**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления КАФЕДРА Системы обработки информации и управления

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

***К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:***

***Решение задачи машинного обучения***

Студент ИУ5-62Б **Р.И. Ким**

(Группа) (Подпись, дата) (И.О.Фамилия)

#### Руководитель курсового проекта

**Ю.Е. Гапанюк**

(Подпись, дата) (И.О.Фамилия)

#### Консультант

**Ю.Е. Гапанюк**

(Подпись, дата) (И.О.Фамилия)

*2020 г.*

**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**З А Д А Н И Е**

#### УТВЕРЖДАЮ

Заведующий кафедрой ИУ5

(Индекс)

#### В.М. Черненький

(И.О.Фамилия)

#### « » 20 г.

**на выполнение курсового проекта**

#### по дисциплине Технологии машинного обучения Студент группы ИУ5-62Б

Ким Радмир Ильшадович

(Фамилия, имя, отчество)

#### Тема курсового проекта Решение задачи машинного обучения

Направленность КП (учебный, исследовательский, практический, производственный, др.) учебный

Источник тематики (кафедра, предприятие, НИР)

График выполнения проекта: 25% к 5 нед., 50% к 9 нед., 75% к 13 нед., 100% к 16 нед.

***Задание\_*** Решение задачи машинного обучения на основе материалов дисциплины.

\_ Результатом курсового проекта является отчет, содержащий описания моделей,

\_ тексты программ и результаты экспериментов.

***Оформление курсового проекта:***

#### Расчетно-пояснительная записка на \_35 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « \_7\_ » фев. 2020 г.

**Руководитель курсового проекта** Ю.Е. Гапанюк

(Подпись, дата) (И.О.Фамилия)

**Студент**  Р.И. Ким

(Подпись, дата) (И.О.Фамилия)

#### Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Содержание

[ВВЕДЕНИЕ 4](#_TOC_250010)

[Задание 5](#_TOC_250009)

1. Поиск и выбор набора данных для построения моделей машинного обучения 5
2. [Проведение разведочного анализа данных. Построение графиков, необходимых для понимания структуры данных. Анализ и заполнение пропусков в данных 6](#_TOC_250008)
3. [Выбор признаков, подходящих для построения моделей. Кодирование категориальных признаков. Масштабирование данных. Формирование вспомогательных признаков, улучшающих качество моделей 12](#_TOC_250007)
4. [Проведение корреляционного анализа данных. Формирование промежуточных выводов о возможности построения моделей машинного обучения 14](#_TOC_250006)
5. [Выбор метрик для последующей оценки качества моделей 16](#_TOC_250005)
6. [Выбор наиболее подходящих моделей для решения задачи классификации или регрессии](#_TOC_250004)

......................................................................................................................................................17

1. [Формирование обучающей и тестовой выборок на основе исходного набора данных 17](#_TOC_250003)
2. [Построение базового решения (baseline) для выбранных моделей без подбора гиперпараметров. Производится обучение моделей на основе обучающей выборки и оценка качества моделей на основе тестовой выборки 18](#_TOC_250002)
3. Подбор гиперпараметров для выбранных моделей. Рекомендуется использовать методы кросс-валидации 22
4. [Повторение пункта 8 для найденных оптимальных значений гиперпараметров. Сравнение качества полученных моделей с качеством baseline-моделей 26](#_TOC_250001)
5. Формирование выводов о качестве построенных моделей на основе выбранных метрик.30 ЗАКЛЮЧЕНИЕ 32

[ЛИТЕРАТУРА 33](#_TOC_250000)

# ВВЕДЕНИЕ

### Курсовой проект – самостоятельная часть учебной дисциплины «Технологии машинного обучения» – учебная и практическая исследовательская студенческая работа, направленная на решение комплексной задачи машинного обучения. Результатом курсового проекта является отчет, содержащий описания моделей, тексты программ и результаты экспериментов.

Курсовой проект опирается на знания, умения и владения, полученные студентом в рамках лекций и лабораторных работ по дисциплине.

В рамках курсового проекта возможно проведение типового или нетипового исследования.

* + Типовое исследование - решение задачи машинного обучения на основе материалов дисциплины. Выполняется студентом единолично.
  + Нетиповое исследование - решение нестандартной задачи. Тема должна быть согласована с преподавателем. Как правило, такая работа выполняется группой студентов.

**Курсовой проект по ТМО**

**Задание**

Схема типового исследования, проводимого студентом в рамках курсовой работы, содержит выполнение следующих шагов:

1. Поиск и выбор набора данных для построения моделей машинного обучения. На основе выбранного набора данных студент должен построить модели машинного обучения для решения или задачи классификации, или задачи регрессии.
2. Проведение разведочного анализа данных. Построение графиков, необходимых для понимания структуры данных. Анализ и заполнение пропусков в данных.
3. Выбор признаков, подходящих для построения моделей. Кодирование категориальных признаков. Масштабирование данных. Формирование вспомогательных признаков, улучшающих качество моделей.
4. Проведение корреляционного анализа данных. Формирование промежуточных выводов о возможности построения моделей машинного обучения. В зависимости от набора данных, порядок выполнения пунктов 2, 3, 4 может быть изменен.
5. Выбор метрик для последующей оценки качества моделей. Необходимо выбрать не менее трех метрик и обосновать выбор.
6. Выбор наиболее подходящих моделей для решения задачи классификации или регрессии. Необходимо использовать не менее пяти моделей, две из которых должны быть ансамблевыми.
7. Формирование обучающей и тестовой выборок на основе исходного набора данных.
8. Построение базового решения (baseline) для выбранных моделей без подбора гиперпараметров. Производится обучение моделей на основе обучающей выборки и оценка качества моделей на основе тестовой выборки.
9. Подбор гиперпараметров для выбранных моделей. Рекомендуется использовать методы кросс-валидации. В зависимости от используемой библиотеки можно применять функцию GridSearchCV, использовать перебор параметров в цикле, или использовать другие методы.
10. Повторение пункта 8 для найденных оптимальных значений гиперпараметров. Сравнение качества полученных моделей с качеством baseline-моделей.
11. Формирование выводов о качестве построенных моделей на основе выбранных метрик. Результаты сравнения качества рекомендуется отобразить в виде графиков и сделать выводы в форме текстового описания. Рекомендуется построение графиков обучения и валидации, влияния значений гиперпарметров на качество моделей и т.д.

