо вычислить их характеристики и построить на них прогноз предобученного случайного леса.

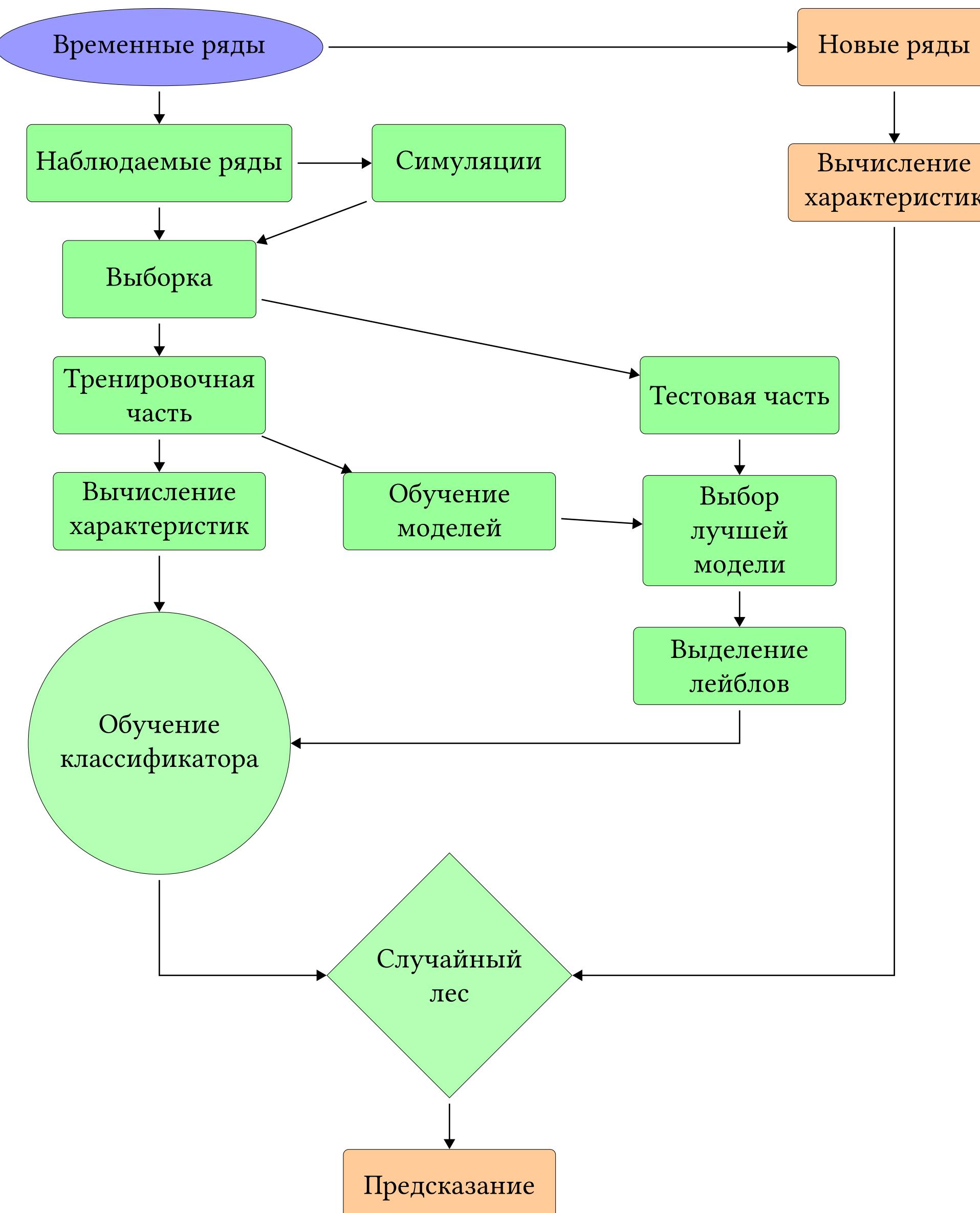


Рис. 1: Алгоритм отбора моделей при помощи мета-обучения

Таким образом, все временные затраты на отбор модели можно ограничить лишь вычислением характеристик ряда и построением прогноза на предобученном случайном лесе.

