Казанский (Приволжский) Федеральный Университет

Экзаменационный проект по курсу Data Science:

Titanic: Machine Learning from Disaster

Работу выполнил
Шайфутдинов Айдар
Высшая школа ИТИС
группа 11-401

Научный руководитель:
Новиков Петр Андреевич

Содержание

1. Описание проекта и набора данных

Был выбран один из проектов с ресурса kaggle.com - Titanic: Machine Learning from Disaster.

Задача заключалась в том, чтобы предсказать, выжил бы потенциальный пассажир судна в результате крушения корабля или нет, опираясь на имеющиеся данные.

Были доступны следующие данные о пассажирах:

- embarked - порт, в котором пассажир взошел на судно. Возможные значения: C = Cherbourg; Q = Queenstown; S = Southampton
- cabin - номер кабины (каюты)
- fare - цена билета
- ticket - номер билета
- parch - число родителей/детей на борту
- sibsp - число братьев/сестер/супругов на борту
- age - возраст
- sex - пол. Возможные значения: male, female
- name - имя
- pclass - социально-экономический статус пассажира. Возможные значения: 1 = Upper; 2 = Middle; 3 = Lower
- passengerId - идентификатор пассажира
- survival - булево значение, определяющее, выжил пассажир или нет. Возможные значения: 0 = Не выжил; 1 = Выжил

Видно, что эта проблема является проблемой **классификации с учителем** (supervised classification problem), так как диапазон значений предсказаний ограничен и есть тренировочный набор данных с известными значениями предсказываемой характеристики (target feature).

2. Преобразования данных (feature selection/scaling)

Как известно, для качества работы алгоритмов машинного обучения большое значение имеет **выбор атрибутов** (feature selection) и их **нормализация** (feature scaling).

Выбор атрибутов:

Для начала был проведен первичный анализ атрибутов и выявлены очевидно бесполезные для решения данной задачи, то есть те, от которых вряд ли могло зависеть выживание пассажиров. Это атрибуты *passengerId*, *name*, *ticket*.

Далее для атрибутов *embarked* и *cabin* был проведен точечный анализ с использованием инструментов графического представления данных из python-библиотеки matplotlib; оказалось, что эти атрибуты также не имеют особого значения для качественного предсказания.

Все остальные атрибуты, в силу здравого смысла, имеют значение для качественного предсказания, но некоторые из них было решено немного модифицировать.

Вместо строкового атрибута *sex* был введен булевый атрибут *female*, который может принимать два значения: 1 = если пассажир женского пола, 0 = если мужского.

Вместо того, чтобы использовать оба атрибута *parch* и *sibsp*, было решено объединить их в один атрибут *family*, который может принимать два значения: 1 = если на борту есть хотя бы один член семьи; 0 = если нет ни одного.

В итоге, для решения задачи были отобраны следующие атрибуты:
pclass, age, fare, female, family.

Нормализация:

Было написано несколько вспомогательных функций для нормализации атрибутов, значения которых лежали за пределами диапазона [0, 1]. Это атрибуты *pclass*, *age* и *fare*.

3. Используемые методы машинного обучения

Для решения данной проблемы были последовательно использованы следующие методы машинного обучения.

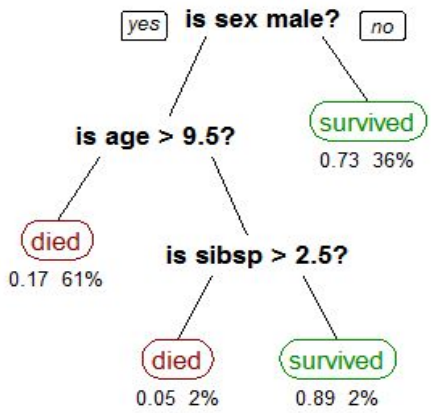
*Метод **k** ближайших соседей (k-NN):*

Пожалуй, самый простой из известных алгоритмов машинного обучения, который можно использовать для решения проблем классификации. Суть метода заключается в том, что новый объект присваивается тому классу, который является наиболее распространённым среди соседей данного элемента, где число соседей определяется параметром **k**. То есть этот метод не пытается построить некую обобщенную внутреннюю модель для предсказаний, а просто хранит и использует данные из обучающей выборки.