Приведенная схема исследования является рекомендуемой. В зависимости от решаемой задачи возможны модификации.

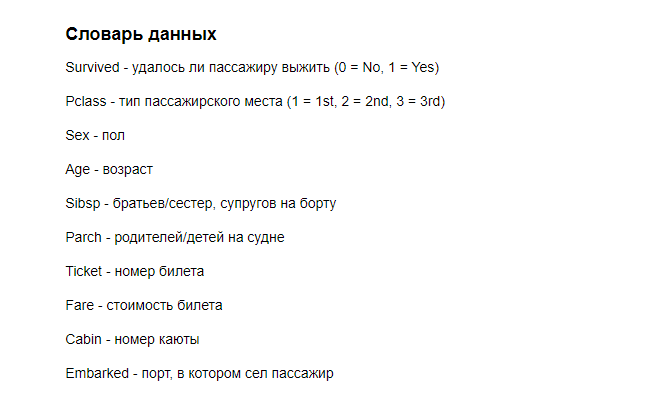
## Ход выполнения курсового проекта

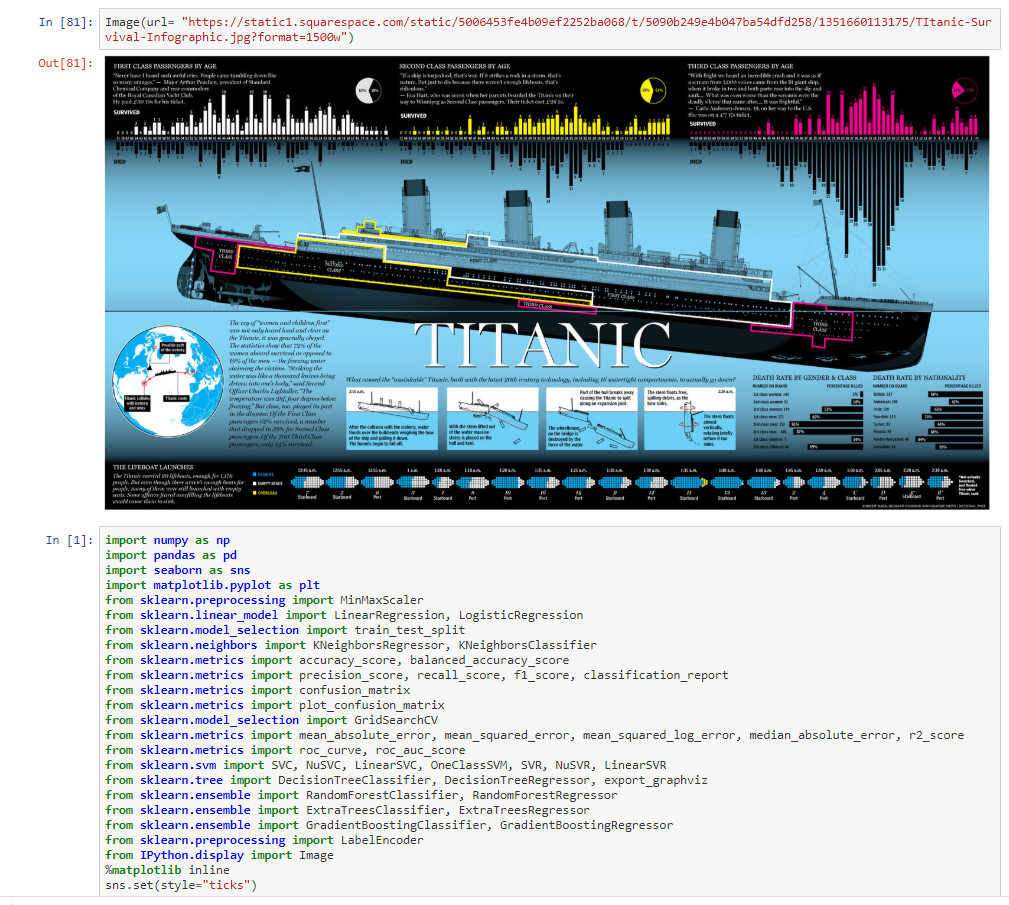
##### Поиск и выбор набора данных для построения моделей машинного обучения. На основе выбранного набора данных студент должен построить модели машинного обучения для решения или задачи классификации, или задачи регрессии.

