УДК 004.85

Определение метеорологической дальности видимости с помощью деревьев решений

Буднов Д.А., Клепиков П.Н.
Алтайский государственный университет, г. Барнаул
danila.budnov@yandex.ru, klepikov.math@gmail.com

Аннотация

Рассматривается возможность применения такого метода машинного обучения, как деревья решений, для определения метеорологической дальности видимости на основе других погодных показателей (температура воздуха, атмосферное давление, относительная влажность, скорость и направление ветра, облачность, текущая погода и др.). Описаны процесс сбора и обработки данных, а также обучение модели и её итоговая точностью.

Ключевые слова: дальность видимости, метеорологическая дальность видимости, деревья решений, машинное обучение, python, scikit-learn.

1. Введение

Погодные условия очень сильно влияют на жизнедеятельность человека. С развитием судоходства и авиации большую роль стала играть такая погодная характеристика, как метеорологическая дальность видимости (МДВ). Метеорологической дальностью видимости называется:

- в светлое время суток – наибольшее расстояние, с которого можно обнаружить (различить) на фоне неба вблизи горизонта абсолютно чёрный объект достаточно больших угловых размеров (более 15 угловых минут);
- в ночное время – расстояние, на котором при наблюдаемой прозрачности воздуха такой объект можно было бы обнаружить, если бы вместо ночи был день.

Как правило, МДВ измеряется визуально с помощью установленных или подобранных дневных и ночных ориентиров видимости, до которых известно расстояние. В этом случае стоит учитывать несколько нюансов:

- в закрытых местностях (леса, горы) или на открытом море не всегда есть возможность установить или выбрать подходящие ориентиры;
- дальность видимости может различаться в зависимости от направления наблюдения;
- визуальная дальность видимости зависит от остроты зрения конкретного наблюдателя.

Для решения вышеуказанных проблем применяются специальные приборы: трансмиссометры, нефелометры и другие датчики видимости. Общим недостатком этих инструментов является высокая стоимость и необходимость наличия оператора для их обслуживания и непосредственного проведения измерений. Более подробно про метеорологию и МДВ можно почитать в источнике [1].

Целью исследования является построение модели машинного обучения, способной автоматически с достаточной точностью определять дальность видимости, основываясь на других метеорологических показателях, таких как облачность, температура, влажность, погода и др.

В качестве модели машинного обучения была выбрана модель дерева решений, а в качестве набора данных для её обучения – архив метеорологических наблюдений посёлка Научный Городок, расположенного в городе Барнауле.

2. Деревья решений. Определение и история возникновения

Деревья решений (или решающие деревья) – это популярный и эффективных инструмент интеллектуального анализа данных. Он применяется в задачах регрессии, классификации и выделения наиболее важных атрибутов.

Идея создания модели дерева решений появилась в 1950-х года после исследований по прогнозированию человеческого поведения компьютерными системами. Решающую роль в этой области сыграли работы К. Ховеленда "Компьютерное моделирование мышления" [2] и Е. Ханта "Эксперименты по индукции" [3]. В дальнейшем большой вклад в развитие деревьев решений внесли Дж. Куинлен, разработавший алгоритм ID3 и его модификации C4.5 и C5.0, и Л. Брейман, создавший алгоритм CART и метод случайного леса.

3. Структура модели дерева решений

Деревья решений представляют из себя иерархическую древовидную структуру, состоящую из узлов и листьев и изображаемую в виде графа-дерева. В каждом узле находится решающее правило вида "Если ..., то ..." (например, "Если *погода ясная*, то дальность видимости *высокая*") и происходит проверка на удовлетворение примеров этим правилам по определённому атрибуту обучающей выборки. Эти правила генерируются автоматически в процессе обучения путём обобщения множества отдельных наблюдений.

