ООО «Центр онлайн-обучения Нетология-групп»

Дипломная работа на тему:

**Churn for Bank Customers. Predict customer churn in a bank.**

**Предсказание клиентского оттока банковской сферы.**

Кавешникова К.Е.

Поток DA-11

2021

Оглавление

[Цель дипломной работы 3](#_Toc70345590)

[Обоснование выбора Random Forest 3](#_Toc70345591)

[Старт работы с Random Forest 3](#_Toc70345592)

[Базовая модель 4](#_Toc70345593)

[Метод главных компонент 6](#_Toc70345594)

[Точность модели (визуализация), выводы 7](#_Toc70345595)

## Цель дипломной работы

Обработка и подготовка исходных данных и последующее построение обучающей модели на основе алгоритма Random Forest. Данные взяты с [Kaggle](https://www.kaggle.com/mathchi/churn-for-bank-customers/tasks?taskId=1479).

Скрипт уже содержит поясняющие комментарии.

## Обоснование выбора Random Forest

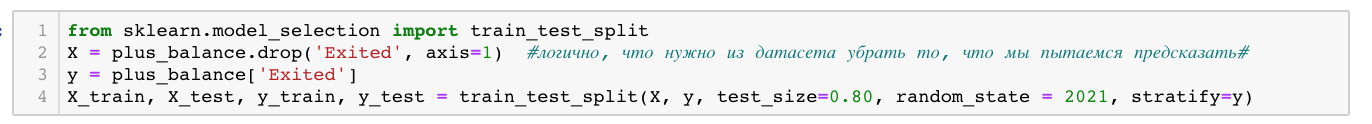
Random Forest хорош для не нормализованных данных, многие алгоритмы машинного обучения (ML) дают плохие результаты и данные предварительно надо обрабатывать. Для RF этот шаг можно попробовать опустить и быстро получить работающий прототип.

Random Forest практически не чувствителен к выбросам в данных из-за случайного сэмплирования, не чувствителен к масштабированию (и вообще к любым монотонным преобразованиям) значений признаков, способен эффективно обрабатывать данные с большим числом признаков и классов, одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки.

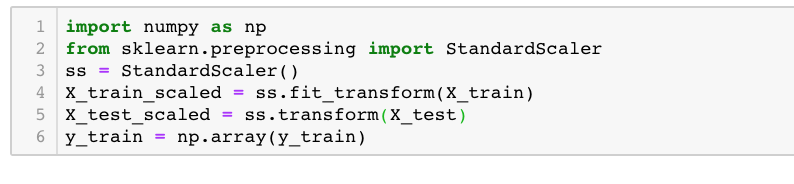
Из некоторых минусов можно выделить то, что обучившись на одних и тех-же данных несколько раз — подход будет предсказывать немного разные значения с разными вероятностями. От этого не уйти, т.к. выборка ансамбля решающих деревьев происходит случайным образом.

