Анализ контрагентов.

Для решения этой задачи было предоставлено 3 набора данных. Данные agents2021.csv включают в себя данные с agents2019.csv и agents2020.csv. То для рассмотрения была взята только база по контрагентам agents2021.csv.

Объем объектов совсем небольшой – всего 325 контрагентов.

Признаки – данные финансовой отчетности за 2016-2020 года, факты под номерами, и данные о задолженности 2019, 2020, 2021 (если я правильно поняла последние колонки).

Требуется: построить модель ML, которая должна рассчитывать возможность просрочки оплаты контрагентом , основываясь на показатели прошлых периодов. И если просрочка возможна, то на какой период.

Признаки, которые были добавлены:

На основании "Методоглогических указаний по проведению анализа финансового состояния организаций" (приказ ФСФО от 23 января 2001г. № 16) в качестве признаков рассчитаны коэффициенты платежеспособности и финансовой устойчивости.

Платежеспособность характеризует возможность организации своевременно расплачиваться по своим обязательствам.

Для отслеживания динамики изменения показателей финансовой отчетности, в качестве признаков были добавлены показатели:

темп роста, темп прироста, ускорение.

В ходе анализа не выявлено никаких явных признаков, которые бы способствовали более четкому разграничению - заплатит контрагент во время или нет.

Создание модели:

Для машинного обучение сформировано 2 варианта выборок по признакам.

Так как данных мало, то выборку смысла дробить нет.

Модель должна рассчитывать возможность просрочки основываясь на показателях прошлых периодов.

Для target_2019 возьмем года 2016-2018 - Проверять будем на target_2020 и годах 2017-2019

Для target_2020 возьмем года 2017-2019 - Проверять будем на target_2021 и годах 2018-2020

Т.е. у нас будет 3 обучающие выборки и 3 с верными значениями в нескольких вариациях

df_2019	Года для обучения 2016-2018	target_2019_01
df_2020	Года для обучения 2017-2019	target_2020_01
df_2021	Года для обучения 2018-2020	target_2021_01

target переменные созданы в трех вариантах:

target_2020_01 Бинарные переменные '0 нет – 1 есть' задолжности

target_2020_012 Есть задолженность и какой срок: 1(1-30), 2(свыше 31) или нет 0

target_2020_01234 Есть задолженность и какой срок: 1(1-30), 2(31-90), 3(31-365), 4(от 366) или нет 0

Так же были созданы такие же по признакам выборки, только по контрагентами, которые платят всегда во время, или у которых ежегодно просрочка.

Таких оказалось всего 88 объектов.

Так как расчет коэффициентов – это сложение, вычитание, деление, умножение между показателями финансовой отчетности, то возникло много признаков с высокой корреляцией между собой.

Для выяления таких признаков, которые можно удалить с наименьшими потерями, воспользовалась:

Feature selector - это инструмент для уменьшения размерности наборов данных машинного обучения.

Его функционал для определения объектов для удаления:

- Отсутствующие значения
- Одиночные уникальные значения
- Коллинеарные признаки
- Признаки нулевой важности
- Признаки с низкой важностью

Проверялись на корреляцию df_2019 и df_2020, так как на них строилось обучение. Признаки у них оказались разными.