Самый первый, верхний узел дерева называется его корнем. В простейшем случае в результате прохода множества примеров через корень оно разбивается на два подмножества, где одно из них удовлетворяет решающему правилу корня, а другое – нет. Затем каждое подмножество снова проходит через узел следующего уровня, где вновь разбивается на ещё на два подмножества. Алгоритм рекурсивно повторяется пока не будет достигнут некоторый критерий его остановки.

В результате в последнем узле проверка и разбиение не производятся и узел становится листом. Лист представляет из себя решение для всех попавших в него примеров. В случае задачи классификации – это класс, а в случае задачи регрессии – некоторый интервал целевой переменной.

Поскольку для попадания в конкретный лист пример должен удовлетворять всем правилам на пути к этому листу и в дереве существует лишь единственный путь от корня к листу, то каждый пример может попасть только в один лист. Это обеспечивает единственность решения.

4. Выбор атрибута разбиения

Для каждого решающего правила выбирается атрибут, по которому производится разбиение множества, попавшего в узел. Лучший атрибут должен разбивать множество примеров в узле так, чтобы получаемые подмножества содержали в себе наблюдения с метками только одного класса или были максимально приближены к этому. Для выбора такого атрибута наиболее часто применяются следующие два критерия.

Теоретико-информационный критерий основан на понятии информационной энтропии, которая вычисляется следующей формуле

$$H = -\sum_{i=1}^{n} \frac{N_i}{N} \log\left(\frac{N_i}{N}\right),$$

где n – число классов в исходном множестве; N_i – число наблюдений i-го класса; N – общее число примеров в множестве.

Энтропия показывает меру неоднородности множества по представленным классам. В случае, когда в исходном множестве каждый класс присутствует в равном количестве, энтропия максимальна и 1. Если же все примеры исходного множества относятся к одному классу, энтропия будет равна нулю.

Таким образом, лучшим объявляется атрибут, обеспечивающий максимальное снижение энтропии получаемого после разбиения множества относительно исходного. Однако на практике работают не с энтропией, а с величиной, обратной ей – информацией. В этом случае лучший атрибут обеспечивает максимальный прирост информации. Таким образом, задача нахождения лучшего атрибута разбиения сводится к задаче максимизации величины прироста информации.

Статистический критерий основан на использовании индекса Джини, показывающего то, насколько случайно выбранные пример обучающей выборки будет распознан неправильно, при условии, что целевые значения в этой выборке были взяты из определённого распределения. Фактически, этот индекс показывает расстояние между распределением целевых значений и распределением, полученным в ходе работы модели. Индекс Джини рассчитывается по формуле:

$$Gini(Q) = 1 - \sum_{i=1}^{n} p_i^2,$$

где Q – результирующее множество; n – число классов в результирующем множестве; p_i – вероятность i-го класса (относительная частота примеров i-го класса).

В случае, если исходное множество состоит из примеров одного класса, индекс Джини равен нулю. Если же каждый класс в исходном множестве встречается одинаковое количество раз, индекс равен 1. Таким образом, поиск лучшего атрибута сводится к минимизации индекса Джини.

5. Ограничение сложности

Теоретически, алгоритм обучения дерева решений будет работать до тех пор, пока в результате прохождения примерами узлов не будут получены подмножества, содержащие примеры только одного класса. В этом случае велика вероятность получения слишком большого дерева. Помимо того, что из-за своего размера оно будет не интерпретируемо, такое дерево скорее всего окажется переобученным. Оно просто подстроится под каждый пример из обучающего множества и, следовательно, будет хорошо работать только на обучающей выборке. Для решения этой проблемы применяются следующие эвристики:

- Ранняя остановка – обучение завершается по достижению какого-то показателя заданного значения. Например, при достижении доли правильно классифицированных примеров не менее 75%. Хоть этот метод и помогает в борьбе с разрастанием и переобучение дерева, он, как правило, сказывается на уменьшении точности.

- Ограничение глубины дерева – обучение останавливается по достижению указанной глубины (высоты) дерева. Данная эвристика также приводит к снижению точности дерева.