Параметры выборки			Задачи и итоги генерации		
Класс	Количество в выборке	Длина ряда	Период сезона	Класс	Исходное количество
Yearly	2000	30	1	Yearly	300
Quarterly	2000	60	4	Quarterly	300
Monthly	2000	156	12	Monthly	300
Weekly	286	315	4	Weekly	100
Daily	2000	500	7	Daily	300
Hourly	245	720	24	Hourly	—
Total	8531			Total	1300
					1273

Таблица 1

Характеристики рядов

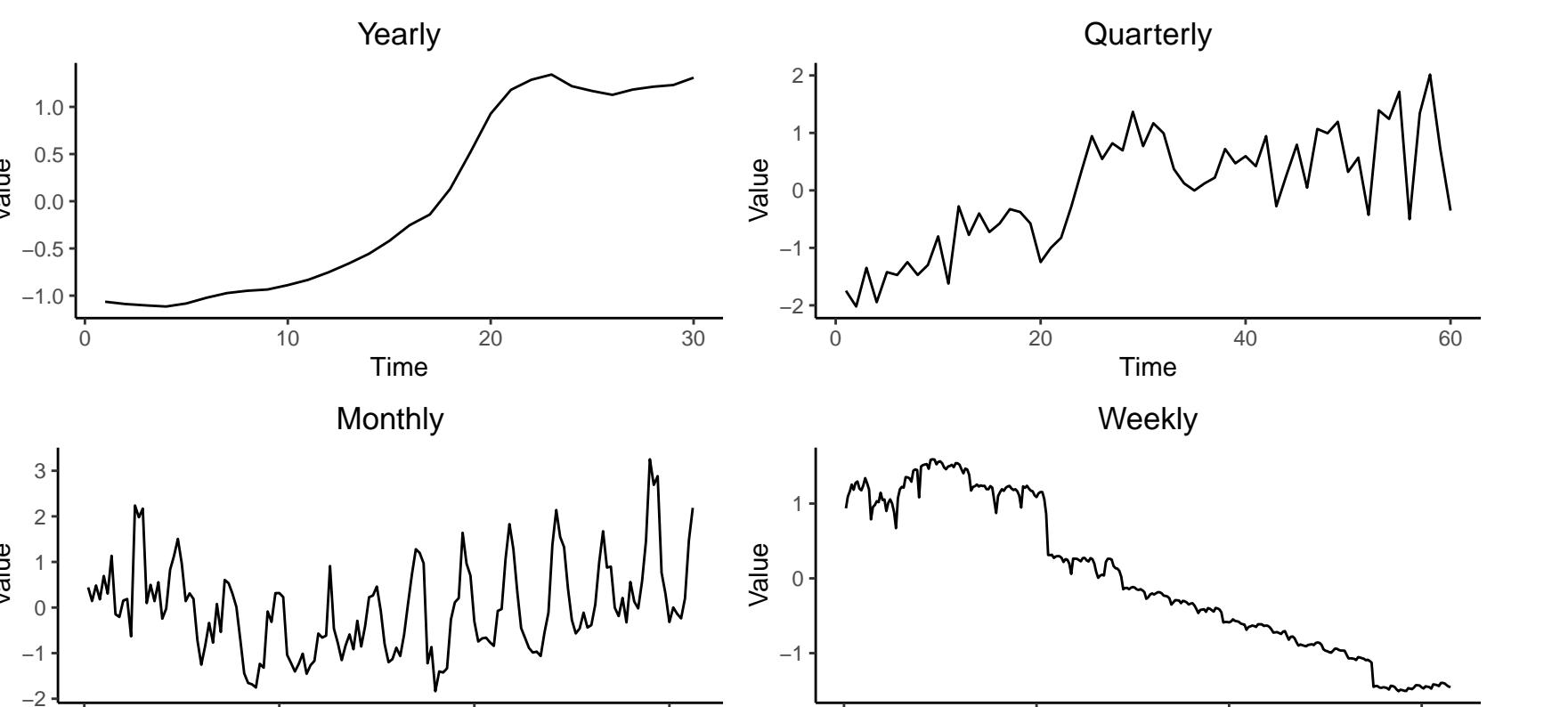


Рис. 2: Примеры рядов из выборки

На графиках на Рис. 2 можно увидеть примеры рядов из нашей выборки. Невооруженным глазом видно, что они различаются по своим признакам. Ниже приведен список из двадцати трех характеристик, которые были вычислены для обучения мета-алгоритма.