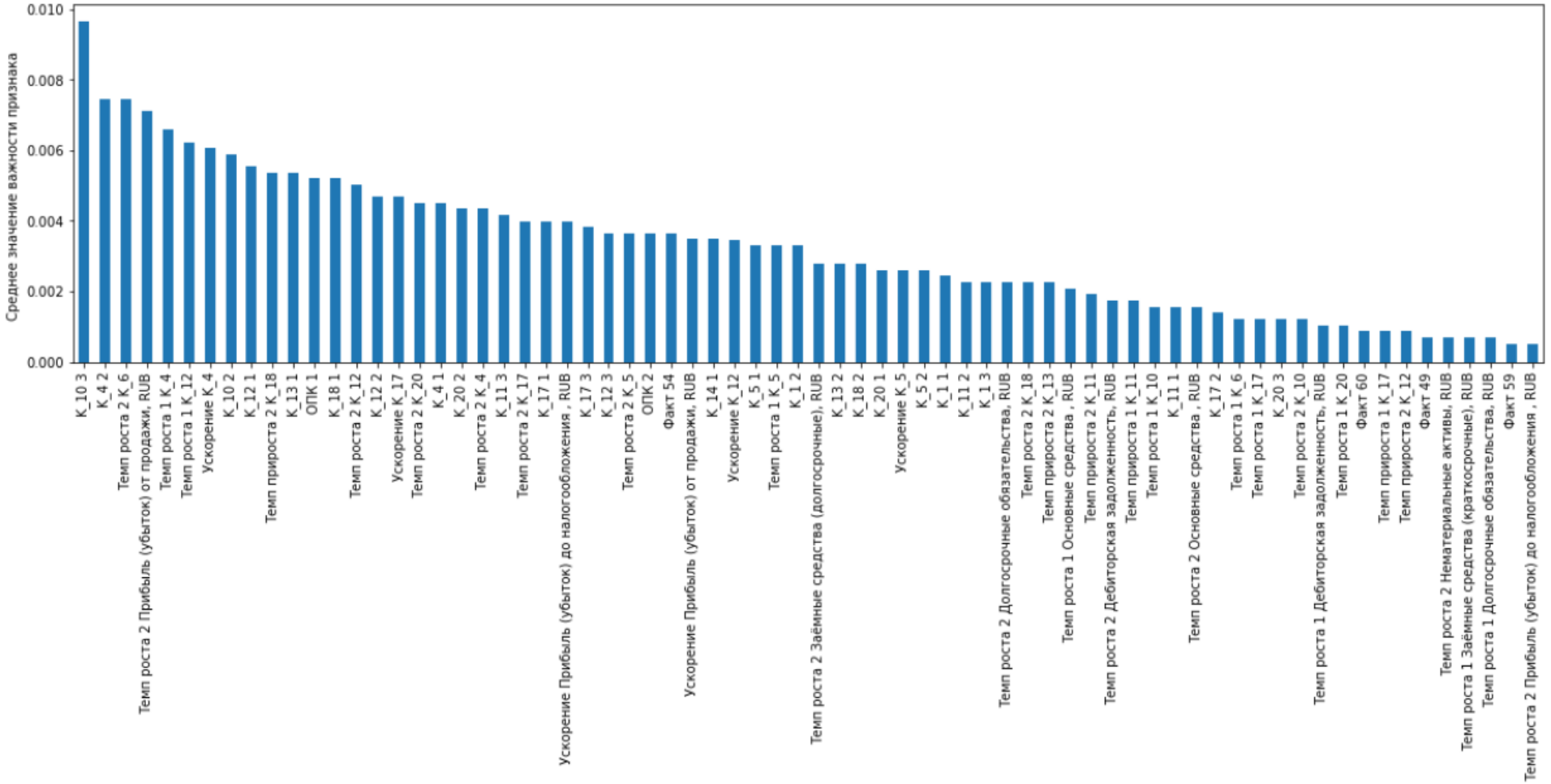
Следующий этап состоял в ранжировании признаков по степени важности. В результате были сформированы списки, на основании которых собирались датасеты для обучения.

Использовались методики:

- **Feature selector**
- **mutual_info_classif**
- **permutation_importance**
- **Recursive feature elimination**
- **LogisticRegression**
- **f_classif**

Вот таким образом было собрано несколько вариантов разделения признаков по степени важности. Признаки сохранены все. При обучении задается порог.

