**Документация, содержащая информацию, необходимую для установки и эксплуатации экземпляра программного обеспечения, предоставленного для проведения экспертной проверки.**

**Оглавление**

[1 ОБЩИЕ ПОЛОЖЕНИЯ 3](#_heading=h.gjdgxs)

[1.1 Общие сведения о python3 распространяемой библиотеке fdwscoring 3](#_heading=h.30j0zll)

[1.2 Информация, необходимая для установки и настройки python3 распространяемой библиотеки fdwscoring 3](#_heading=h.3znysh7)

[1.3 Описание структуры 3](#_heading=h.tyjcwt)

[2 ПОДГОТОВКА К РАБОТЕ](#_heading=h.2s8eyo1) 7

[3 ОПИСАНИЕ РАБОТЫ](#_heading=h.17dp8vu) 7

[3.1 Установка и настройка ПО](#_heading=h.3rdcrjn) 7

[3.2 Установка обновлений](#_heading=h.26in1rg) 8

[3.3 Штатное функционирование](#_heading=h.lnxbz9) 8

[3.4 Резервное копирование и восстановление данных](#_heading=h.35nkun2) 8

[3.5 Проведение диагностики ИС](#_heading=h.1ksv4uv) 8

# ОБЩИЕ ПОЛОЖЕНИЯ

# 1.1 Общие сведения о python3 распространяемой библиотеке fdwscoring

# Программное обеспечение fdwscoring (Fordewind Scoring) представляет собой python3 распространяемую библиотеку для оценки кредитоспособности юридических лиц малого и среднего бизнеса и индивидуальных предпринимателей. Продукт позволяет проводить многосторонний скоринг заемщиков на основе финансовых показателей бухгалтерского учета и данных коммерческой бизнес-деятельности. Последнее, не ограничиваясь, включает в себя участие в гос. закупках, арбитражные дела, исполнительные производства и прочее. Библиотека содержит модели, отражающие актуальную практику оценки кредитного качества заемщиков малого и среднего бизнеса. Часть моделей дает возможность репрезентировать качество заемщика простой числовой метрикой. Для более комплексного описания заемщиков, имеются модели, позволяющие подсветить конкретные риск-факторы заемщиков.

# 1.2 Информация, необходимая для установки и настройки python3 распространяемой библиотеки fdwscoring

# Установка ПО включает в себя регистрацию аккаунта Заказчика на платформе GitLab и дальнейшее получение экземпляра python3 распространяемой библиотеки. Регистрация аккаунта выполняется администратором системы после подписания документов. Администратор системы создает аккаунт клиента и передает логин и пароль клиенту посредством электронной почты. Заказчику не требуется выполнять действия по настройке ПО, поскольку жизненный цикл системы предполагает однократную регистрацию аккаунта Заказчика на платформе GitLab и дальнейшую поддержку работы системы силами персонала Исполнителя.

# 1.3 Описание структуры

Программа обеспечивает выполнение следующих функций:

1. Комплексная оценка кредитоспособности юридических лиц малого и среднего бизнеса и индивидуальных предпринимателей.

2. Проведение скоринга надежности заемщика в рамках одной числовой метрики.

3. Оценка вероятности дефолта юридического лица.

4. Поиск и анализ потенциальных риск-факторов заемщика.

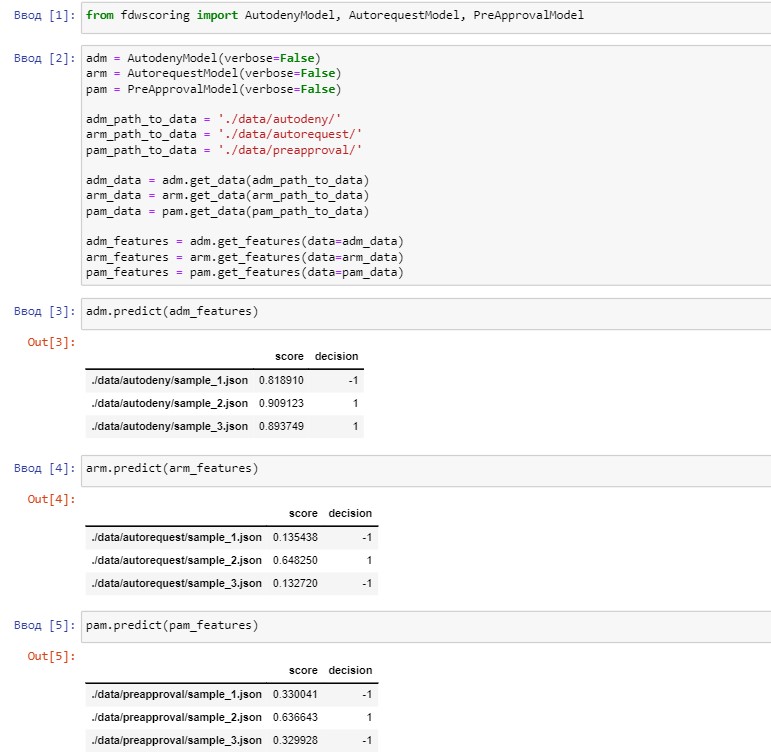
5. Оценка лимитов по займам, выдаваемых заемщикам.