- ⊗ Автокорреляция первого порядка.
- ⊗ Автокорреляция первого порядка после однократного и двукратного взятия разностей.
- ⊗ Частная автокорреляция пятого порядка.
- ⊗ Спектральная энтропия
- ⊗ Lumpiness и stability – выборочные дисперсии выборочных средних и выборочных дисперсий по непересекающимся окнам
- ⊗ Crossing points – количество пересечений рядом медианы
- ⊗ Показатель Хёрста, Статистика KPSS-теста, Нелинейность
- ⊗ Куртозис, Скошенность, Хаос
- ⊗ Изменение уровня, Плоские пятна
- ⊗ Максимальное расстояние Кульбака-Лейблера по пересекающимся окнам
- ⊗ Коэффициент формы экстремального распределения
- ⊗ Сила сезонности и сила тренда
- ⊗ Spikiness – выборочная дисперсия "leave-one-out" выборочных дисперсий
- ⊗ Линейность и Кривизна

Визуализируем наши ряды в признаковом пространстве. Для начала попробуем алгоритм UMAP. Хорошо видно чётко выделяющиеся кластеры. На этом рисунке разные люди видели разные рисунки. А что видите вы?

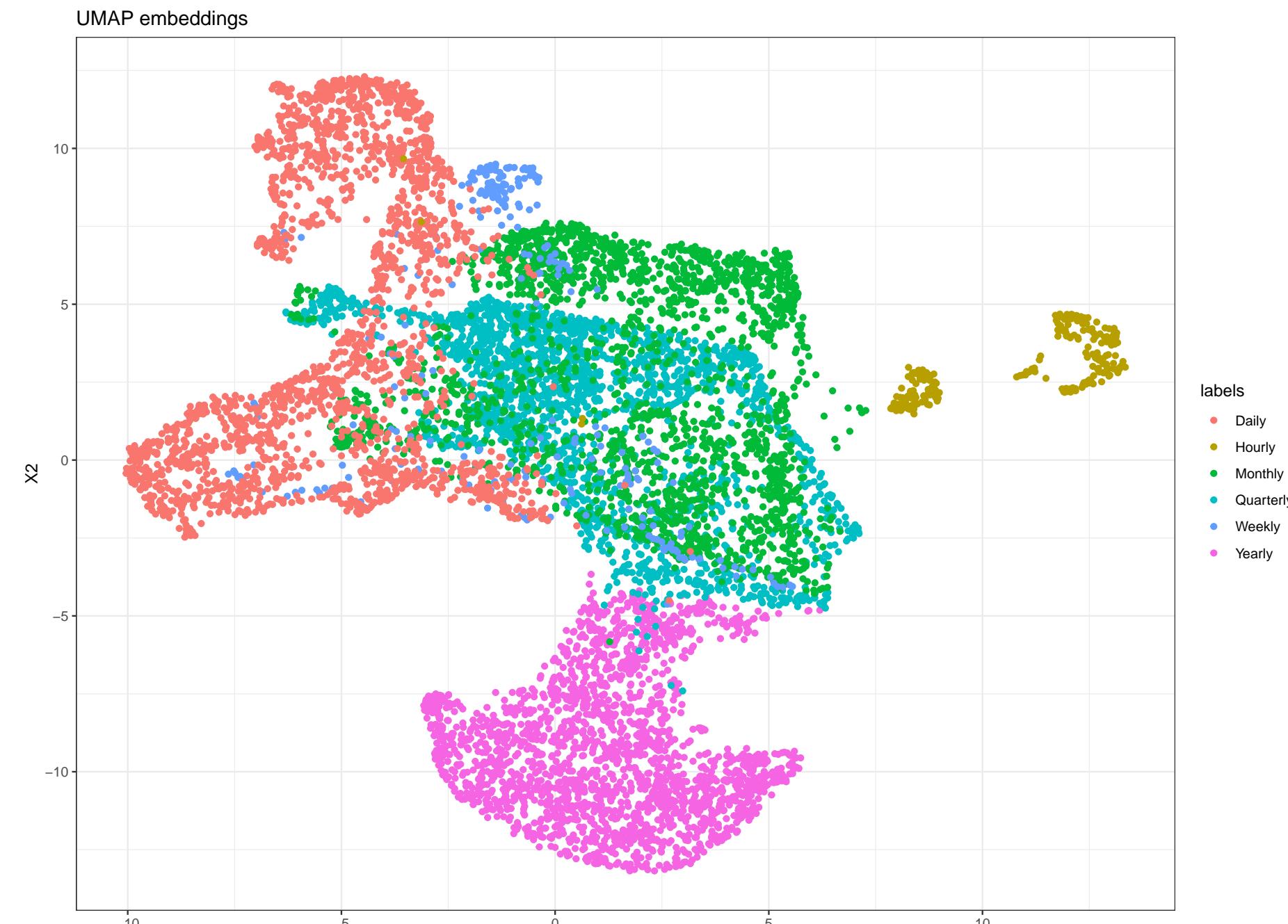


Рис. 3: Двумерные эмбеддинги временных рядов

PCA и генерация новых рядов

Теперь воспользуемся линейным алгоритмом PCA. Визуализируем нашу выборку в пространстве первых двух главных компонент. Результат можно увидеть на первом графике Рис. 4. Как мы видим, кластеры рядов довольно хорошо различимы, однако квартальные, месячные и недельные данные сильно смешиваются.

Все исходные характеристики были нормированы на отрезке $[0, 1]$. Следовательно, можно визуализировать силу каждой характеристики в выборке. Результаты можно увидеть на графиках 2-4 Рис. 4. Например, можно видеть, что дневные ряды наиболее склонны к нестационарности. Они же являются наиболее линейными и наиболее нелинейными. В области максимума первой главной компоненты находятся ряды с наибольшей энтропией. Такие ряды скорее всего будут плохо прогнозироваться.