## Старт работы с Random Forest

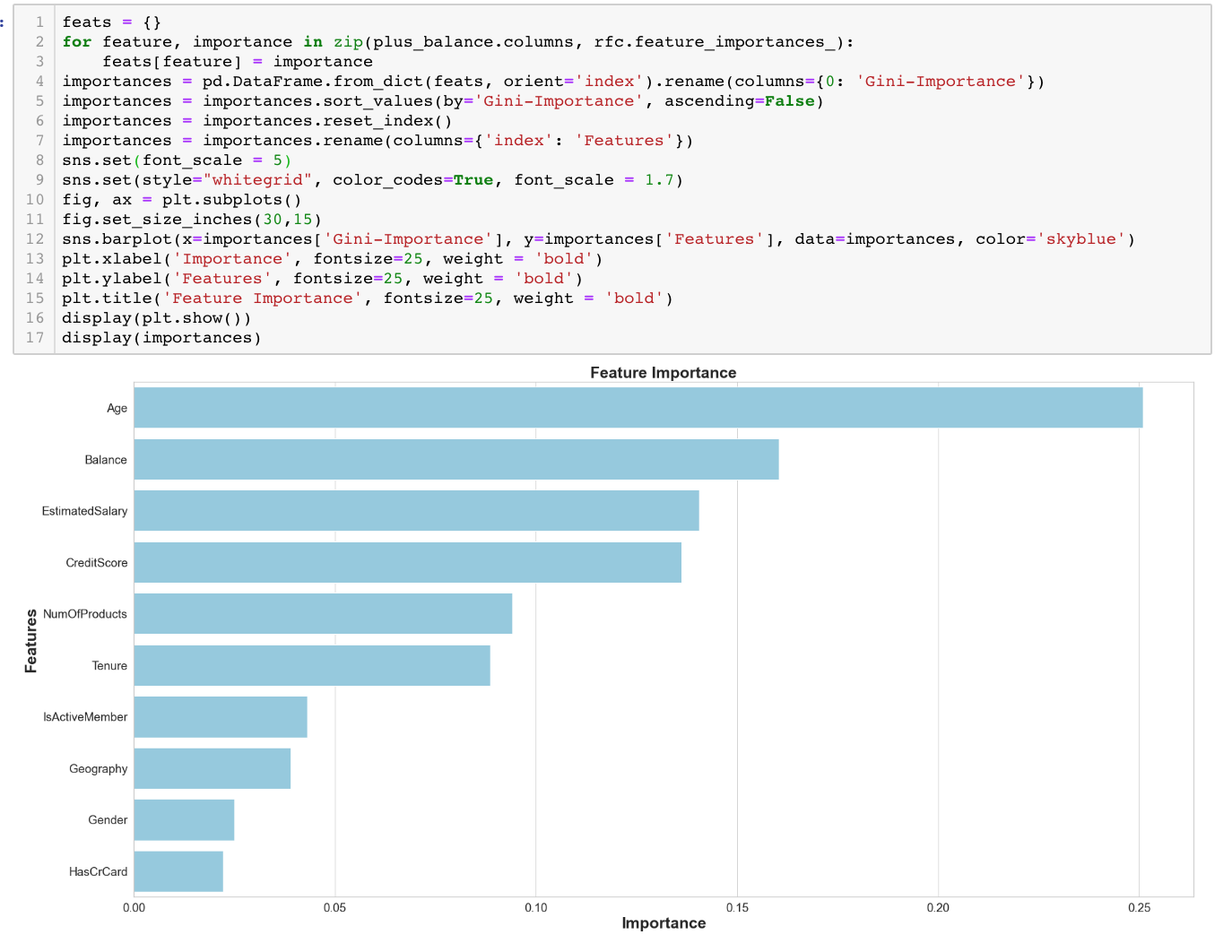
В данной работе была использована базовая модель Random Forest (далее мы будем сокращать название до RF). После обработки данных можно уже приступать к разделению набора данных на учебные и проверочные. Чтобы дать модели как можно больше учебных данных, мы разделим набор по следующему принципу: 80% — учебные данные и 20% — проверочные. Далее устанавливаем stratify=y для обеспечения того, чтобы и в учебном, и в проверочном наборах данных присутствовало бы то же соотношение 0 и 1, что и в исходном наборе данных.

  
Прежде чем приступать к моделированию, нужно выполнить «центровку» и «стандартизацию» данных путём их [масштабирования](https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02). Масштабирование выполняется из-за того, что разные величины выражены в разных единицах измерения. Эта процедура позволяет организовать «честную схватку» между признаками при определении их важности. Для базовой модели это не нужно, так как в деревьях не используется взвешивание признаков, но для РСА это полезно.

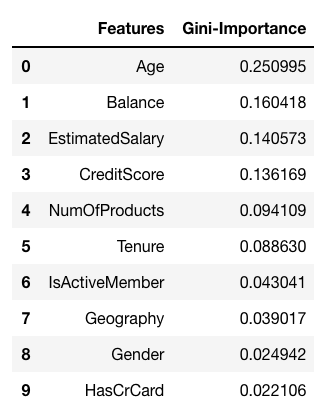
Кроме того, мы конвертируем y\_train из типа данных Pandas Series в массив NumPy для того чтобы позже модель смогла бы работать с соответствующими целевыми показателями.