- Задание минимального допустимого числа примеров в узле – метод заключается в запрете на создание узлов с количеством примеров меньше заданного. Этот подход позволяет избежать создание тривиальных разбиений и малозначимых правил.

Как было сказано, все эти методы являются лишь эвристиками и не гарантируют улучшения результатов обучения. На практике экспертам приходится подбирать лучшее сочетание этих методов путём проб и ошибок. Дополнительную информацию по деревьям решения в доступном формате можно найти в источниках [4–6].

6. Существующие реализации

Деревья решений особенно часто применяются в машинном обучении и их составление давно автоматизировано. Так, в языке программирования Python существует свободно распространяемая библиотека для анализа данных scikit-learn, содержащая в том числе и реализацию модели дерева решений. Руководство по работе с библиотекой хорошо изложено в источнике [7].

7. Преимущества и недостатки деревьев решений

Преимущества метода дерева решений:

- Поскольку алгоритм работы решающих деревьев близок к алгоритму принятия решений человеком, то он является очень интерпретируемым.
- Возможность работы с разными типами переменных.
- Это инструмент позволяет выделить наиболее значимы параметры – их узлы будут находиться ближе к корню дерева.
- Деревья решений способны самостоятельно формировать правила в малознакомых специалисту областях.
- Метод является легко визуализируемым.
- Позволяет решать как задачи классификации, так и задачи регрессии.

Недостатки:

- Существует значительная вероятность ошибки в задачах классификации в случае большого количества классов при маленьком количестве наблюдений.
- Изменение параметров одного узла могут привести к лавинообразному изменению всей структуры дерева.
- Обучение модели может быть весьма трудоёмким, поскольку в каждом узле происходит перебор всех решений для выбора наилучшего.

8. Описание набора данных

Онлайн-сервис “RP5.ru” представляет из себя хранилище архивов погодных данных с большого количества метеорологических станций, расположенных по всей планете. В том числе, сервис предоставляет такой архив и по посёлку Научный Городок, расположенному в Ленинском районе г. Барнаула (Россия, Алтайский край). Архив представляет из себя Excel-таблицу, в которой каждый столбец отвечает за показатель погоды, а каждая строка — за запись этих показателей в некоторый момент времени.

Всего учитываются 28 погодных характеристик, каждой из которых соответствуют следующие названия столбцов, которые приведены в таблице 1.

Таблица 1

Погодные характеристики и соответствующие им названия атрибутов в наборе данных

Название столбца	Погодная характеристика
T	Температура воздуха (градусы Цельсия) на высоте 2 метра над поверхностью земли
Po	Атмосферное давление на уровне станции (миллиметры ртутного столба)
P	Атмосферное давление, приведенное к среднему уровню моря (миллиметры ртутного столба)
Pa	Барическая тенденция: изменение атмосферного давления за последние три часа (миллиметры ртутного столба)
U	Относительная влажность (%) на высоте 2 метра над поверхностью земли
DD	Направление ветра (румбы) на высоте 10-12 метров над земной поверхностью, осредненное за 10-минутный период, непосредственно предшествовавший сроку наблюдения
Ff	Скорость ветра на высоте 10-12 метров над земной поверхностью, осредненная за 10-минутный период, непосредственно предшествовавший сроку наблюдения (метры в секунду)
$ff10$	Максимальное значение порыва ветра на высоте 10-12 метров над земной поверхностью за 10-минутный период, непосредственно предшествующий сроку наблюдения (метры в секунду)
$ff3$	Максимальное значение порыва ветра на высоте 10-12 метров над земной поверхностью за период между сроками (метры в секунду)
N	Общая облачность
WW	Текущая погода, сообщаемая с метеорологической станции
$W1$	Прошедшая погода между сроками наблюдения 1
$W2$	Прошедшая погода между сроками наблюдения 2
Tn	Минимальная температура воздуха (градусы Цельсия) за прошедший период (не более 12 часов)
Tx	Максимальная температура воздуха (градусы Цельсия) за прошедший период (не более 12 часов)
Cl	Слоисто-кучевые, слоистые, кучевые и кучево-дождевые облака
Nh	Количество всех наблюдающихся облаков Cl или, при отсутствии облаков Cl, количество всех наблюдающихся облаков Cm
H	Высота основания самых низких облаков (м)
Cm	Высококучевые, высокослоистые и слоисто-дождевые облака
Ch	Перистые, перисто-кучевые и перисто-слоистые облака
VV	Горизонтальная дальность видимости (км)
Td	Температура точки росы на высоте 2 метра над поверхностью земли (градусы Цельсия)
RRR	Количество выпавших осадков (миллиметры)
tR	Период времени, за который накоплено указанное количество осадков (часы)
E	Состояние поверхности почвы без снега или измеримого ледяного покрова