Дерево принятия решений (Decision tree):

Алгоритм, получивший широкое распространение в математике и программировании, может также использоваться в машинном обучении для решения проблем классификации. Рассматривается знакомое нам из теории графов дерево, где в узлах представлены атрибуты, от значений которых зависит выбор следующего узла; а в листьях дерева соответственно лежат значения целевого атрибута, которые и требуется предсказать.

Для рассматриваемой проблемы классификации можно визуализировать очень простое дерево решений следующим образом:



Случайный лес (Random forest):

Алгоритм машинного обучения, суть которого заключается в использовании множества *деревьев принятий решений*, рассмотренных ранее. Каждое дерево решений строится по определенной подвыборке из обучающей выборки. Для классификации объектов проводится голосование: каждое дерево из множества построенных деревьев относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев. В реализации данного алгоритма очень важно не допустить наличия идентичных деревьев, так как это может серьезно понизить точность предсказаний.

Искусственные нейронные сети (Artificial neural networks):

Искусственная нейронная сеть - это математическая модель, построенная по принципу организации и функционирования биологических нейронных сетей (например, мозга человека). Типичная нейронная сеть представляет собой некое множество нейронов, связанных между собой. Обычно у нейронной сети выделяют несколько слоев: Входной слой (input layer), какое-то количество скрытых слоев (hidden layers) и выходной слой (output layer). На входной слой нейронной сети поступают атрибуты в своем первоначальном представлении, нейроны внутренних слоев преобразуют эти атрибуты в новые, производные, атрибуты, используя которые выходной слой и делает предсказания. В качестве функции активации нейрона (activation function) была использована Сигмоида (Sigmoid function).

4. Анализ результатов

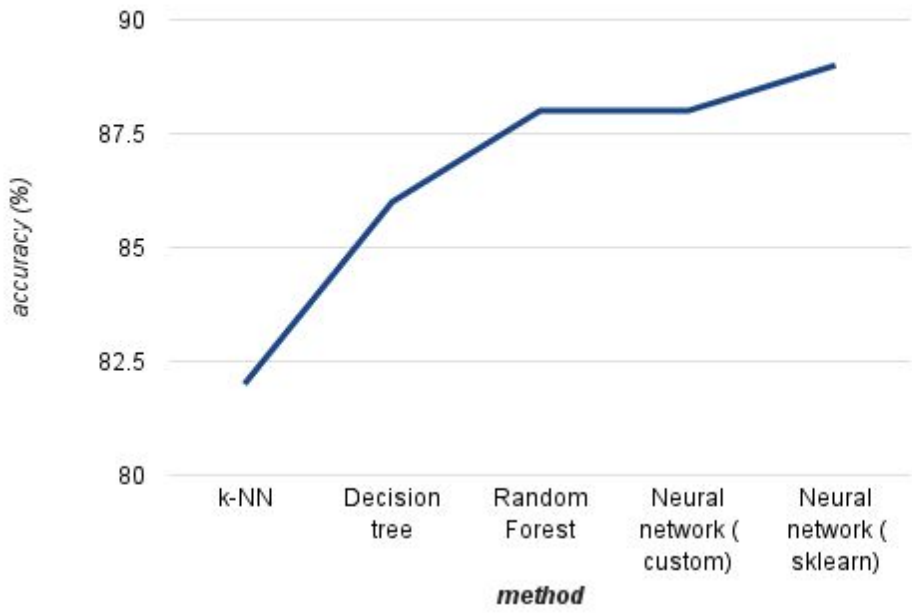
Для тестирования алгоритмов было написал несколько вспомогательных функций. Ниже представлен сравнительный анализ работы использованных алгоритмов. Были использованы следующие характеристики:

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)