Будем использовать набор данных, который описывает имеющуюся информацию о пассажирах “Титаника”. “Титаник” — британский трансатлантический пароход, второй лайнер класса «Олимпик». Строился в Белфасте на верфи «Harland & Wolff» с 1909 по 1912 год по заказу судоходной компании «White Star Line». На момент ввода в эксплуатацию являлся самым большим судном в мире. В ночь с 14 на 15 апреля 1912 года во время первого рейса потерпел крушение в северной Атлантике, столкнувшись с айсбергом. Целью работы будет задача классификации, модель должна будет предсказать выживет пассажир или нет.

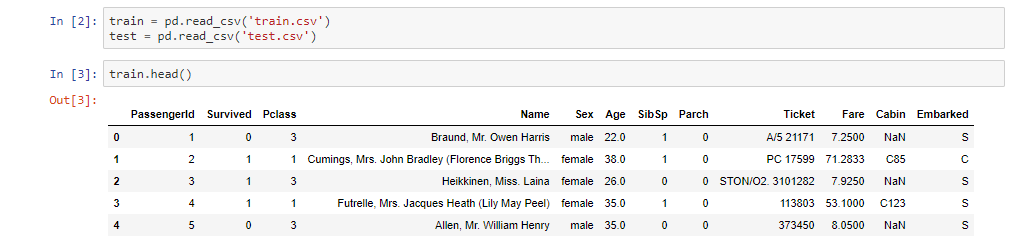
Каждый пассажир описывается 5 непрерывными переменными и 5 переменными класса.

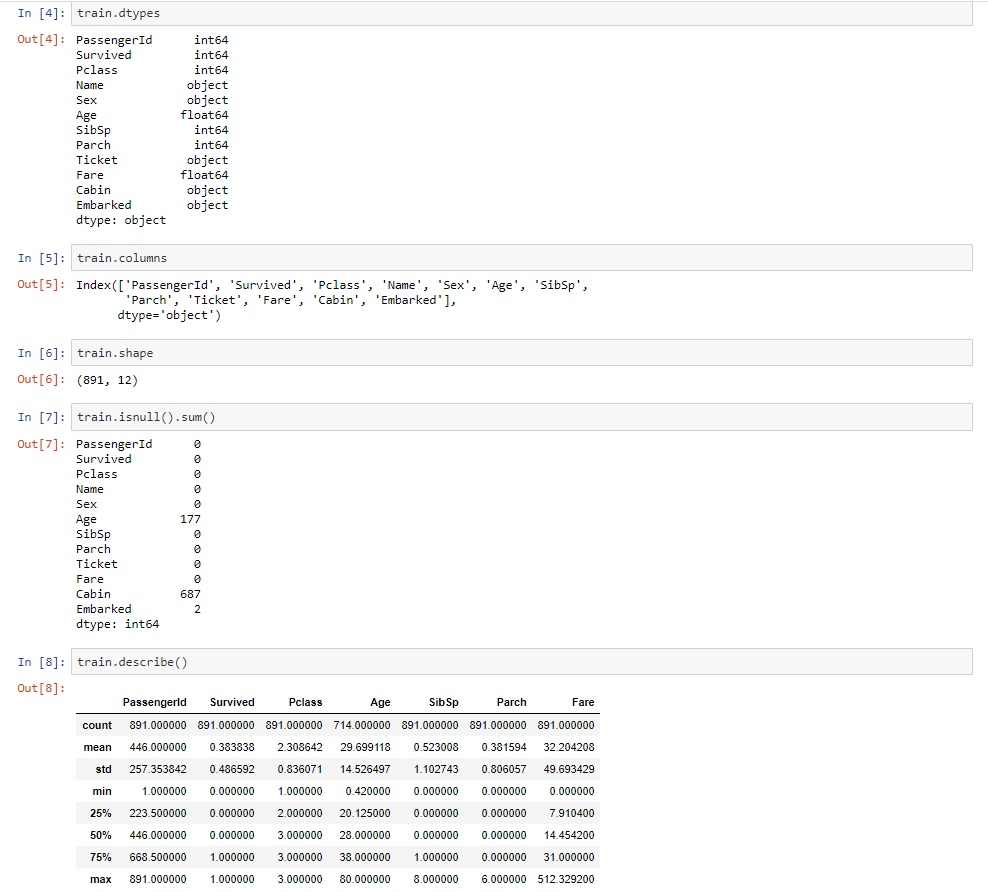
Таким образом, набор данных содержит следующие колонки:



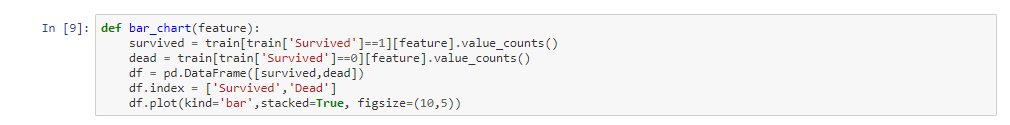


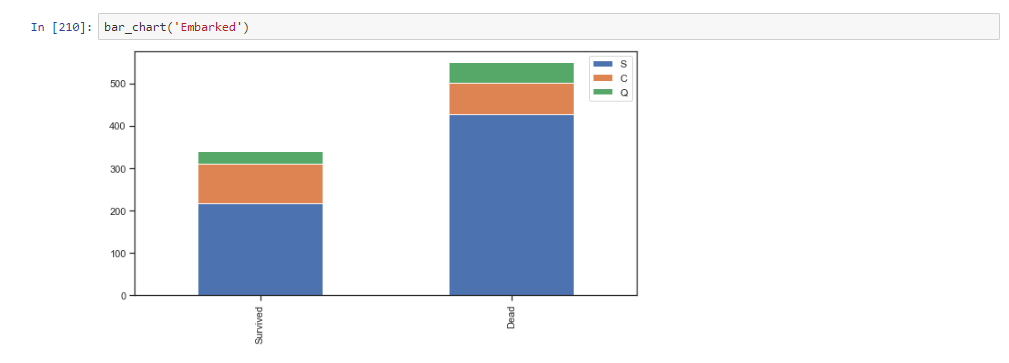
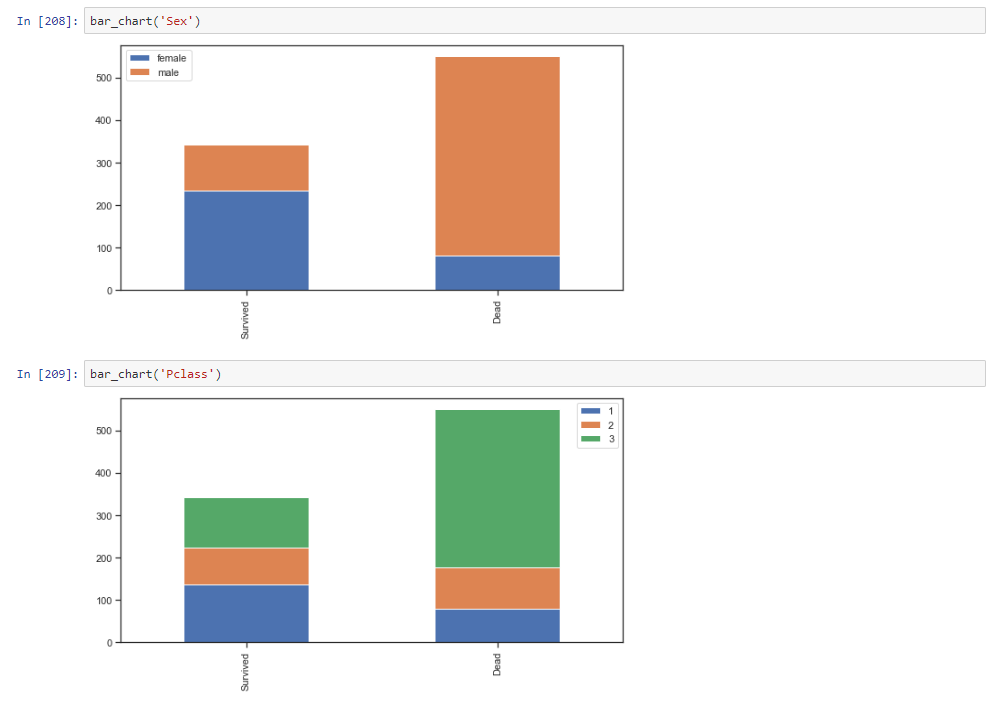
##### Проведение разведочного анализа данных. Построение графиков, необходимых для понимания структуры данных. Анализ и заполнение пропусков в данных.

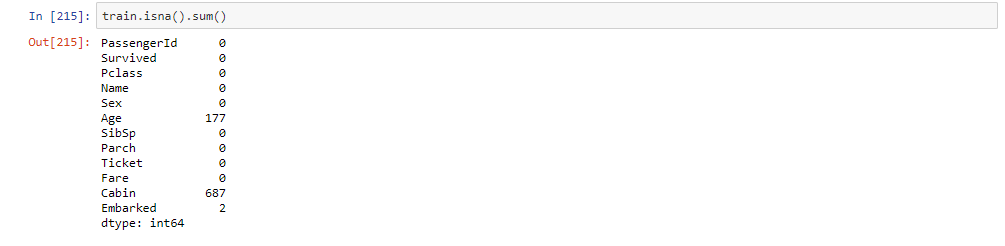
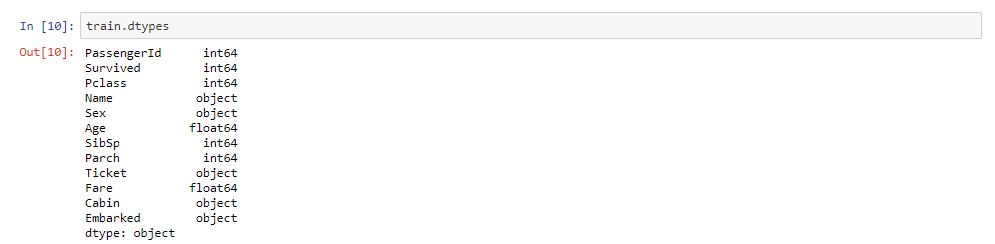