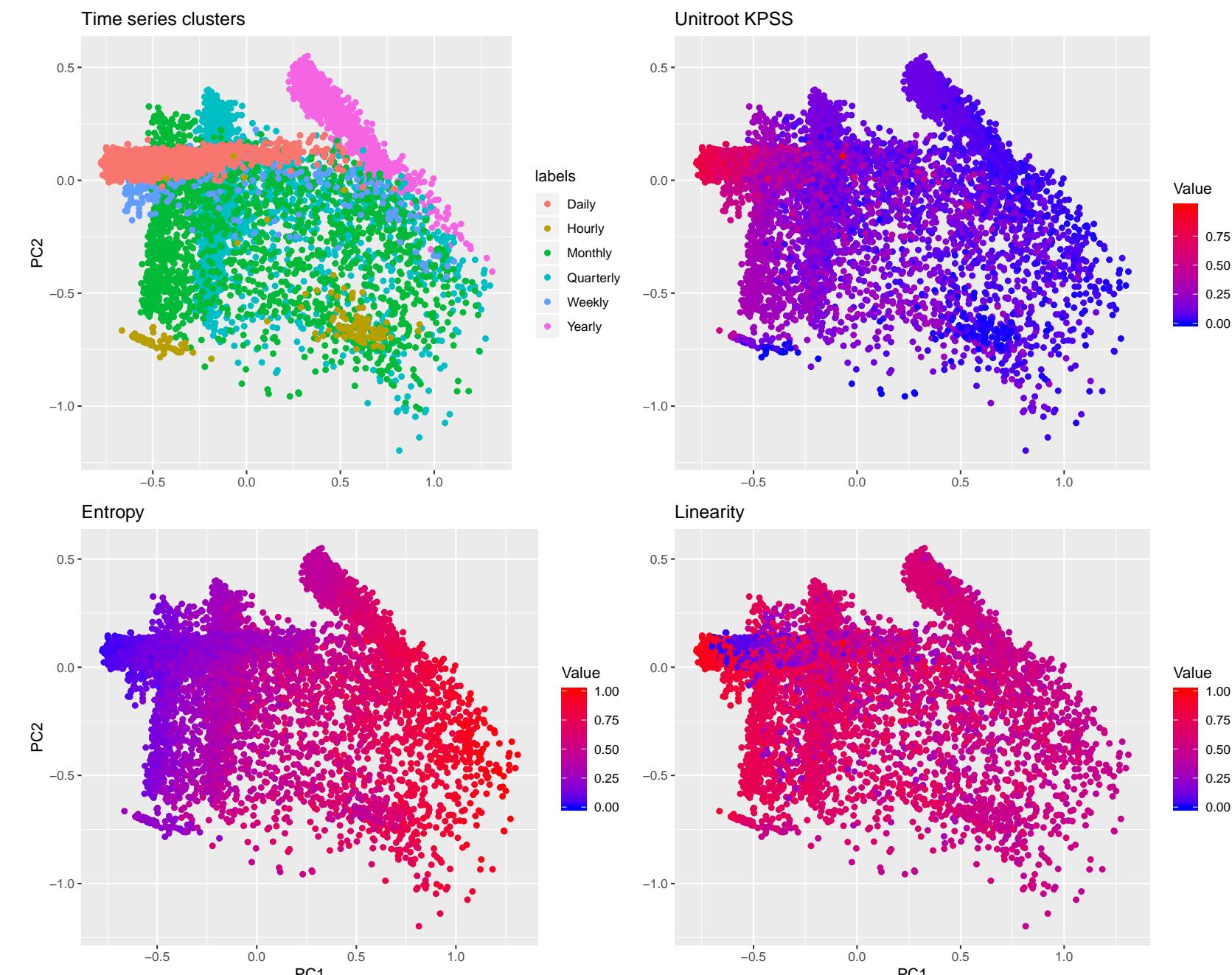


Рис. 4: Выборка в пространстве двух первых главных компонент

Для того, чтобы заполнить белые пятна выборки в пространстве первых двух главных компонент, были симулированы дополнительные ряды. Целевые точки лежали в наименее плотно заполненных участках каждого кластера периодичности.

- ⊗ Для каждой целевой точки формируется стартовая популяция из 10 ближайших соседей
- ⊗ Для каждого кандидата вычисляется вектор характеристик, который проецируется на двумерное пространство.
- ⊗ Вычисляется близость каждого кандидата к целевой точке и фиксируется лучший
- ⊗ С помощью кроссовера, мутаций и выживания ближайших к цели кандидатов формируется следующее поколение

Результаты генерации можно увидеть в виде фиолетовых точек на Рис. 5, а количественные задачи – в Таблице 1. Некоторые симуляции не попали внутрь полигонов, окружающих кластеры, и были отсечены.

