Санкт-Петербургский Государственный Университет
Математическое обеспечение и администрирование информационных систем

Информационно-аналитические системы

Волжина Елена Григорьевна

Методы машинного обучения в задаче предсказания погоды

Курсовая работа

Научный руководитель:
доцент Михайлова Е. Г.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY
Software and Administration of Information Systems

Analytical Information Systems

Elena Volzhina

Machine learning methods in weather prediction problem

Course Work

Scientific supervisor:
associate professor Elena Mikhaylova

Saint-Petersburg
2017

Оглавление

Введение

С ростом производительности вычислительных систем, объемов накопленных данных об окружающем мире и опыта в их обработке находятся всё новые и новые области для применения анализа данных. Одной из таких областей сегодня является и прогнозирование погодных условий и в целом состояния атмосферы.

Прогнозы погоды полезны как обычным людям для принятия бытовых решений, так и в более серьезных областях: в авиации, судоходстве, а также в сельском хозяйстве. Большую пользу знания о погоде в будущем могут принести и бизнесу, в особенности сезонному, в котором спрос на услуги или возможность проводить работы сильно зависит от погоды.

В Яндексе для прогноза погоды используется технология Метеум, комбинирующая прогнозы от нескольких поставщиков и другие данные с помощью методов машинного обучения, а именно градиентного бустинга над решающими деревьями. Модель, полученная таким образом, улучшает качество прогноза относительно каждого из поставщиков.

В рамках этой работы я добавляла к используемым моделью данным информацию о погоде в интересующей нас области в тот момент, когда мы вычисляем свой прогноз. Это может улучшить предсказания для ближайшего будущего, так как оно естественным образом зависит от настоящего.

1. Существующие решения

Наблюдения за погодой и попытки её предсказывать ведутся с тем или иным успехом уже несколько веков. Изначально прогнозы делались на основе опыта наблюдений и примет, но уже в XIX веке с развитием гидродинамики и термодинамики появляются первые математические инструменты для прогнозов.

К концу XIX века научное сообщество располагало инструментами для исследования поведения газов, законами, описывающими это поведение, а также представлением о крупных системах, обуславливающих состояние атмосферы и, как следствие, погоду в конкретной точке. Были сформулированы идеи о решении системы уравнений с заданными начальными условиями для предсказания погоды в будущем.

В начале XX века Льюис Фрай Ричардсон составил систему уравнений, описывающих процессы, по которым меняется состояние атмосферы. Подставив в качестве начальных условий текущее состояние атмосферы можно было решить систему методом конечных разностей и получить прогноз изменения атмосферного давления[4]. При этом задавать начальные условия требовалось с большой точностью, так как даже небольшая погрешность в них приводила к большим изменениям в результате расчётов. Также значительной проблемой была вычислительная сложность процесса, она делала невозможным сколько-нибудь оперативный прогноз погоды в реальных условиях.

Во второй половине XX века появляются всё новые сведения об атмосферных процессах, уточняющие точность моделирования, а также вычислительные ресурсы и технологии, необходимые для регулярного решения подобных систем в разумных временных рамках. Численное моделирование крупномасштабных явлений в атмосфере значительно улучшилось благодаря информации с искусственных спутников Земли[8].

1.1. Классические методы

Классический подход к задаче прогноза погоды включает в себя несколько компонентов. В основе всего лежит сбор данных: как долгосрочные наблюдения о температуре, осадках и прочих параметрах в конкретной местности, так и регулярные замеры с помощью разнообразной техники: приборов, установленных на наземных метеостанциях, метеорологических зондов и даже метеорологических спутников. Далее в дело вступают математические модели атмосферы, настроенные с помощью собранных данных. В этих моделях уравнения гидрогазодинамики описывают процессы в атмосфере, которые влияют на интересующие нас величины, а подставив в качестве параметров реальные данные о состоянии атмосферы в данный момент мы можем получить прогноз её состояния через некоторое время.

Модели атмосферы можно разделить на *глобальные*, покрывающие всю планету, и *региональные*, описывающие ограниченную территорию. Первые более универсальны, зато вторые могут давать лучшее качество для своих областей за счет более тонкой настройки и высокого расширения. Сами прогнозы можно разделить на группы по времени, на которое мы смотрим в будущее: *краткосрочные* (до 72 часов), *среднесрочные* (от 72 часов до 10 суток) и *долгосрочные* (более 10 суток).