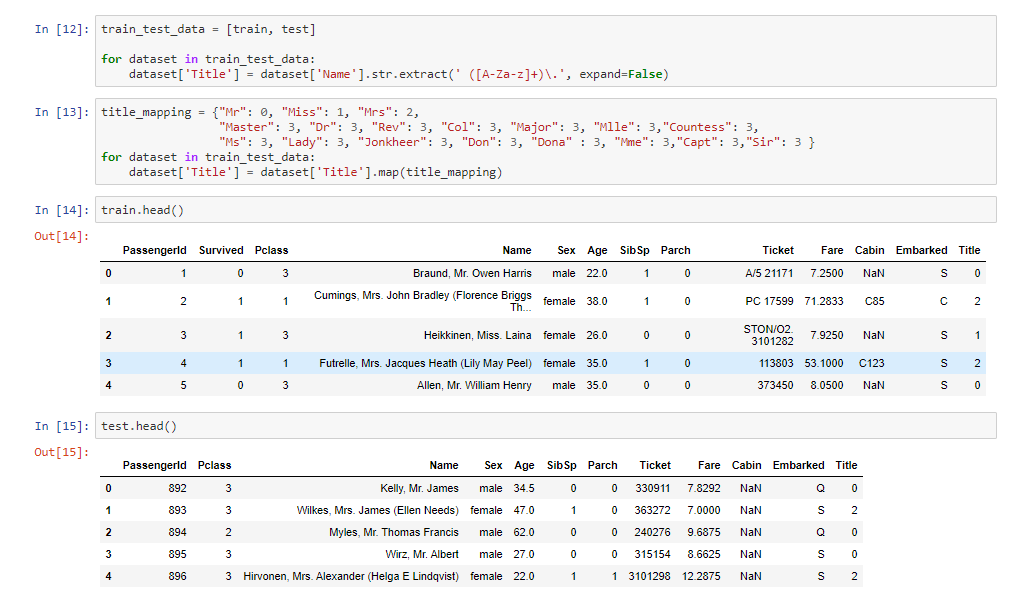


Создадим функцию для отрисовки гистограмм для первичного анализа



Видим, что в данных присутствует много категориальных признаков и пропусков, необходимо их обработать

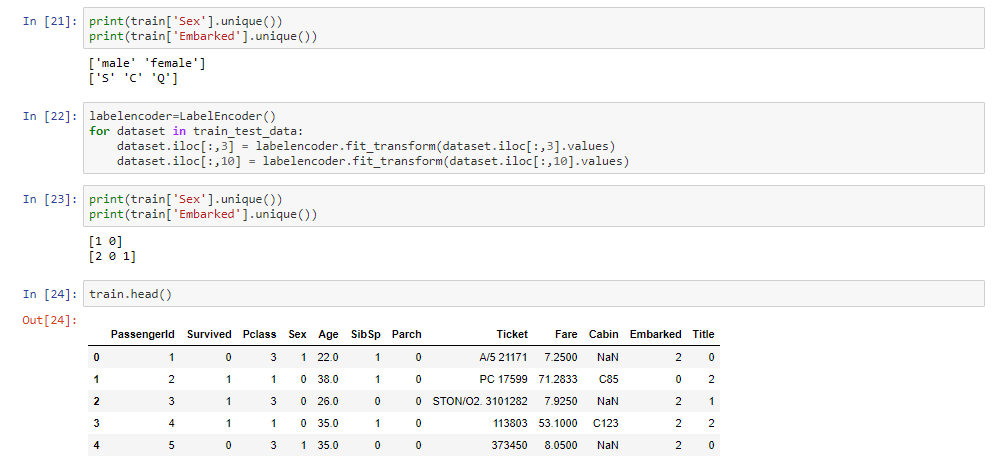




Из анализа ниже видим, что пропуски в столбце Embarked мы можем заменить модой, пропуски возраста заменим медианой



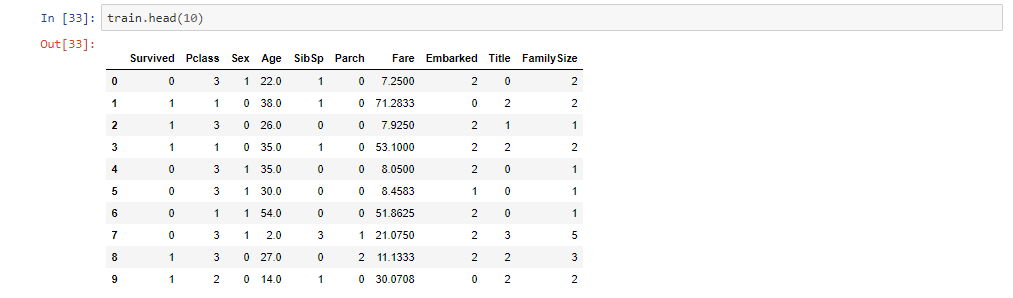
Закодируем категориальные признаки:

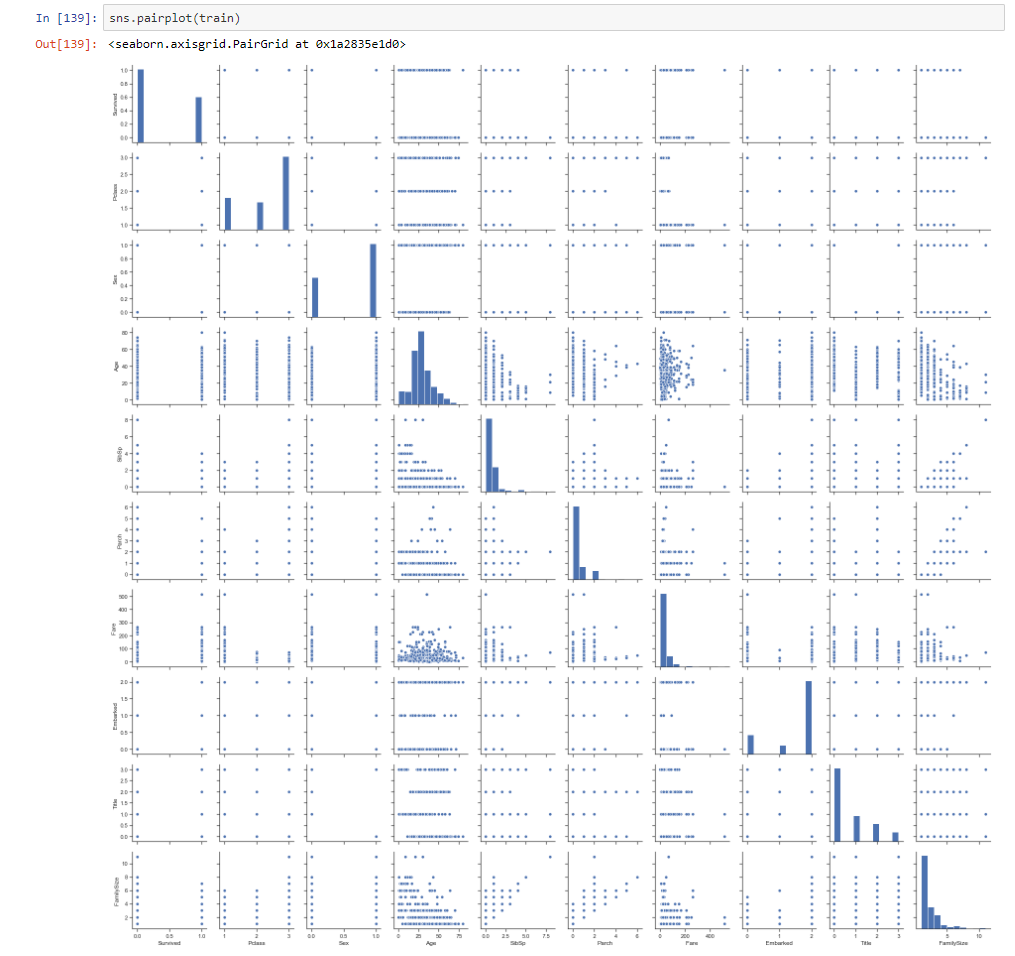


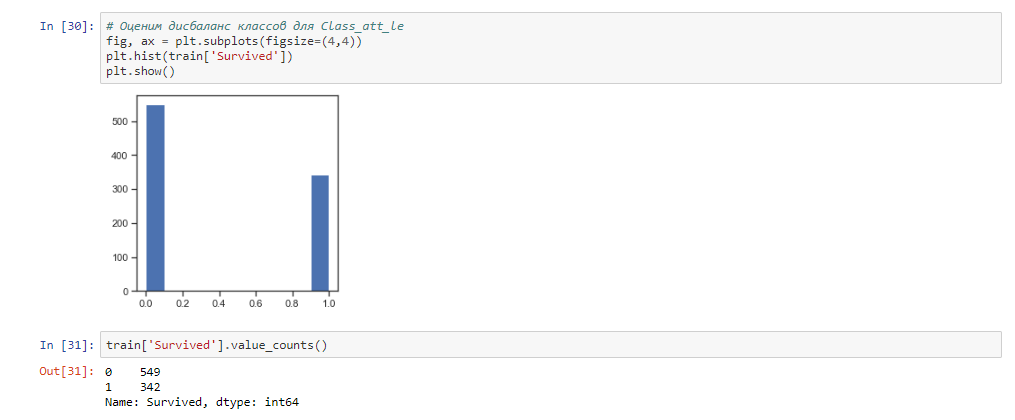
Столбцы с большим количеством пропусков удалям из нашей выборки:



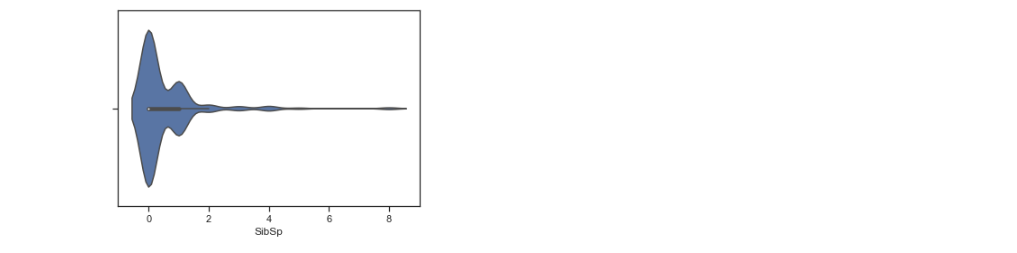
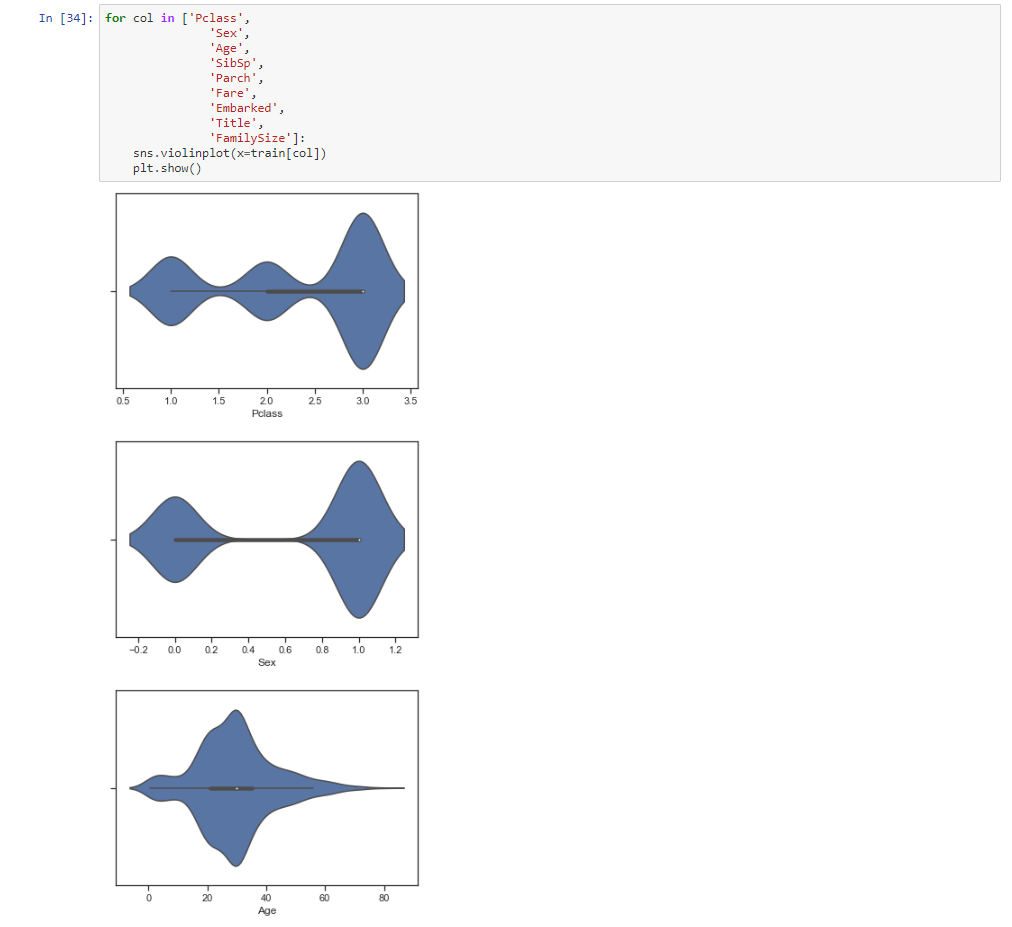
Получаем датасет полностью готовый к анализу