# Модели анализа надежности заемщика

# Библиотека содержит триплет схожих моделей, каждая из которых позволяет оценить надежность заемщика: AutodenyModel, AutorequestModel и PreApprovalModel. Указанные модели построены на семантически разнородных данных и инкапсулируют актуальную практику финансового анализа юридических лиц малого и среднего бизнеса.

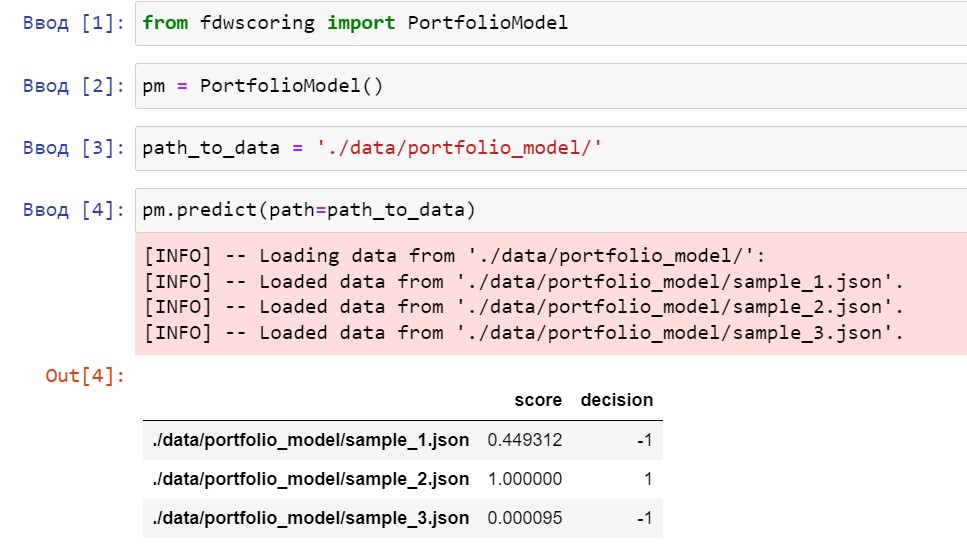
# Модель AutodenyModel статистически значимо оценивает отсутствие надежности заемщика в рамках числовой метрики от 0 до 1. Чем выше score (величина числовой метрики), тем менее надежен заемщик. В случае если score выше порогового значения, полученного в ходе оптимизации ML алгоритмов, модель рекомендует отказ заемщику (decision 1). В противном случае результат модели не является статистически значимым и модель не может дать обоснованную рекомендацию (decision -1).

# Модель PreApprovalModel работает в инвертированной к AutodenyModel логике. Чем выше score, тем более надежен заемщик. В случае если score выше порогового значения, модель статистически значимо рекомендует одобрить заявку (decision 1); в противном случае результат модели следует игнорировать (decision -1). Модель AutorequestModel схожа с PreApprovalModel и выдает статистически значимое решение по запросу внутренних документов, что является показателем хорошего качества заемщика. Пример использования моделей представлен ниже:



# Модель оценки вероятности дефолта юридических лиц

Для оценки вероятности дефолта юридических лиц, функционал ПО подразумевает использование модели PortfolioModel. Данная модель построена на объективных данных по дефолтам. Чем выше score модели, тем выше вероятность дефолта. Итоговым решением модели является статистически значимая оценка наступления дефолта: в случае если score выше порогового значения, модель рекомендует отказывать заемщику (decision 1); в противном случае результат модели не является статистически значимым (decision -1). Пример использования модели:



**Модели поиска и анализа риск-факторов заемщика**

Для поиска и анализа риск-факторов юридических лиц и индивидуальных предпринимателей в библиотеки предусмотрено наличие моделей StopFactorULModel и StopFactorIPModel, соответственно. Модели производят поиск риск-факторов, которые в итоговом выводе "обезличены", но семантически приблизительно равноценны. Каждый фактор может принимать три значения: 0 – статистически значимо не рискованный фактор; 1 – статистически значимо рискованный фактор; 42 – негативный риск-фактор, который, однако, не является полноценно статистически значимым. Модель анализирует найденные риск-факторы и принимает итоговое решение: 0, 1 или 42 с описанной выше семантикой. Пример использования моделей:



# Оценка лимитов выдаваемых займов

# Для оценки лимитов выдаваемых займов в библиотеке реализована модель LimitsModel. Указанная модель построена на квантилях лимитов, рассчитанных в рамках актуальных практик при работе с юридическими лицами малого и среднего бизнеса. Функционал модели позволяет рассчитать три значения лимита в млн. руб., которые соответствуют q25%, q50% и q75% квантилям. Рассчитываемые значения лимитов отражают мало рисковое значение лимита, значение комфортного лимита и высоко рисковое значение лимита, соответственно. Пример использования модели:



# ПОДГОТОВКА К РАБОТЕ

Для корректной работы с ПО необходима следующая конфигурация рабочего места пользователя:

· Windows

Требования к системе:

Windows 7, Windows 8, Windows 10 или более поздней версии.

· macOS

Требования к системе:

macOS 10.9 (64-разрядная версия) или более поздней версии.

· Linux

Требования к системе:

Ubuntu 18.04 (64-разрядная версия) или более поздней версии, Debian 10 или более поздней версии, openSUSE 15.2 или более поздней версии, Fedora Linux 32 или более поздней версии.

Установите ПО как python распространяемую библиотеку согласно инструкции по установки экземпляра ПО. По завершению установки библиотеки ПО готово к использованию.

# ОПИСАНИЕ РАБОТЫ

Для получения экземпляра ПО и дальнейшей эксплуатации системы, требуется регистрация аккаунта Заказчика на платформе GitLab. Регистрация аккаунта выполняется администратором системы после подписания документов. Администратор системы создает аккаунт клиента и передает логин пароль клиенту посредством электронной почты. После получения доступа к аккаунту платформы GitLab, Заказчик самостоятельно получает ПО согласно инструкции по установке экземпляра ПО. Полученное ПО может быть использовано Заказчиком в рамках импортируемой python библиотеки.

Для проведения Заказчиком скоринга необходимо наличие данных о заемщиках установленного формата. Сбор данных о заемщиках проводится Заказчиком самостоятельно согласно шаблонам, представленным в библиотеке. Заказчик может получить необходимые для проведения скоринга данные у Исполнителя в рамках отдельной дополнительно оплачиваемой услуги.

# 3.1 Установка и настройка ПО

Установка ПО включает в себя регистрацию аккаунта Заказчика на платформе GitLab и дальнейшее получение экземпляра python распространяемой библиотеки. Заказчику не требуется выполнять действия по настройке ПО, поскольку жизненный цикл системы предполагает однократную регистрацию аккаунта Заказчика на платформе GitLab и дальнейшую поддержку работы системы силами персонала Исполнителя.

# 3.2 Установка обновлений

Установка обновлений ПО подразумевает установку свежей версии python распространяемой библиотеки fdwscoring по средствам платформы GitLab. Все обновления устанавливаются Заказчиком согласно инструкции установки экземпляра ПО.

# 3.3 Штатное функционирование

ПО fdwscoring реализовано в архитектуре python распространяемой библиотеки. Штатное функционирование ПО осуществляется путем импортирования библиотеки во время активной сессии интерпретатора python или работы ядра интерактивной среды разработки, такой как, например, Jupyter Notebook. Взаимодействие пользователя ПО с библиотекой осуществляется с помощью верхнеуровневого API, представленного в руководстве по эксплуатации экземпляра ПО. Для проведения скоринга заемщиков необходимо наличие данных установленного образца. Сбор данных осуществляется Заказчиком самостоятельно или проводится Исполнителем по запросу Заказчика в рамках дополнительно оплачиваемой услуги.

# 3.4 Резервное копирование и восстановление данных

Структура и внутренние данные в ходе эксплуатации ПО не изменяются, поэтому резервное копирование системы со стороны Заказчика необязательно. В случае утраты экземпляра ПО, Заказчик может обратиться по каналам тех. поддержки для получения копии ПО. Заказчику рекомендуется регулярно проводить резервное копирование пользовательских данных, полученных в ходе эксплуатации системы. Резервное копирование репозиториев исходного кода производится Исполнителем согласно принятому в организации регламенту.

# 3.5 Проведение диагностики ИС

Диагностика ПО производится периодически силами персонала Исполнителя изолировано от экземпляра ПО Заказчика. Изменения и корректировки экземпляра ПО Заказчика возможны только путем обновления ПО после выхода новой версии библиотеки fdwscoring. В случае возникновения неисправностей при работе ПО, Заказчик может обратиться в техническую поддержку Исполнителя.

# 