Построение и обновление математических моделей атмосферы очень трудозатратно и требует проведения большого количества различных экспериментов, поэтому из-за постоянных изменений климата все использующиеся модели являются в большей или меньшей степени устаревшими. Помимо этого, так как физика многих процессов в атмосфере еще недостаточно изучена, у всех моделей есть те или иные погрешности в прогнозировании, при этом зачастую эти погрешности имеют постоянную природу: например, одна из моделей может завышать температуру, а другая – занижать давление в горах.

Учитывая объемы информации, человеку сложно найти эти шаблоны в ошибках моделей. Но благодаря современным статистическим методам и вычислительным мощностям этот процесс

можно автоматизировать. Именно эта идея лежит в основе технологии Метеум, которая используется для прогноза погоды в Яндексе. По накопленным данным о прогнозах разных моделей и фактической погоде в моменты, на которые делались эти прогнозы, можно построить модель, корректирующую прогнозы поставщиков и улучшающую качество итогового прогноза.

1.2. Машинное обучение, технология Метеум

Машинным обучением называют подход к решению задач, заключающийся в анализе большого количества накопленных наблюдений с целью найти в них ранее неизвестные зависимости. Нередко такие зависимости сложно или невозможно (например, в силу недоступности информации о части факторов) описать формально, но благодаря статистическому подходу их можно приблизить с достаточным для практического применения качеством[1].

В задаче предсказания погоды удобно использовать методы *обучения с учителем* – они применяются, когда на каждое наблюдение, описанное вектором факторов, имеется "правильный ответ". Среди них выделяют методы *классификации*, когда ответом является метка из конечного множества (например, если предсказывается наличие осадков или их тип), и методы *регрессии*, в этом случае ответ – непрерывная величина (например, температура или количество осадков в миллиметрах).

В качестве примера можно упомянуть The Weather Company, которая для подсчёта прогноза погоды комбинирует информацию из более чем 150 источников данных с помощью машинного обучения[3].

Сервис Яндекс.Погода использует для прогноза погоды технологию Метеум. На основе накопленных данных, которые будут описаны подробнее далее, составляются отдельные модели машинного обучения для интересующих нас параметров: температуры, давления, скорости и направления ветра, а также типа облачности и осадков. Важно уточнить, что температура предсказывается не сама по себе, а