## Базовая модель

В ней применяется только алгоритм Random Forest. Она использует все признаки и настроена с использованием значений, задаваемых по умолчанию (подробности об этих настройках можно найти к [документации](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)). Сначала инициализируем модель. После этого обучим её на масштабированных данных.   
Если нам интересно узнать о том, какие признаки являются самыми важными для RF-модели, мы можем визуализировать и квантифицировать показатели важности признаков, обратившись к атрибуту feature\_importances\_:  


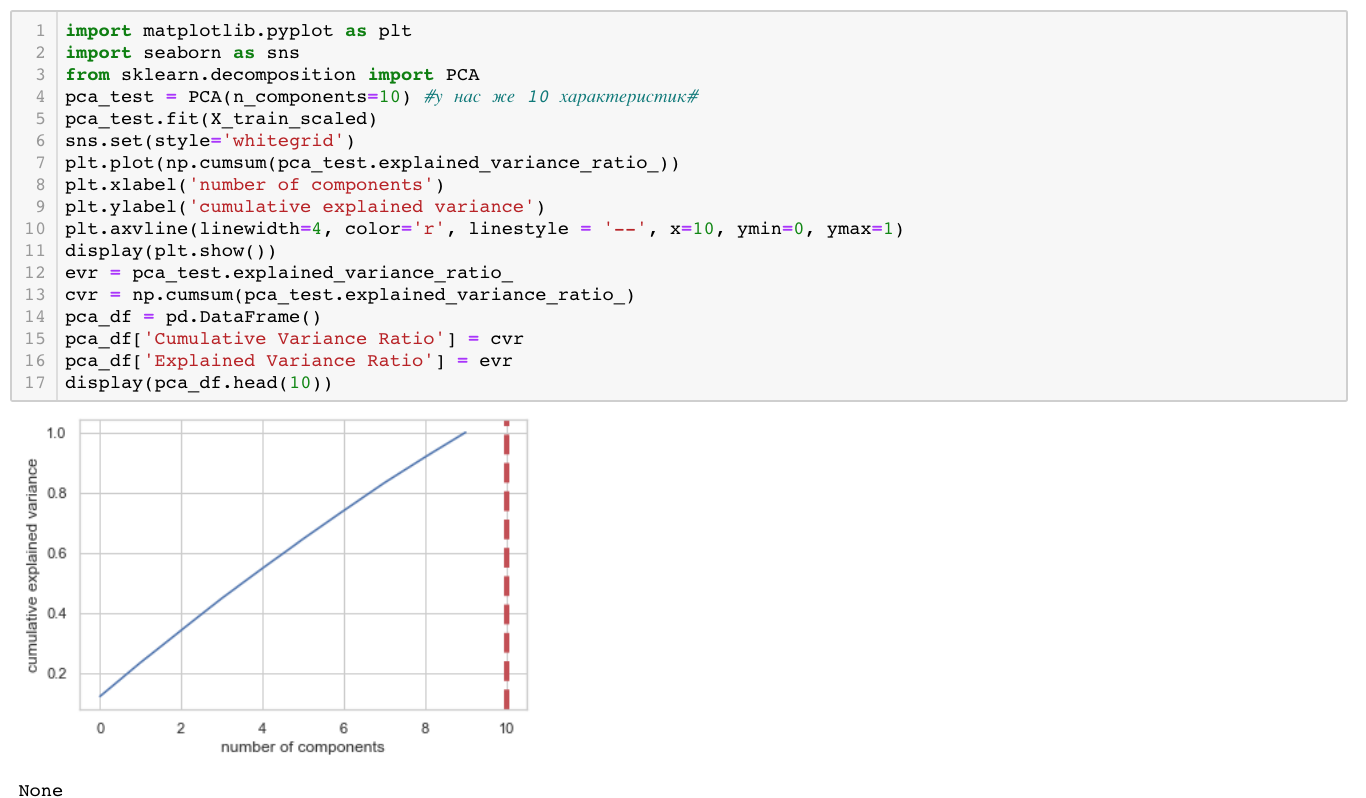
*Визуализация «важности» признаков*

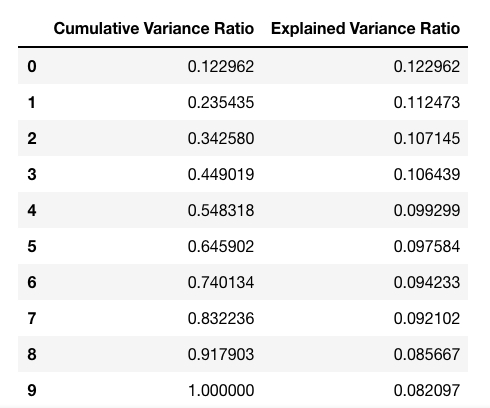


*Показатели важности признаков*

## Метод главных компонент

А как можно улучшить базовую RF-модель? С использованием методики снижения размерности пространства признаков можно представить исходный набор данных через меньшее количество переменных и при этом снизить объём вычислительных ресурсов, необходимых для обеспечения работы модели. Используя PCA, можно изучить кумулятивную выборочную дисперсию этих признаков для того чтобы понять то, какие признаки объясняют большую часть дисперсии в данных.  
Инициализируем объект PCA (pca\_test), указывая количество компонент (признаков), которые нужно рассмотреть. Устанавливаем этот показатель в 10 для того чтобы увидеть объяснённую дисперсию всех сгенерированных компонент до принятия решения, сколько компонент нужно оставить. Затем передаём в pca\_test масштабированные данные X\_train, пользуясь методом pca\_test.fit(). После этого визуализируем данные.