Это пример **permutation_importance на базе RandomForestClassifier**



ML обучение

Был проведен тест на разных алгоритмах

Линейные алгоритмы:

— Логистическая регрессия* / Logistic Regression (‘LR’)
— Линейный дискриминантный анализ / Linear Discriminant Analysis (‘LDA’)

Нелинейные алгоритмы:

— Метод k-ближайших соседей (классификация) / K-Neighbors Classifier (‘KNN’)
— Деревья принятия решений / Decision Tree Classifier (‘CART’)
— Наивный классификатор Байеса / Naive Bayes Classifier (‘NB’)
— Линейный метод опорных векторов (классификация) / Linear Support Vector Classification (‘LSVC’)
— Метод опорных векторов (классификация) / C-Support Vector Classification (‘SVC’)

Алгоритм искусственной нейронной сети:

— Многослойный персептрон / Multilayer Perceptrons (‘MLP’)

Ансамблевые алгоритмы:

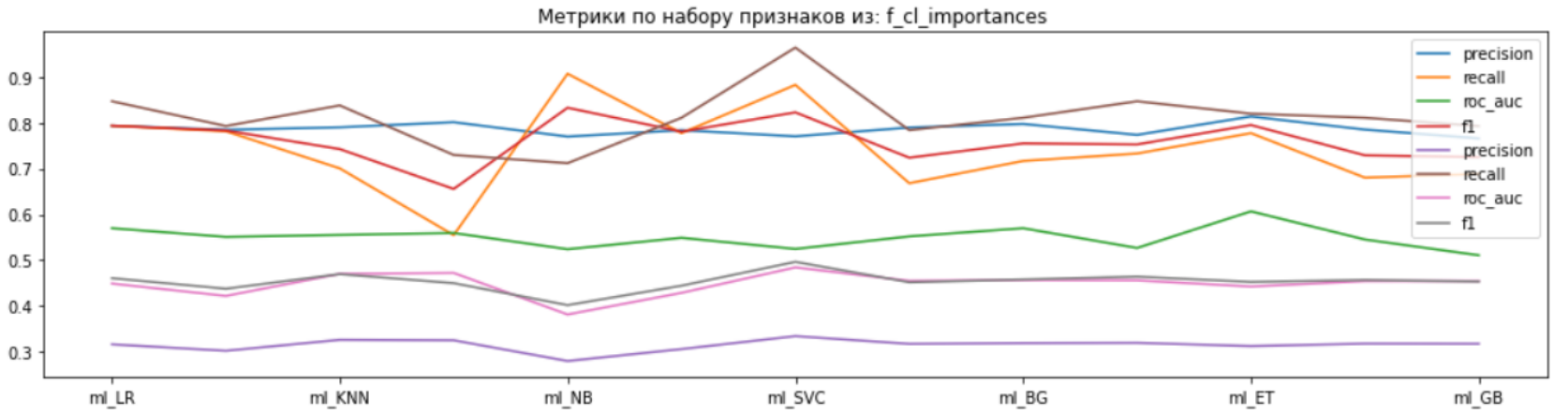
— Bagging (классификация) / Bagging Classifier (‘BG’) (Bagging = Bootstrap aggregating)
— Случайный лес (классификация) / Random Forest Classifier (‘RF’)
— Экстра-деревья (классификация) / Extra Trees Classifier (‘ET’)
— AdaBoost (классификация) / AdaBoost Classifier (‘AB’) (AdaBoost = Adaptive Boosting)
— Градиентный boosting (классификация) / Gradient Boosting Classifier (‘GB’)