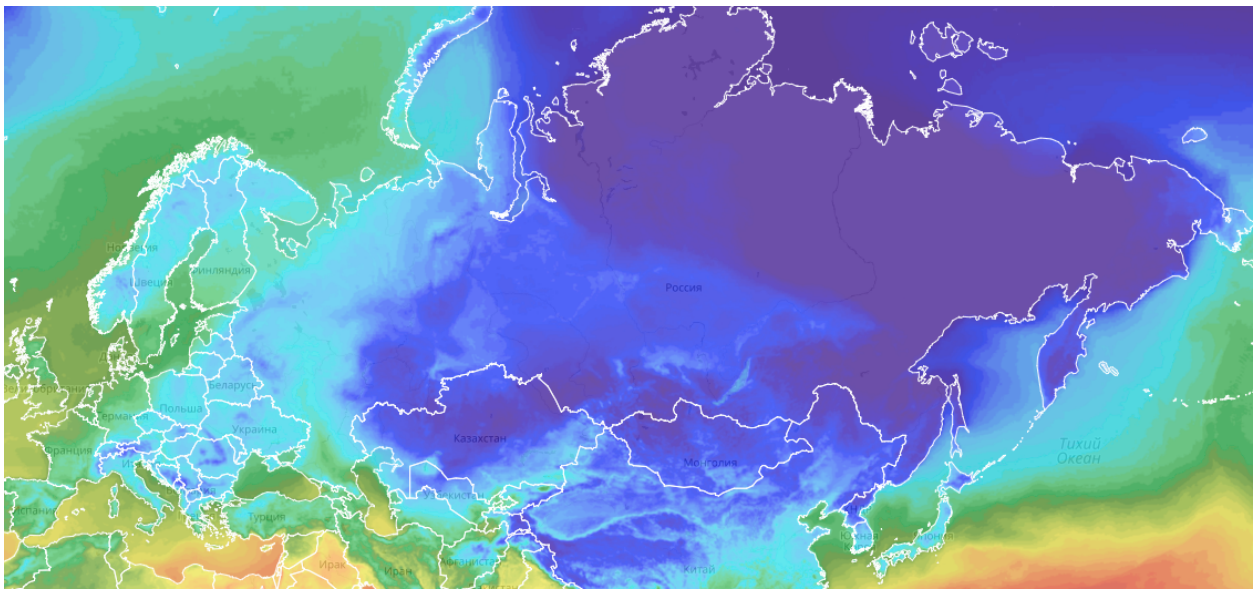

Рис. 1: Карта температуры на Яндекс.Погоде

как разница между температурой и климатическими данными – усредненными показаниями за десятки лет для этого времени года. Это позволяет частично избавиться от сезонности в изменениях температуры.

Для обучения в числе прочего используются данные, вычисляемые через классические модели прогнозирования. Сравнивая эти прогнозы с фактическими показаниями, модель находит зависимости целевой переменной от прогнозов разных показателей (например, для осадков кроме, собственно, прогнозов осадков, важными факторами оказываются также прогнозы облачности), а также учится исправлять упомянутые выше погрешности разных моделей.

В рамках данной работы будет проверена гипотеза о том, что полезной информацией для предсказаний погоды в ближайшем будущем окажется состояние погоды на данный момент. Для этого будут использоваться показания с одной или двух ближайших метеорологических станций.

2. Методология

Прежде чем приступать к экспериментам, следует разобраться с данными, которыми мы располагаем, решить, как мы будем собирать обучающую выборку с известными ответами и готовить её к использованию. Далее нужно выбрать метод для поиска решающей функции, которая бы приближала правильные ответы, находя зависимости в данных. Также необходимо определить метрики, с помощью которых будет приниматься решение, успешным считать эксперимент или нет. После этого можно будет приступать к основной задаче: добавить к обучающей выборке данные о текущих погодных условиях в месте, для которого рассчитывается прогноз; попробовать обучить на расширенных данных модель для предсказания температуры; оценить по метрикам улучшение.

2.1. Данные для обучения

При составлении прогноза мы располагаем множеством данных, полученных от поставщиков или вычисленных самостоятельно. Можно выделить несколько групп этих данных:

1. *прогнозы погоды*, сделанные при помощи математического моделирования физических процессов в атмосфере; в числе поставщиков следующие глобальные модели: Global Forecast System (GFS [5]), Japan Meteorological Agency (JMA), European Centre for Medium-Range Weather Forecasts (ECMWF [6]), Canadian Meteorological Centre (CMC). Также на серверах Яндекс.Погоды рассчитывается региональная модель для собственных прогнозов – Weather Research and Forecasting Model (WRF). У разных моделей могут различаться частота расчётов, координатная сетка, а также набор параметров в прогнозе, это необходимо учитывать при сборе обучающей выборки;

2. *фактические данные* о погоде, снятые на метеостанциях по всему миру с помощью статических приборов и метеозондов.

Используются станции из сети Всемирной метеорологической организации;

3. *климатические данные*, посчитанные на основе десятков лет наблюдений метеостанций;

4. *радарные данные* об осадках и облачности, *спутниковые снимки.*

Чтобы применять на этих данных какие-либо алгоритмы машинного обучения, нужно очистить их от выбросов и шума, привести всех поставщиков к единой координатной сетке и собрать обучающую выборку. В момент, когда мы делаем прогноз, мы можем использовать полученные ранее прогнозы от поставщиков, статистические данные о климате, а также информацию о погоде в интересующей нас точке в данный момент (или, с учётом задержек, в недавнем прошлом).