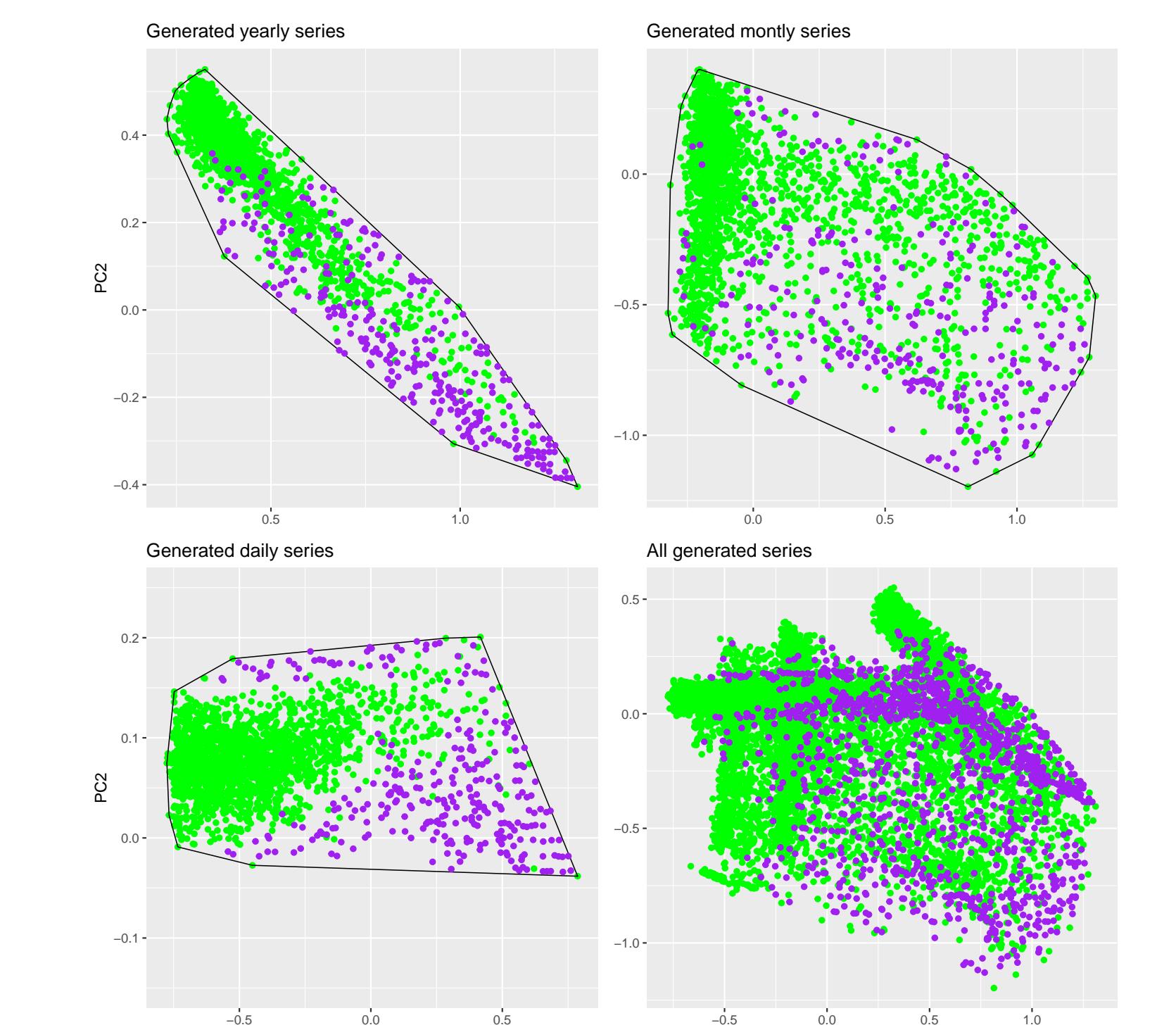


Рис. 5: Генерированные ряды

Разметка и классификатор

Для разметки данных используем девять моделей: Наивная, Сезонная, Случайное блуждание с дрейфом, Простое экспоненциальное слаживание, ETS, Theta, TBATS, Нейронная сеть. Вычислим ошибку каждой модели на каждом ряду и выберем наименьшую. Для измерения ошибки используем SMAPE.