	Продолжение таблицы 1
Tg	Минимальная температура поверхности за ночь (градусы Цельсия)
E'	Состояние поверхности почвы со снегом или измеримым ледяным покровом
sss	Высота снежного покрова (см)

Наблюдения записываются в архив ежедневно каждые три часа (по 8 раз в сутки). Первая запись была сделана 1-го января 2005-го года. Всего к четвёртому ноября 2022-го года архив содержит 51658 записей погодных условий.

9. Предобработка данных

Обработка набора данных и разработка модели машинного обучения проводились на языке программирования Python в интерактивной онлайн-среде Google Colab. Для предобработки и визуализации данных были использованы библиотеки Pandas и NumPy, а для построения модели дерева решений – библиотека scikit-learn. Ниже приведён фрагмент кода, в котором происходит подключение этих пакетов (первые 6 строк) и загрузка набора данных в формате CSV (последняя строка):

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.utils import resample
df = pd.read_csv("/content/drive/MyDrive/Coursework/meteo_data.csv",
sep=";")
```

Поскольку главной целью исследования является определение дальности видимости (столбце VV) на основе других показателей, первым делом из набора данных были удалены записи, в которых отсутствует значение столбца VV. Далее из таблицы были удалены следующие столбцы: Pa, $ff10$, $ff3$, $WW1$, $WW2$, Tn, Tx, Cl, Nh, Cm, Ch, RRR, tR, E, Tg, E', sss. Основными причинами для удаления была незначимость данных показателей и большое количество наблюдений, в которых отсутствовали их значения. Таким образом, в наборе данных остались следующие столбцы: температура (T), атмосферное давление (Po, P), относительная влажность воздуха (U), направление ветра (DD), скорость ветра (Ff), общая облачность (N), текущая погода (WW), дальность видимости (VV) и температура точки росы (Td). Описанным шагам соответствует следующий фрагмент кода:

```
df = df[~df['VV'].isnull()]
df = df[['T', 'Po', 'P', 'U', 'DD', 'Ff', 'N', 'WW', 'VV', 'Td']]
```

Следующим этапом подготовки данных была оцифровка показателей направления ветра (столбец DD), общей облачности (столбец N) и текущей погоды (столбец WW). Словесные описания направления ветра были заменены на азимуты, соответствующие этим направлениям. Так, например, направление “с востока” было заменено на 90, “с юго-востока” — на 135, а “с юго-юго-востока” на – 157,5. В таблице 2 отображены изначальные и оцифрованные значения общей облачности.

Таблица 2

Изначальные и оцифрованные значения столбца N

Изначальное значение	Оцифрованное значение
Облаков нет	0
10% или менее, но не 0	5
20-30%	25
40%	40
50%	50
60%	60
70-80%	75
90 и более, но не 100%	95
100%	100
Небо не видно из-за тумана и/или других метеорологических явлений	100

Ниже приведён код, отвечающий за оцифровку столбцов DD и N:

```
wind_directions = {"с севера": 360, "с северо-северо-востока": 22.5,
"с северо-востока": 45, "с востоко-северо-востока": 67.5, "с востока": 90,
"с востоко-юго-востока": 112.5, "с юго-востока": 135, "с юго-юго-востока":
157.5, "с юга": 180, "с юго-юго-запада": 202.5, "с юго-запада": 225, "с
западо-юго-запада": 247.5, "с запада": 270, "с западо-северо-запада": 292.5,
"с северо-запада": 315, "с северо-северо-запада": 337.5, "Штиль": 0}

for wind_dir_str, wind_dir_num in wind_directions.items():
    df.loc[df['DD'].str.contains(wind_dir_str, na=False), 'DD'] = float(wind_dir_num)
df['DD'] = df['DD'].astype(float)

df = df[~df['N'].isnull()]
df.loc[df['N'].str.contains("Облаков нет", na=False), 'N'] = 0.0
df.loc[df['N'].str.contains("10% или менее, но не 0", na=False), 'N'] = 5.0
df.loc[df['N'].str.contains("20--30%", na=False), 'N'] = 25.0
df.loc[df['N'].str.contains("40%", na=False), 'N'] = 40.0
df.loc[df['N'].str.contains("50%", na=False), 'N'] = 50.0
df.loc[df['N'].str.contains("60%", na=False), 'N'] = 60.0
df.loc[df['N'].str.contains("70 -- 80%", na=False), 'N'] = 75.0
df.loc[df['N'].str.contains("90 или более, но не 100%", na=False), 'N'] = 95.0
df.loc[df['N'].str.contains("100%", na=False), 'N'] = 100.0
df.loc[df['N'].str.contains("Небо не видно из-за тумана и/или других
метеорологических явлений.", na=False), 'N'] = 100.0
df['N'] = df['N'].astype(float)
```

Для преобразования столбца WW (текущая погода) в цифровой формат был применён следующий метод. Сначала были выбраны ключевые слова, используемые для описания погоды (“туман”, “дождь”, “слабый”, “сильный”, “снег”, “град”, “поземок” и др.), и в таблицу были добавлены соответствующие им столбцы. Далее, если ключевое слово встречалось в словесном описании погоды (столбец WW), то значению в соответствующем этому ключевому слову столбце присваивалась 1, иначе – 0. За оцифровку столбца WW производилось с помощью следующего фрагмента кода:

```
new_columns = ["fog", "smoke", "rain", "snow", 'shower', 'weak',
```

```
'medium','heavy', 'continuous', "snow_groats", 'hail', 'thunder',
'silent_lightning', 'drifting_snow', 'storm', 'disappear_clouds',
'appear_clouds', 'diamond_dust', 'mist', 'snowstorm', 'drizzle',
'snow_crystals', 'snow_grains']

for column_name in new_columns:
    df[column_name] = 0
df.loc[df['WW'].str.contains('дым', case=False), 'smoke'] = 1
df.loc[df['WW'].str.contains('дожд', case=False), 'rain'] = 1
df.loc[df['WW'].str.contains('снег', case=False), 'snow'] = 1