Нам потребуется множество наблюдений $X = \{x_i = (f_i^1, \cdots, f_i^M), i \in \overline{1..N}\}$, элементы которого соответствуют векторам признаков, полезных для предсказывания в текущий момент $gentime$ состояния погоды на момент в будущем $time$. Например, в числе этих признаков будут прогнозы различных погодных параметров: температуры, влажности, скорости и направления ветра, осадков и облачности. Эти прогнозы должны быть сделаны на время, близкое к интересующему нас $time$, при этом при сборе обучающей выборки важно не заглядывать в будущее, то есть время генерации этих прогнозов поставщиками должно быть не позже, чем $gentime$ (а с учётом задержек при передаче данных лучше брать прогнозы, сделанные еще раньше). Помимо множества наблюдений X нам понадобится множество истинных значений $y = \{y_i, i \in \overline{1..N}\}$, в нашем случае это разница температуры, которая была получена с метеостанции в момент $time$ (на это время мы делали прогноз), с усредненной за много лет температурой в этот день и в это время: $temperature\_delta = fact\_temperature - climate\_temperature$.

2.2. Решающие деревья и градиентный бустинг

Для проведения экспериментов по прогнозу температуры на основе имеющихся данных, был выбран алгоритм *градиентного бустинга над решающими деревьями*. Использовалась реализация этого алгоритма под названием Матрикснет, разработанная и использующаяся в компании Яндекс.

Решающее дерево – это алгоритм предсказания, описывающийся бинарным деревом, у которого каждой внутренней вершине v поставлен в соответствие некоторый предикат $\beta_v : X \to \{0, 1\}$, а каждому листу соответствует метка с ответом алгоритма[1]. После обучения применяется дерево следующим образом: для фиксированного $x \in X$ начинаем с корневой вершины (v_{root}). Вычисляем значение $\beta_{v_{root}}(x)$. Если получили 0, переходим в левого потомка, если 1 – в правого. Продолжаем до тех пор, пока не окажемся в листе, метку которого и возвращаем в качестве ответа.

Градиентный бустинг – алгоритм, с помощью которого по обучающим данным последовательно строится композиция из простых алгоритмов предсказания, причём каждый следующий алгоритм в композиции стремится уменьшить ошибку, которую даёт уже накопленный ансамбль. Такой метод построения называют *жадным*, так как он вместо попытки сразу найти оптимальное решение, идёт к нему итеративно, делая на каждом шаге локально-оптимальный выбор[2].

Градиентный бустинг позволяет из простых предсказательных моделей, не приносящих хороших результатов при самостоятельном использовании, получить композицию, хорошо приближающую целевую функцию. При этом модель оказывается устойчивой к переобучению и эффективной в применении (так как простые элементы композиции можно быстро вычислить параллельно) [7].

2.3. Метрики

Для ответа на вопрос, улучшают изменения прогноз или ухудшают, необходимо выбрать метрики, с помощью которых в дальнейшем можно будет сравнивать качество разных моделей.

Стандартной метрикой в задаче регрессии (предсказания действительного числа) является *квадратный корень из среднеквадратичной ошибки* (Root Mean Square Error, RMSE). Он показывает, как сильно в среднем прогнозы модели отличаются от фактических данных. Для столбца фактических показаний y и столбца ответов модели $\hat{y}$ значение этой метрики вычисляется следующим образом:

$$RMSE(y, \hat{y}) = \sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

Чтобы значения метрик объективно отражали качество моделей, нужно считать их на новых для модели примерах. В нашем случае достаточно разделить обучающую выборку на два непересекающихся подмножества (X_{train} и X_{test}): наблюдения до определенной даты и после неё, на первом подмножестве обучить модель, а на втором вычислить метрики. Так мы не только будем проверять модель на примерах, которые не встречались ей при обучении, но и не дадим ей подглядывать в будущее (это плохо, так как погоду в последовательные моменты времени нельзя считать независимой).

3. Эксперименты

Для улучшения прогноза погоды в ближайшем будущем было опробовано добавление в обучающую выборку данных о текущей погоде в области, для которой делается прогноз. Обученная модель сравнивалась с использующейся в данный момент в Яндекс.Погоде. Целевой переменной была разница между фактической температурой и средней для этих местности и времени ($temperature\_delta = fact\_temperature - climate\_temperature$).

Данные о текущих погодных условиях могут помочь предсказать погоду в будущем только на небольшой срок. Для экспериментов было выбрано ограничение в 7 часов, и при более дальних прогнозах данные с ближайших станций не использовались.