	Предсказания на 2020					Предсказания на 2021				
Обучение на отборе 45 признаков из f_cl_importances										
	conf_matrix	precision	recall	roc_auc	f1	conf_matrix	precision	recall	roc_auc	f1
ml_LR	[[27, 51], [51, 196]]	0.79	0.79	0.57	0.79	[[11, 203], [17, 94]]	0.32	0.85	0.45	0.46
ml_LDA	[[25, 53], [54, 193]]	0.78	0.78	0.55	0.78	[[11, 203], [23, 88]]	0.30	0.79	0.42	0.44
ml_KNN	[[32, 46], [74, 173]]	0.79	0.70	0.56	0.74	[[22, 192], [18, 93]]	0.33	0.84	0.47	0.47
ml_CART	[[44, 34], [110, 137]]	0.80	0.55	0.56	0.66	[[46, 168], [30, 81]]	0.33	0.73	0.47	0.45
ml_NB	[[11, 67], [23, 224]]	0.77	0.91	0.52	0.83	[[11, 203], [32, 79]]	0.28	0.71	0.38	0.40
ml_LSVC	[[25, 53], [55, 192]]	0.78	0.78	0.55	0.78	[[10, 204], [21, 90]]	0.31	0.81	0.43	0.44
ml_SVC	[[13, 65], [29, 218]]	0.77	0.88	0.52	0.82	[[1, 213], [4, 107]]	0.33	0.96	0.48	0.50
ml_MLP	[[34, 44], [82, 165]]	0.79	0.67	0.55	0.72	[[27, 187], [24, 87]]	0.32	0.78	0.46	0.45
ml_BG	[[33, 45], [70, 177]]	0.80	0.72	0.57	0.75	[[22, 192], [21, 90]]	0.32	0.81	0.46	0.46
ml_RF	[[25, 53], [66, 181]]	0.77	0.73	0.53	0.75	[[14, 200], [17, 94]]	0.32	0.85	0.46	0.46
ml_ET	[[34, 44], [55, 192]]	0.81	0.78	0.61	0.80	[[14, 200], [20, 91]]	0.31	0.82	0.44	0.45
ml_AB	[[32, 46], [79, 168]]	0.79	0.68	0.55	0.73	[[21, 193], [21, 90]]	0.32	0.81	0.45	0.46
ml_GB	[[26, 52], [77, 170]]	0.77	0.69	0.51	0.72	[[25, 189], [23, 88]]	0.32	0.79	0.45	0.45

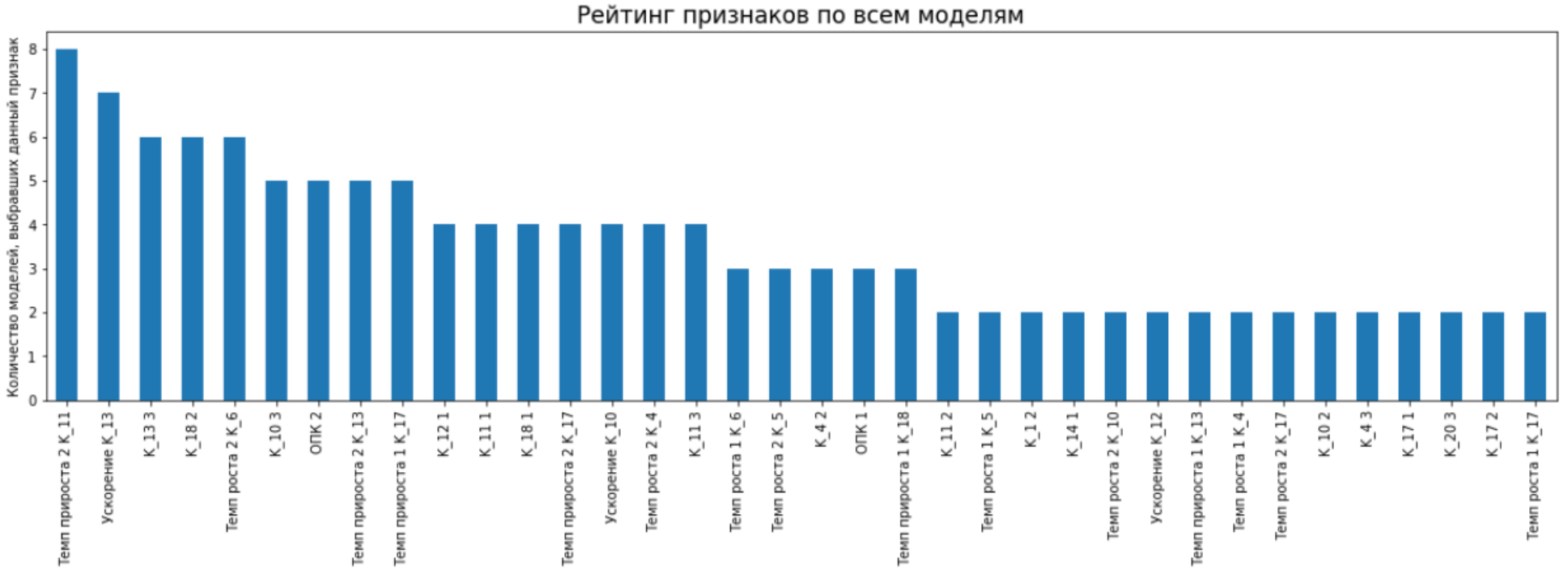
Первая модель. Она предназначается для прогнозирования факта просрочки по любым клиентам, как новым, так и с которыми давно работают.