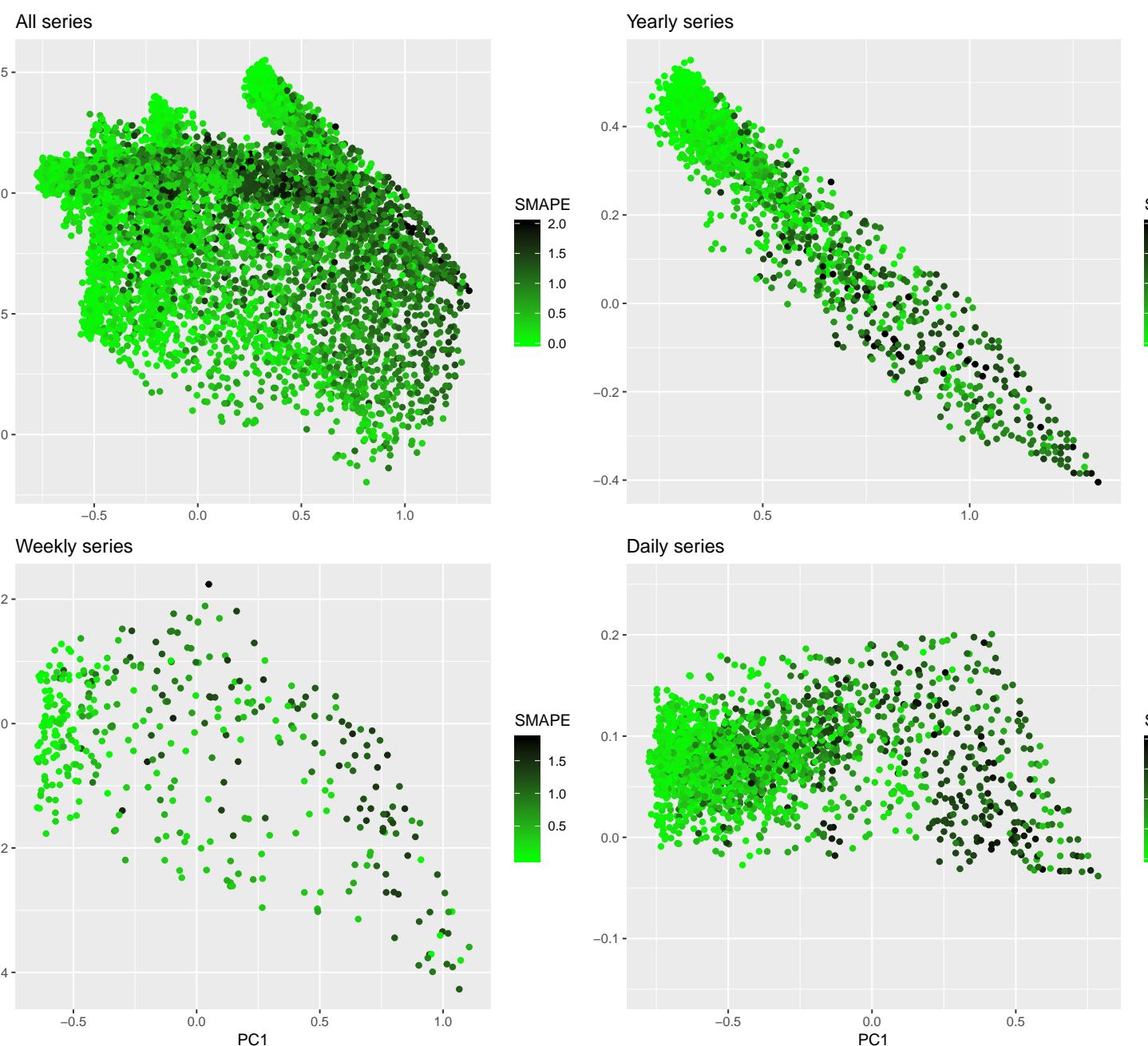


Рис. 6: Наименьшая ошибка на каждом наблюдении

В Таблице 2 можно увидеть, с какой частотой на каждом кластере побеждала ТВАТС, что слегка контриintuitивно, так как одно из главных достоинств модели – моделирование сезонности. Доминирование ARIMA-модели в квартальных и месячных данных может быть косвенно объяснено высокой линейностью этих рядов. Что же касается недельных и дневных рядов, преобладание нейронной сети сложно интерпретировать. Ват. Он не согласуется с графиками линейности и нелинейности.

Модель/Класс	Yearly	Quarterly	Monthly	Weekly	Daily	Качество классификатора
Naive	204 [9%]	161 [7%]	112 [5%]	39 [10%]	90 [4%]	0.384
Seasonal Naive	—	—	266 [12%]	39 [10%]	230 [10%]	0.306
RW with drift	333 [14%]	339 [15%]	271 [12%]	75 [7%]	595 [26%]	0.320
SES	112 [5%]	89 [3%]	86 [4%]	19 [5%]	88 [4%]	0.351
ETS	311 [14%]	288 [13%]	328 [14%]	24 [6%]	56 [2%]	0.320
ARIMA	366 [16%]	443 [19%]	466 [20%]	57 [15%]	247 [12%]	0.320
Theta	166 [7%]	127 [6%]	100 [4%]	11 [3%]	152 [7%]	0.320
TBATS	444 [19%]	288 [13%]	323 [14%]	40 [10%]	181 [8%]	0.320
Neural net	348 [15%]	127 [6%]	345 [15%]	82 [21%]	655 [29%]	0.351
Total	2284	2297	2297	386	2294	

Таблица 2

Результаты обучения классификатора можно увидеть в таблице 2. Пусть качество и не слишком высокое, но оно существенно лучше случайного выбора. Для большого количества классов это вполне естественно. Важно заметить несколько особенностей, очевидных из матрицы ошибок:

- ⊗ При простой истинной модели алгоритм часто прогнозирует более сложную модель
- ⊗ При сложной истинной модели алгоритм часто тоже предсказывает сложную модель, но ошибается.