3.1. Сбор обучающей выборки

К уже существующей обучающей выборке, собранной из векторов прогнозов поставщиков и других данных в качестве множества наблюдений ($X = \{x_i = (f_i^1, \cdots, f_i^M), i \in \overline{1..N}\}$) и реальных $temperature\_delta$ в качестве верных ответов ($y = \{y_i, i \in \overline{1..N}\}$), нужно присоединить данные о фактической погоде в момент генерации прогноза. Для этого было построено следующее соответствие: для каждой станции, с которой поступают фактические данные, вычислены две ближайшие станции в пределах 150 километров. Далее в обучающую выборку для конкретной станции попадают либо данные с этих двух ближайших, либо только с одной из них (а на место второй – данные с самой станции).

Необходимо при сборе данных для обучения учесть, что в условиях реального применения модели будут задержки в получении данных от поставщиков прогнозов и фактов. Таким образом, если для генерирующегося в момент $gentime$ прогноза мы будем брать фактические данные, снятые на станции в тот же момент $gentime$, мы получим слишком позитивную оценку метрик формулы. В реальности

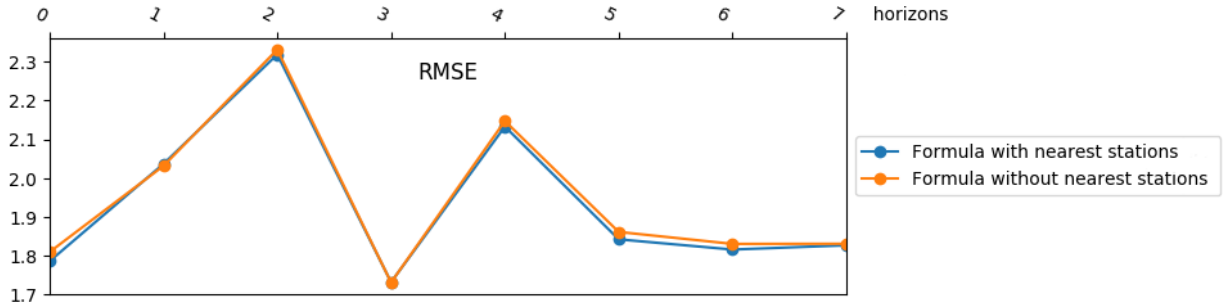

Рис. 2: Модели, обученные на всех краткосрочных горизонтах

задержки имеют порядок десятков минут, и при генерации прогноза на сервисе использоваться будут соответственно устаревшие данные.

3.2. Удалённость времени прогноза

Будем называть *горизонтом* прогноза разницу между временем, на которое делается прогноз ($time$), и моментом, когда мы его подсчитываем ($gentime$), в часах.

Для начала был проведен эксперимент с новой обучающей выборкой для всех краткосрочных горизонтов (вплоть до 72-го часа), при этом данные о ближайших станциях были только для первых семи часов. Такое усовершенствование не принесло видимых улучшений, даже на первых семи горизонтах изменения оказались незначительными (Рис. 2), что объясняется относительно небольшим числом строк с данными с ближайших станций в обучающей выборке. Эффект от этих данных оказался невысок – если упорядочить приблизительно 250 признаков каждого наблюдения в обучающей выборке по влиянию на ответ модели, фактор $fact1\_temperature\_delta$ (вычисленный как разница температуры с ближайшей станции и $climate\_temperature$) оказался на 115 месте.

Было решено попробовать обучение отдельной модели только для первых семи часов. Тогда для всех строк в обучающей выборке модель будет знать показания с ближайших станций, благодаря чему эффект от этих данных должен повыситься. Чтобы не увидеть мнимое улучшение относительно базовой модели только за счёт