В ней учитываются только те признаки, которые можно рассчитать, получив информацию о финансовом состоянии организации.

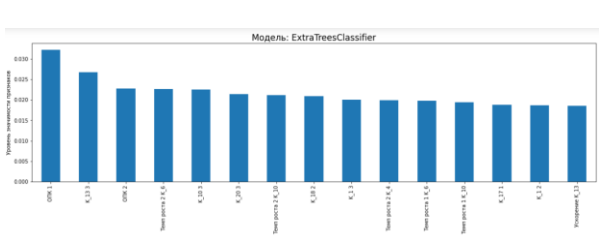
Показатели качества прогноза 2019 => 2020 гораздо выше, чем 2020 => 2021
Верхние (яркие, 4 шт) метрики это 2019 => 2020, нижние (пастельные, 4 шт) 2020 => 2021

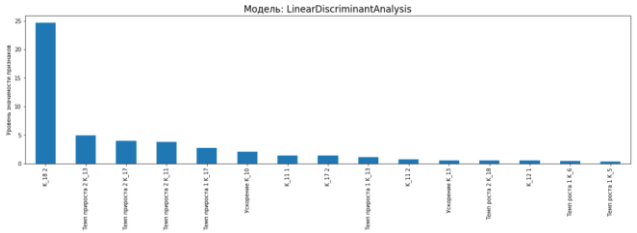


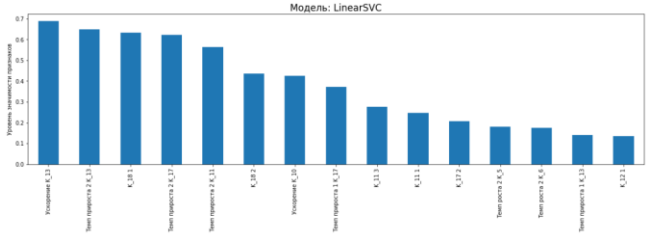
Выборка «Какие признаки модели посчитали более важными»



Так же в работе представлены рейтинги признаков по каждой модели отдельно. Отбиралось по 15 признаков у каждой модели. Предпочтения у всех разные.







Вторая модель создана с учетом дополнительных признаков характеристик задолженности за прошлые периоды. Она может использоваться для прогнозирования по клиентам, с которыми уже давно работают и имеют информацию о фактах просрочки или ее отсутствия.

В этом случае модель училась на df_2020, прогноз строился на df_2021.

Предсказания на 2021

:ances

conf_matrix	precision	recall	roc_auc	f1
[[11, 203], [17, 94]]	0.32	0.85	0.45	0.46
[[11, 203], [23, 88]]	0.30	0.79	0.42	0.44
[[22, 192], [18, 93]]	0.33	0.84	0.47	0.47
[[46, 168], [30, 81]]	0.33	0.73	0.47	0.45
[[11, 203], [32, 79]]	0.28	0.71	0.38	0.40
[[10, 204], [21, 90]]	0.31	0.81	0.43	0.44
[[1, 213], [4, 107]]	0.33	0.96	0.48	0.50
[[27, 187], [24, 87]]	0.32	0.78	0.46	0.45
[[22, 192], [21, 90]]	0.32	0.81	0.46	0.46
[[14, 200], [17, 94]]	0.32	0.85	0.46	0.46
[[14, 200], [20, 91]]	0.31	0.82	0.44	0.45
[[21, 193], [21, 90]]	0.32	0.81	0.45	0.46
[[25, 189], [23, 88]]	0.32	0.79	0.45	0.45