df.loc[df['WW'].str.contains('ливн', case=False), 'shower'] = 1
df.loc[df['WW'].str.contains('слаб', case=False), 'weak'] = 1
df.loc[df['WW'].str.contains('слаб', case=False), 'weak'] = 1
df.loc[df['WW'].str.contains('умерен', case=False), 'medium'] = 1
df.loc[df['WW'].str.contains('сильн', case=False), 'heavy'] = 1
df.loc[df['WW'].str.contains('непр', case=False), 'continuous'] = 1
df.loc[df['WW'].str.contains('крупа', case=False), 'snow_groats'] = 1
df.loc[df['WW'].str.contains('град', case=False), 'hail'] = 1
df.loc[df['WW'].str.contains('гроз', case=False), 'thunder'] = 1
df.loc[df['WW'].str.contains('молния', case=False), 'silent_lightning'] = 1
df.loc[df['WW'].str.contains('поземок', case=False), 'drifting_snow'] = 1
df.loc[df['WW'].str.contains('буря', case=False), 'storm'] = 1
df.loc[df['WW'].str.contains('образовывал', case=False), 'appear_clouds'] = 1
df.loc[df['WW'].str.contains('алмаз', case=False), 'diamond_dust'] = 1
df.loc[df['WW'].str.contains('мгла', case=False), 'mist'] = 1
df.loc[df['WW'].str.contains('метель', case=False), 'snowstorm'] = 1
df.loc[df['WW'].str.contains('морось', case=False), 'drizzle'] = 1
df.loc[df['WW'].str.contains('кристаллы', case=False), 'snow_crystals'] = 1
df.loc[df['WW'].str.contains('зерна', case=False), 'snow_grains'] = 1
df = df.drop(columns='WW')
```

В связи с тем, что значения целевого параметра – дальности видимости – были распределены крайне неравномерно, было принято решение объединить некоторые значения. Так, значения 0.05, 0.1, 0.2, 0.4, 0.5, 0.7 и 1 были объединены в значение 1, а значения 10, 20 и 50 – в значение 10. Изначальное распределение значений дальности видимости приведено в таблице 3.

Таблица 3

Исходные значения дальности видимости и их количестве в наборе данных

Значение	Количество
менее 0.05	2
0.05	32
менее 0.1	2
0.1	3
0.2	87
0.4	1
0.5	263
0.7	1
1	170

Продолжение таблицы 3	
2	623
4	5003
10	45257
20	37

Получившееся распределение указано в таблице 4.

Таблица 4

Распределение значений дальности видимости после объединения

Значение	Количество
1	561
2	623
4	5003
10	45314

Далее из таблицы были удалены все записи, в которых хотя бы один из оставшихся столбцов не имел значения. В результате всех действий размер таблицы сократился до 51391 строк. Распределение значений столбца VV в итоговом датасете отображено в таблице 5.

Таблица 5

Распределение значений дальности видимости в итоговом наборе данных

Значение	Количество
1	551
2	619
4	4995
10	45226

Ниже приведён фрагмент кода, в котором происходит объединение значений столбца VV:

```
df.loc[( (df['VV'] == 'менее 0.05') | (df['VV'] == '0.05') |
(df['VV'] == 'менее 0.1') | (df['VV'] == '0.1') | (df['VV'] ==
'0.2') | (df['VV'] == '0.4') | (df['VV'] == '0.5') |
(df['VV'] == '0.7') | (df['VV'] == '1')), 'VV' ] = 1
df.loc[( (df['VV']=='10') | (df['VV']=='20') |
(df['VV']=='50')) , 'VV'] = 10
df['VV'] = df['VV'].astype(float)
```

Как видно из таблицы, набор данные всё ещё крайне несбалансирован, а, значит, при обучении модели будет иметь место явный “перекос” в сторону мажоритарных классов “4” и, особенно, “10”. Для устранения этой проблемы было принято решение сократить количество записей каждого класса до 551. Таким образом, размер датасет изменился до 2204 строк. Процесс балансировки датасета отображён в следующем фрагменте кода:

```
df_vv_1 = df[df['VV'] == 1.0]
df_vv_2 = df[df['VV'] == 2.0]
df_vv_4 = df[df['VV'] == 4.0]
df_vv_10 = df[df['VV'] == 10.0]
```

```
df_vv_1_sample = df_vv_1
df_vv_2_sample = resample(df_vv_2, replace=False,
n_samples=551, random_state=42)
df_vv_4_sample = resample(df_vv_4, replace=False,
n_samples=551, random_state=42)
df_vv_10_sample = resample(df_vv_10, replace=False,
n_samples=551, random_state=42)

df_sample = pd.concat([df_vv_1_sample, df_vv_2_sample,
df_vv_4_sample, df_vv_10_sample])
```

10. Обучение модели

Для обучения и проверки модели выборка была разбита на обучающую и тестовую в соотношении 3:1, то есть получилось 1653 записи в обучающем множестве и 551 в тестовом.

При построении дерева без ограничений, как и ожидалось, модель получалась переобученной и чересчур сложной (глубина дерева достигала 22 узла). Для решения этой проблемы было добавлено ограничение на максимальную глубину дерева. Методом проб и ошибок было найдено оптимальное значение глубины – 5 узлов. В качестве критерия для выбора наилучшего атрибута разбиения был использован статистический критерий, основанный на индексе Джини. В этом случае точность на обучающей выборке составила 89,4%, а на тестовой – 87,5%.

Ниже приведён фрагмент кода, с помощью которого производилось построение, обучение модели и проверка модели.