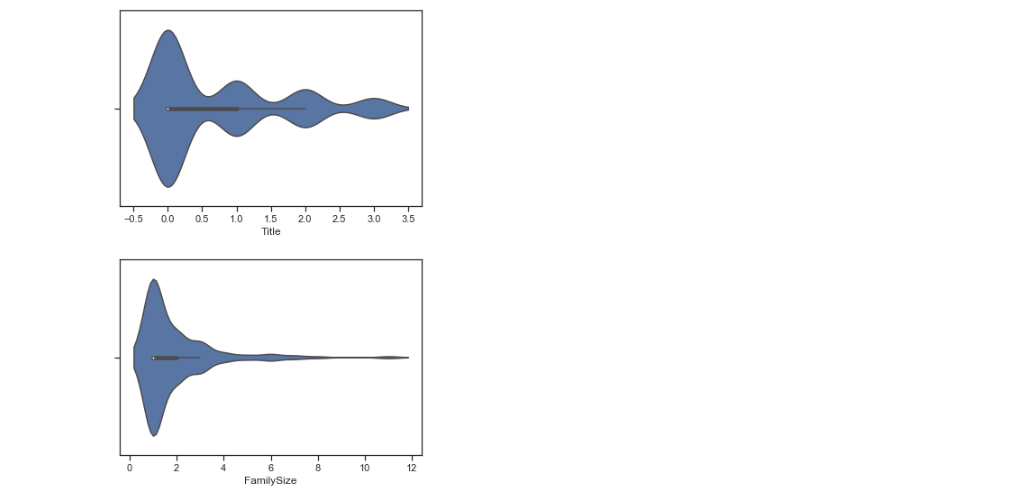
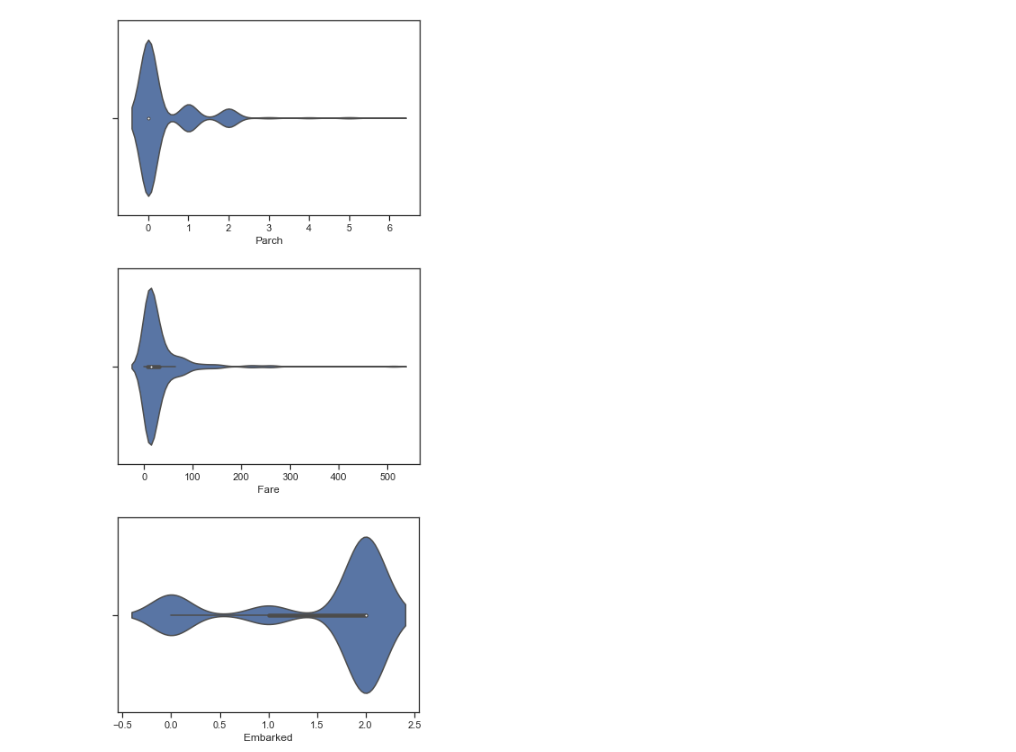