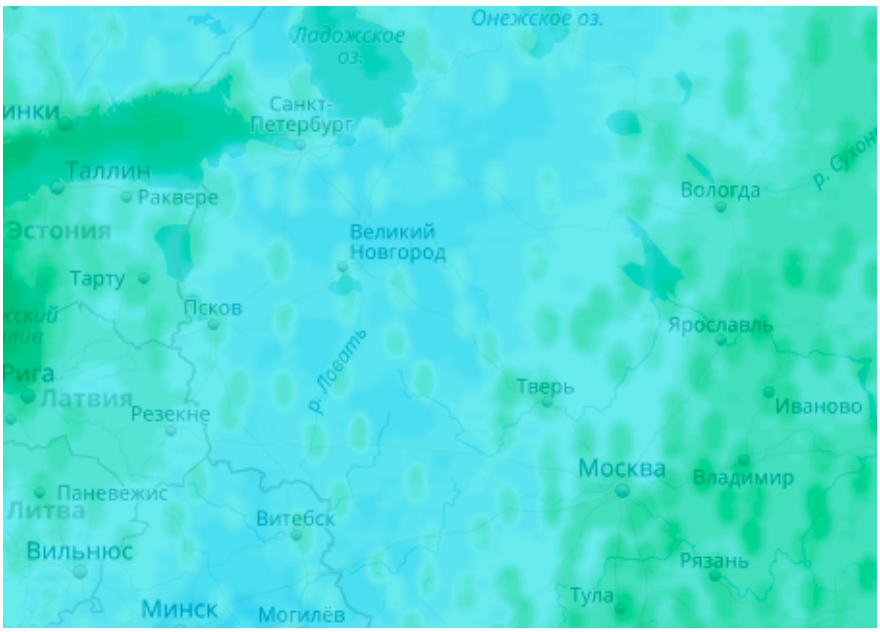

Рис. 3: Пятна вокруг станций

того, что у усовершенствованной модели в обучающей выборке нет удалённых по времени наблюдений с меньшим качеством прогноза у поставщиков, сравнение проводилось с моделью без данных от ближайших станций, но обученной также только на первых 7 часах. Однозначного улучшения по метрикам также не получили, хотя фактор $fact1\_temperature\_delta$ стал вносить значительный вклад в ответ модели – он оказался на 5 месте по эффекту, наравне с прогнозами температуры от поставщиков.

Далее были испробованы другие ограничения на число часов, которым ограничивается действие прогноза в будущее, хотелось найти баланс между сокращением применимости модели и улучшением метрик. По результатам экспериментов было выбрано ограничение в 3 часа от момента генерации прогноза. Далее прогнозы полученной модели были нарисованы на карте, чтобы увидеть, как они будут выглядеть для пользователя.

На карте были обнаружены пятна вокруг станций, поставляющих данные (Рис. 3). Они демонстрируют, что раз $temperature\_delta$ с ближайшей станции имеет большой вклад в прогноз модели, то на границе, после которой станция перестаёт считаться ближайшей, и данные от неё для модели пропадают, видны различия в прогнозе.

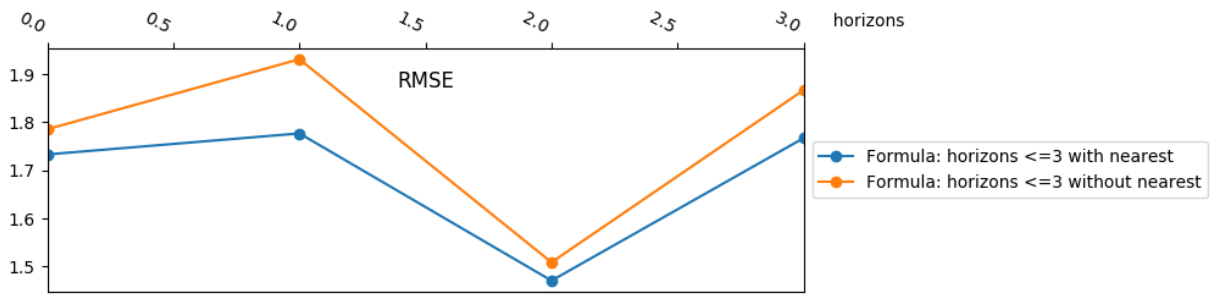

Рис. 4: Модель с использованием расстояний до ближайших станций

3.3. Расстояние до ближайших станций

Для решения проблемы с пятнами было решено использовать простое сглаживание данных от ближайших станций, чтобы на границе их присутствия не было резкого скачка в значениях. Кроме этого в данные для обучения были добавлены расстояния от точки прогноза до ближайших станций, но по результатам экспериментов они не оказались полезны для модели.