```
X = df_sample.loc[ : , df_sample.columns != 'VV']
Y = df_sample['VV']

x_train, x_test, y_train, y_test = train_test_split(X, Y,
test_size=0.25, random_state=1)

clf = DecisionTreeClassifier(max_depth=5)
clf = clf.fit(x_train, y_train)
y_test_pred = clf.predict(x_test)
```

На рисунке 1 приведён граф, соответствующий полученному дереву.

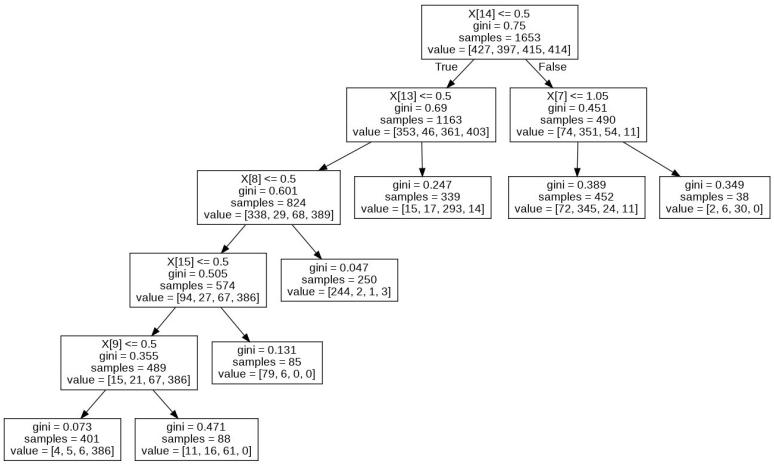

Рисунок 1. Граф, отображающий полученное дерево решений

Здесь в листьях указан индекс Джини, общее количество примеров, попавших в лист и количество примеров по каждому классу “1”, “2”, “4”, и “10” слева направо. В узлах, не являющихся листьями, также указан атрибут и условие разбиения. Здесь X[14] соответствует столбцу heavy (в нём значение равно 1, если в письменном описании погоды встречалось слово “сильный”, иначе - 0), X[13] — столбцу medium (1, если встречалось слово “умеренный”), X[7] — столбцу fog (1, если встречалось слово “туман”), X[8] — столбцу smoke (1, если встречались слова “дым” или “дымка”), X[15] — столбцу continuous (1, если встречалось слово “непрерывный”), а X[9] — столбцу rain (1, если встречалось слово “дождь”).

11. Уменьшение количества признаков

Также была попытка уменьшения размерности модели путём отбрасывания факторов, имеющих малую корреляцию с целевой переменной. Для этого сначала была выведена таблица корреляции:

Таблица 6

Корреляция переменной VV с остальными факторами

	VV		VV		VV
T	0.171191	Po	-0.021669	U	-0.317683
Ff	-0.143893	N	-0.262186	Td	0.089401
P	-0.037667	Fog	-0.245522	Smoke	-0.274866
Rain	-0.094114	Snow	-0.653044	Shower	-0.230052
Weak	-0.475367	Medium	-0.296767	Heavy	-0.306639
Continuous	-0.642097	Snow_groats	-0.047933	Hail	-0.033403
Thunder	0.000254	Silent_lightning	0.010169	Drifting_snow	0.041941
Storm	-0.002999	Disappear_clouds	0.008507	Appear_clouds	0.004546
Diamond_dust	0.016878	Mist	-0.096241	Snowstorm	-0.074103
Drizzle	-0.017274	Snow_crystals	-0.006570	Snow_grains	-0.034045

Как можно заметить, переменные Po, P, Hail, Tunder, Silent_lightning, Drifting_snow, Storm, Disappear_clouds, Appear_clouds, Diamond_dust, Drizzle, Snow_crystals и Snow_grains имеют корреляцию, по модулю меньше, чем 0.05, и, как можно предположить, не влияют на работу модели или вовсе вносят в неё лишний шум. Но после отбрасывания как всех этих признаков, так и их различных комбинаций, точность модели не увеличилась и составила в лучшем случае 84% на тестовом наборе данных.

12. Расширение набора данных

Как было написано выше, онлайн-сервис “RP5.ru” предоставляет архив погодных наблюдений со множества метеорологических станций. Для увеличения набора данных и улучшения точности модели были дополнительно загружены и обработаны архивы из следующих городов: Архангельск, Астрахань, Белгород, Брянск, Волгоград, Вологда, Воронеж, Иркутск, Краснодар, Курск, Липецк, Нижний Новгород, Оренбург, Саранск и Якутск.