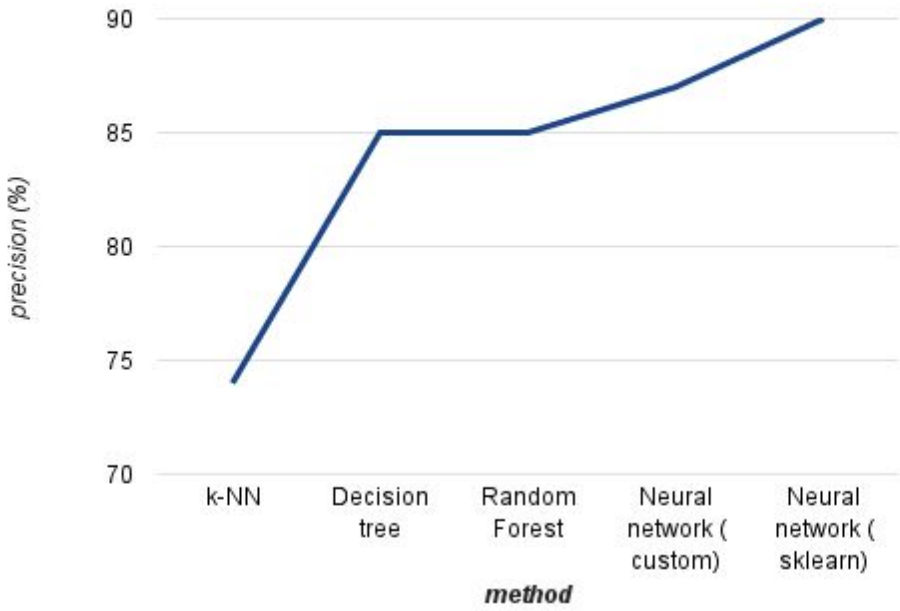
, где

- True positive (TP) - верное предсказание положительного результата
- False positive (FP) - ошибочное предсказание положительного результата
- True negative (TN) - верное предсказание отрицательного результата
- False negative (FN) - ошибочное предсказание отрицательного результата

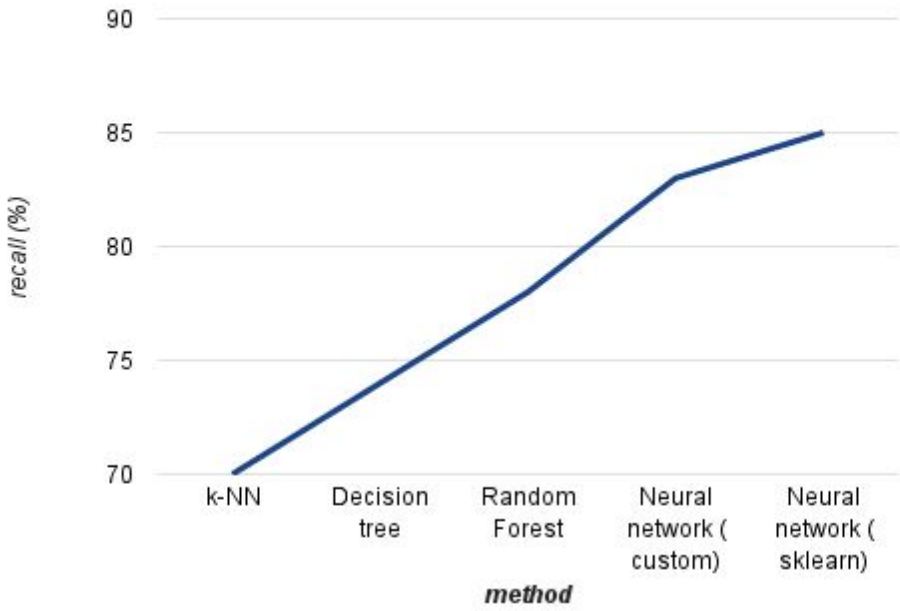
Accuracy:



Precision:



Recall:



5. Выводы

В процессе работы над данным проектом было получено представление о полном цикле решения задач машинного обучения - от анализа атрибутов до применения различных алгоритмов. Также было получено понимание всей важности качественного выбора атрибутов и их нормализации.

Если же говорить о различных методах машинного обучения, то лучше всего показали себя нейронные сети, а хуже всего - метод **k** ближайших соседей. И это вполне логично, так как нейронные сети, по сути, являются неким собирательным образом своих предшественников. Они не просто учатся на входных атрибутах, они многократно модифицируют их и учатся уже на новых и так по кругу.

Подводя итог, можно сказать, что работа над проектом была очень интересной, познавательной и продуктивной.

GitHub-репозиторий проекта