Для сглаживания был взят прогноз самого точного из наших поставщиков, которым подменялись значения с ближайших станций за пределами радиуса их использования ($max\_distance$). Далее для каждого наблюдения для двух ближайших станций были вычислены такие значения:

$$smoothed\_fact\_temperature\_delta := \begin{cases} fact \cdot (1 - \frac{dist}{max\_distance}) + forecast \cdot \frac{dist}{max\_distance}, \\ \quad \text{if } dist \leq max\_distance \\ forecast, \text{ if } dist > max\_distance \end{cases},$$

где $fact$ – исходное значение с одной из ближайших станций, $dist$ – расстояние до неё, а $forecast$ – прогноз поставщика для точки, где она находится, на момент $time$.

С таким сглаживанием получили модель, которая показывает лучшие метрики, чем модель, обученная также на первых трёх часах, но без показаний с ближайших станций (Рис. 4), и при этом не даёт пятен на картах. Сглаженные температуры с двух ближайших станций оказались на 1 и 4 местах по влиянию на ответ модели.

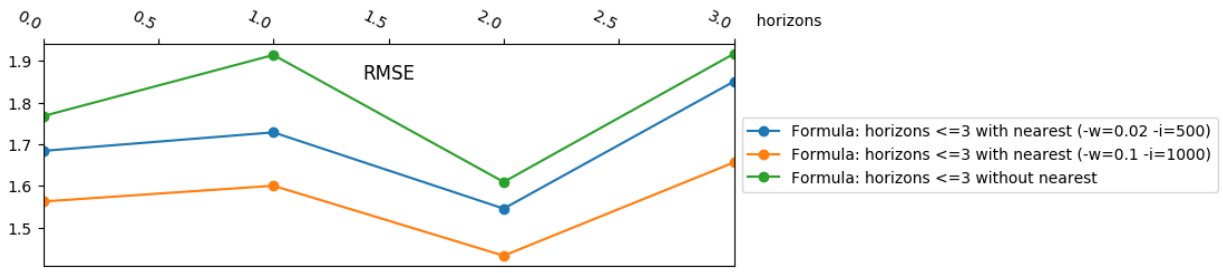

Рис. 5: Эксперимент с параметрами алгоритма Matrixnet

Заключение

При работе над этой задачей было показано, что данные с ближайших станций могут помочь улучшить прогноз температуры. Использовать их нужно с некоторым сглаживанием, чтобы не получить ярко выраженных окрестностей вокруг станций.

В качестве возможных направлений развития этой задачи перечислю следующие:

- использовать ближайшие станции для предсказания не только $temperature\_delta$, но и других показателей;
- использовать кросс-валидацию, чтобы уменьшить влияние конкретного разбиения на обучающую и тестовую выборки и более объективно оценивать изменения;
- поэкспериментировать с параметрами градиентного бустинга, по первым шагам в этом направлении похоже, что возможны значительные улучшения (Рис. 5);
- попробовать другие способы сглаживания показаний со станций по мере удаления точки прогноза от них.

Список литературы

[1] Bishop Christopher M. Pattern Recognition and Machine Learning (Information Science and Statistics). — Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006.

[2] Friedman Jerome H. Greedy function approximation: a gradient boosting machine // Annals of statistics. — 2001. — P. 1189–1232.

[3] IBM. Driving the World's Most Accurate Weather Forecasts // IBM Website. — 2016. — URL: `https://www.ibm.com/blogs/think/2016/12/accurate-weather-forecasts/`.

[4] Lynch Peter. The origins of computer weather prediction and climate modeling // Journal of Computational Physics. — 2008. — Vol. 227, no. 7. — P. 3431–3444.

[5] The NCEP climate forecast system / S Saha, S Nadiga, C Thiaw et al. // Journal of Climate. — 2006. — Vol. 19, no. 15. — P. 3483–3517.

[6] Persson Anders. User Guide to ECMWF forecast products. — 2001.

[7] Stochastic gradient boosting // Computational Statistics & Data Analysis. — 2002. — Vol. 38, no. 4. — P. 367 – 378. — Nonlinear Methods and Data Mining.

[8] Васильев А.А. Вильфанд Р.М. Прогноз погоды // М.:«Моби Дик. — 2008.