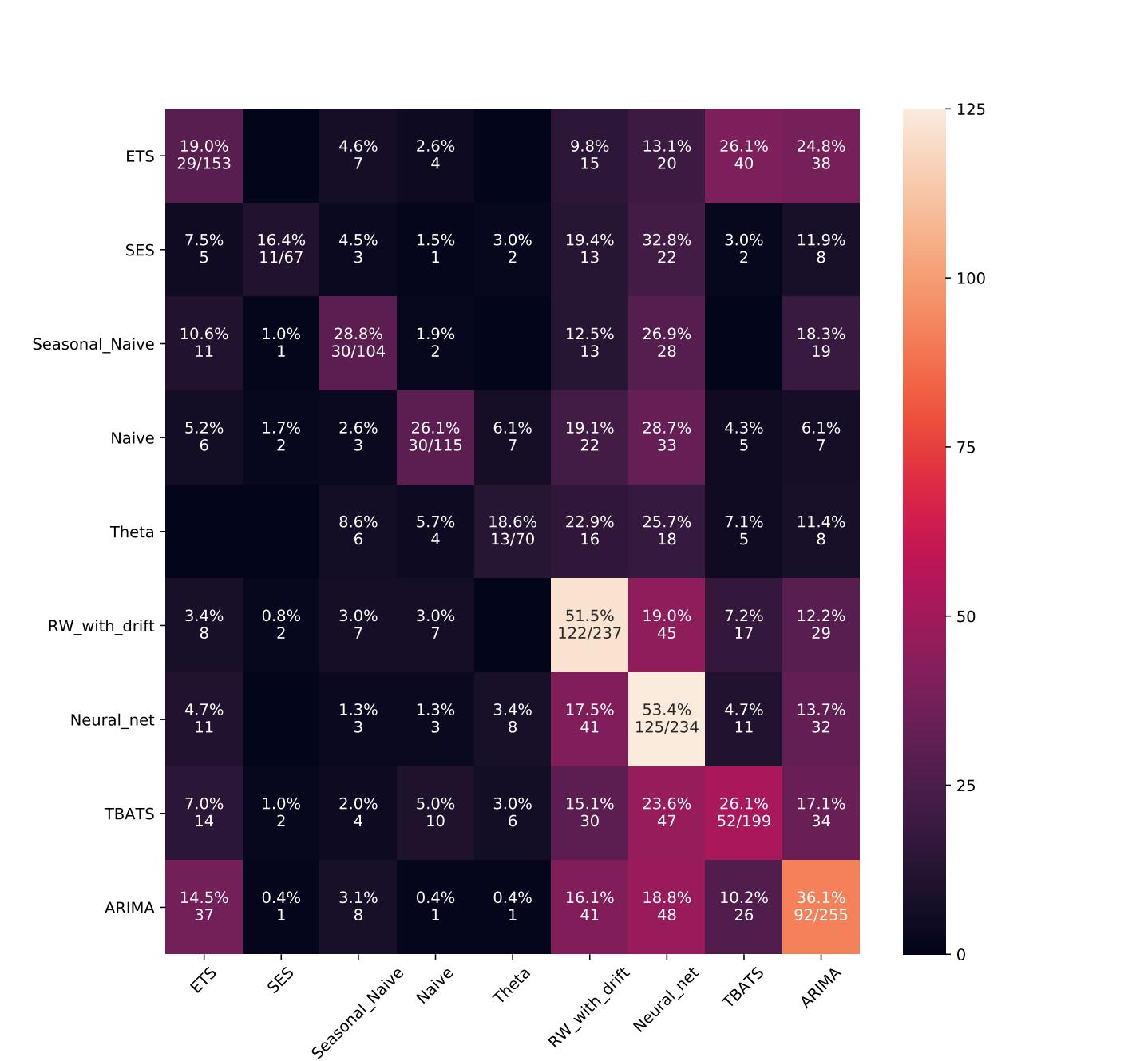


Рис. 7: Матрица ошибок случайного леса