Прелсказания на 2021 с учетом информации о фактах просрочки с прошлых периодов

	conf_matrix	precision	recall	roc_auc	f1
ml_LR	[[22, 192], [22, 89]]	0.32	0.80	0.45	0.45
ml_LDA	[[20, 194], [23, 88]]	0.31	0.79	0.44	0.45
ml_KNN	[[23, 191], [12, 99]]	0.34	0.89	0.50	0.49
ml_CART	[[43, 171], [37, 74]]	0.30	0.67	0.43	0.42
ml_NB	[[13, 201], [32, 79]]	0.28	0.71	0.39	0.40
ml_LSVC	[[22, 192], [21, 90]]	0.32	0.81	0.46	0.46
ml_SVC	[[2, 212], [4, 107]]	0.34	0.96	0.49	0.50
ml_MLP	[[32, 182], [20, 91]]	0.33	0.82	0.48	0.47
ml_BG	[[25, 189], [21, 90]]	0.32	0.81	0.46	0.46
ml_RF	[[24, 190], [17, 94]]	0.33	0.85	0.48	0.48
ml_ET	[[22, 192], [18, 93]]	0.33	0.84	0.47	0.47
ml_AB	[[32, 182], [16, 95]]	0.34	0.86	0.50	0.49
ml_GB	[[27, 187], [18, 93]]	0.33	0.84	0.48	0.48

Точность прогнозов на 2021 все равно низкая.

Предположительные причины низкой точности прогнозов на 2021 год:

	t_2019	t_2020	t_2021
1	195	247	111
0	130	78	214

Приведено количество клиентов по годам: 1 – имеющие просрочку, 0 – не имеющие просрочки оплаты.

Модели ML, при прогнозе на 2021 год, очень много контрагентов относят к должникам, хотя они уже исправились. Но это произошло в 2021 году (в табличке выше приведены данные), а обучается и прогнозируется она на 2017-2020. Поэтому такой вариант.

В 2021 сократилось количество должников больше, чем в 2 раза. Было в 2020 - 247, а стало в 2021 - 111. 2020 год был самый такой напряженный . И в 2019 году тоже неплательщиков много по сравнению в 2021.

Это модель не в состоянии спрогнозировать.

Вариант обучить модель на df_2019, прогнозировать на df_2021 не помогло. Точность низкая.

При формировании датасетов мы сделали выборку двух крайностей (контрагентов, которые не имеют просрочки за все года и контрагентов, которые имеют просрочку ежегодно.)

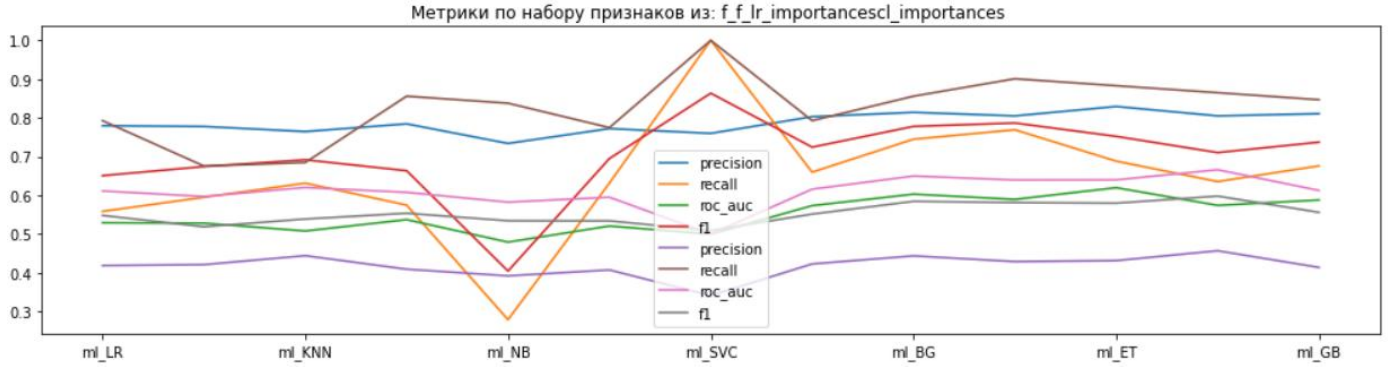
Качество метрики по прогнозам на 2021г. выросло. А вот на 2020 упало.