После этого размер всего датасета составил 752903 наблюдения, а сбалансированного – 66508. Но, несмотря на такой большой прирост данных, точность модели стала значительно хуже – 73,8% на тестовой выборке. По всей видимости, связано это с тем, что на разных метеостанциях применяется разный подход к измерению дальности видимости. Где-то это значение указывают с точностью до километра, а где-то выделяют лишь значения в 1, 2, 4, 10 и 20 километров.

Для решения этой проблемы в набор данных были добавлены лишь записи с метеостанций, в которых разделение МДВ на категории было схоже с тем, что используется в Научном Городке (Астрахань, Брянск, Липецк, Воронеж и Нижний Новгород). При этом размер выборки после балансировки составил 24188 записей. Но и в этом случае не получилось добиться увеличения точности модели. Причинами такого поведения могут быть наличие каких-то неупомянутых в наборе данных факторов, которые меняются от города к городу и влияют на дальность видимости, и ограничения самой модели машинного обучения, не способной выдавать лучший результат на данном датасете.

13. Заключение

Таким образом, была получена работоспособная модель машинного обучения, способная автономно, без каких-либо дополнительных естественных или искусственных ориентиров с приемлемой точностью определять метеорологическую дальность видимость. Данный подход совмещает в себе плюсы как визуального, так и инструментального методов измерения МДВ, ведь он позволяет проводить автономное определение дальности видимости, не зависящее от остроты зрения наблюдателя и не нуждающееся в установке и обслуживании дополнительных приборов и ориентиров.

Помимо этого, модель позволила выделить главные метеорологические характеристики, влияющие на МДВ: тип осадков, их интенсивность и непрерывность, а также наличие тумана, дыма или дымки.

Список литературы

1. Хромов С.П., Петросянц М.А. Метеорология и климатология: учебник. — 7-е изд. — М. : Изд-во Моск. ун-та, 2006.

2. Hovland C.I. Computer simulation of thinking // American Psychologist. — 1960. — Vol. 15(11). — P. 687–693.

3. Hunt Earl B., Janet Marin, Philip J. Stone. Experiments in Induction. — New York : Academic Press, 1966.

4. Шахиди А. Loginom. — URL: https://loginom.ru/blog/decision-tree-p1. Дата обращения: 10.06.2022.

5. Синицин Ф. Решающие деревья // Академия Яндекса. Учебник по машинному обучению. — URL: https://academy.yandex.ru/handbook/ml/article/reshayushchiye-derevya. Дата обращения: 15.06.2022.

6. Толмачёв А., Классен Н. Для чего начинающим аналитикам нужны деревья решений // Яндекс Практикум. — URL: https://practicum.yandex.ru/blog/chto-takoe-derevo-reshenii-kak-ego-postroit. Дата обращения: 15.06.2022.

7. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. : Пер. с англ. — СПб. : ООО “Альфа-книга”, 2017.