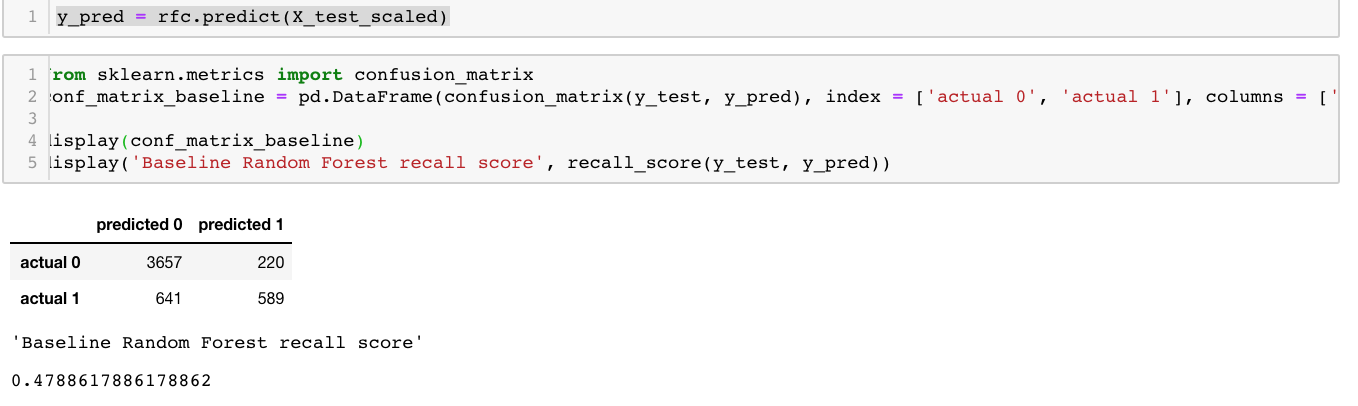
  
*Рост дисперсии не останавливается вплоть до 10 компоненты.*



*Этот датафрейм содержит такие показатели, как Cumulative Variance Ratio (кумулятивный размер объяснённой дисперсии данных) и Explained Variance Ratio (вклад каждой компоненты в общий объём объяснённой дисперсии)*

Если взглянуть на вышеприведённый датафрейм, то окажется, что все 10 компонентов объясняют 100% дисперсии данных, а сокращение компонентов хотя бы на 30% снизит дисперсию данных до 74%. Но метод РСА есть смысл применять при очень большом датасете с множеством компонент, это позволит ускорить работу модели.

## Точность модели (визуализация), выводы



Как можно увидеть точность модели невысока. Благодаря предварительной работе с данными точность удалось улучшить на 2 процента. При большом желании улучшить предсказания нашей модели можно воспользоваться оптимизацией [гиперпараметров](https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74). Гиперпараметры можно рассматривать как что-то вроде «настроек» модели. Настройки, которые отлично подходят для одного набора данных, для другого не подойдут — поэтому и нужно заниматься их оптимизацией. Например, алгоритм RandomizedSearchCV и многие многие другие.

Иногда RF-модель, в которой используется метод главных компонент и широкомасштабная оптимизация гиперпараметров, может работать не так хорошо, как самая обыкновенная модель со стандартными настройками.