Т.е. получается, что в стабильной ситуации лучше работает обогащенный вариант (там, где присутствую разные категории: всегда во время платят, ежегодно имеют просрочку, и год на год не приходится.)

А вот когда нестабильная и метрики, особенно точность в 2 раза меньше, лучше сработал вариант четкого разделения. Я понимаю, что осталось всего 88 объектов и это очень мало, но с другой стороны модели лучше уловили взаимосвязи.

	Предсказания на 2020					Предсказания на 2021				
	conf_matrix	precision	recall	roc_auc	f1	conf_matrix	precision	recall	roc_auc	f1
ml_LR	[[42, 36], [107, 140]]	0.80	0.57	0.55	0.66	[[92, 122], [23, 88]]	0.42	0.79	0.61	0.55
ml_LDA	[[40, 38], [77, 170]]	0.82	0.69	0.60	0.75	[[111, 103], [36, 75]]	0.42	0.68	0.60	0.52
ml_KNN	[[34, 44], [106, 141]]	0.76	0.57	0.50	0.65	[[119, 95], [35, 76]]	0.44	0.68	0.62	0.54
ml_CART	[[46, 32], [91, 156]]	0.83	0.63	0.61	0.72	[[77, 137], [16, 95]]	0.41	0.86	0.61	0.55
ml_NB	[[64, 14], [199, 48]]	0.77	0.19	0.51	0.31	[[70, 144], [18, 93]]	0.39	0.84	0.58	0.53
ml_LSVC	[[41, 37], [97, 150]]	0.80	0.61	0.57	0.69	[[89, 125], [25, 86]]	0.41	0.77	0.60	0.53
ml_SVC	[[17, 61], [32, 215]]	0.78	0.87	0.54	0.82	[[0, 214], [0, 111]]	0.34	1.00	0.50	0.51
ml_MLP	[[41, 37], [131, 116]]	0.76	0.47	0.50	0.58	[[94, 120], [23, 88]]	0.42	0.79	0.62	0.55
ml_BG	[[34, 44], [84, 163]]	0.79	0.66	0.55	0.72	[[95, 119], [16, 95]]	0.44	0.86	0.65	0.58
ml_RF	[[38, 40], [77, 170]]	0.81	0.69	0.59	0.74	[[81, 133], [11, 100]]	0.43	0.90	0.64	0.58
ml_ET	[[35, 43], [78, 169]]	0.80	0.68	0.57	0.74	[[85, 129], [13, 98]]	0.43	0.88	0.64	0.58
ml_AB	[[41, 37], [95, 152]]	0.80	0.62	0.57	0.70	[[100, 114], [15, 96]]	0.46	0.86	0.67	0.60
ml_GB	[[37, 41], [74, 173]]	0.81	0.70	0.59	0.75	[[81, 133], [17, 94]]	0.41	0.85	0.61	0.56

Графики метрик поперемешались. Одни 2021 подросли, а 2020 просели.



Ранжирование признаков немного изменилось (признаки сместились, поменялись местами).

Модели уловили другие закономерности.

Заключение:

Достаточно высокой точности добиться не удалось. В основном проведена исследовательская работа.

Дальнейшее улучшение моделей ML не проводилось в силу того, что время ограничено, необходимы дополнительные данные.

Подбор гиперпараметров ничего существенного не изменил.

Скорее всего, необходимо:

- информация о наличие судебных процессов у этого контрагента по вопросам взыскания долгов.
- вид экономической деятельности
- является ли Северсталь основным поставщиком этого контрагента
- суммы контрактов, регулярность поставок
- возможно регион осуществления деятельности.

Работа выполнена в нескольких ноутбуках (каждый блок в отдельном). Обмен между ними производится путем выгрузки – загрузки файлов.

Размещена на Гитхабе: https://github.com/NataliaKolesnik/Hackathon_Severstal