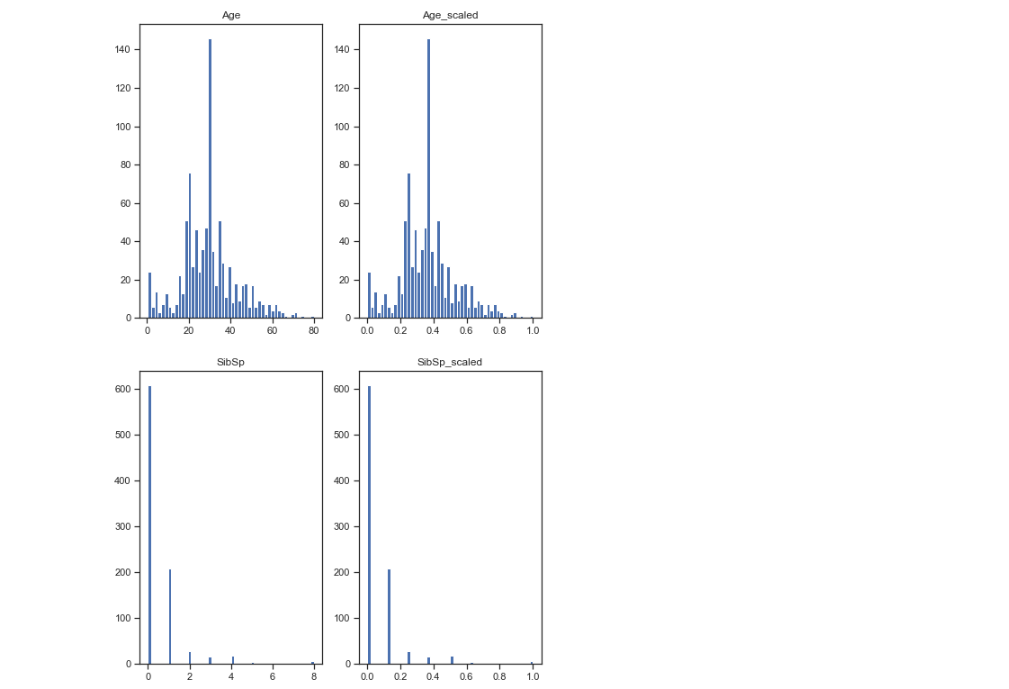
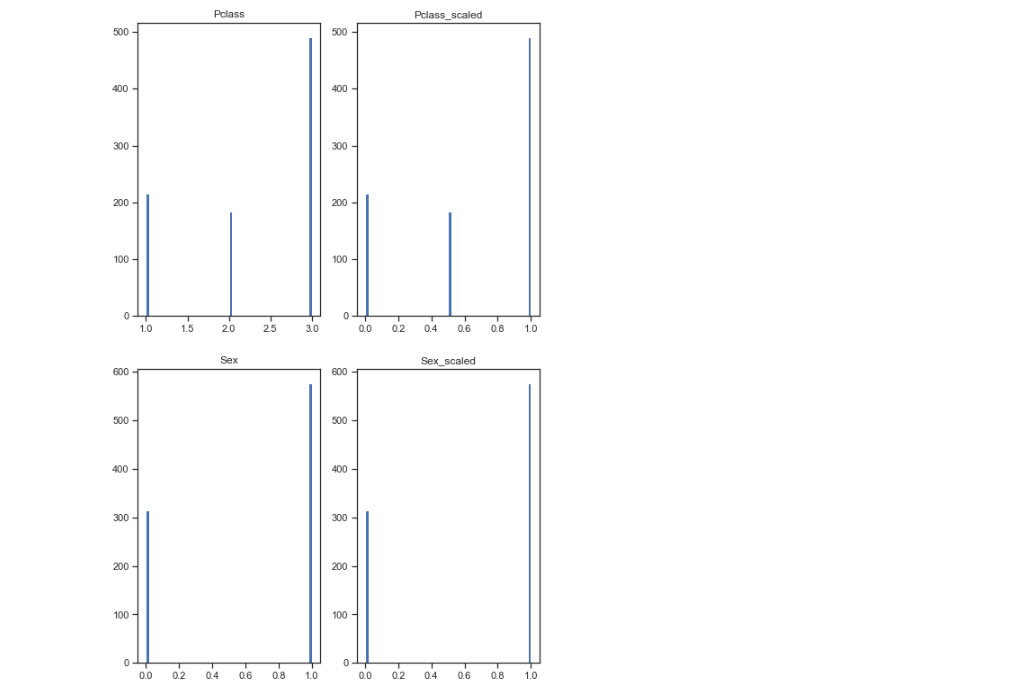
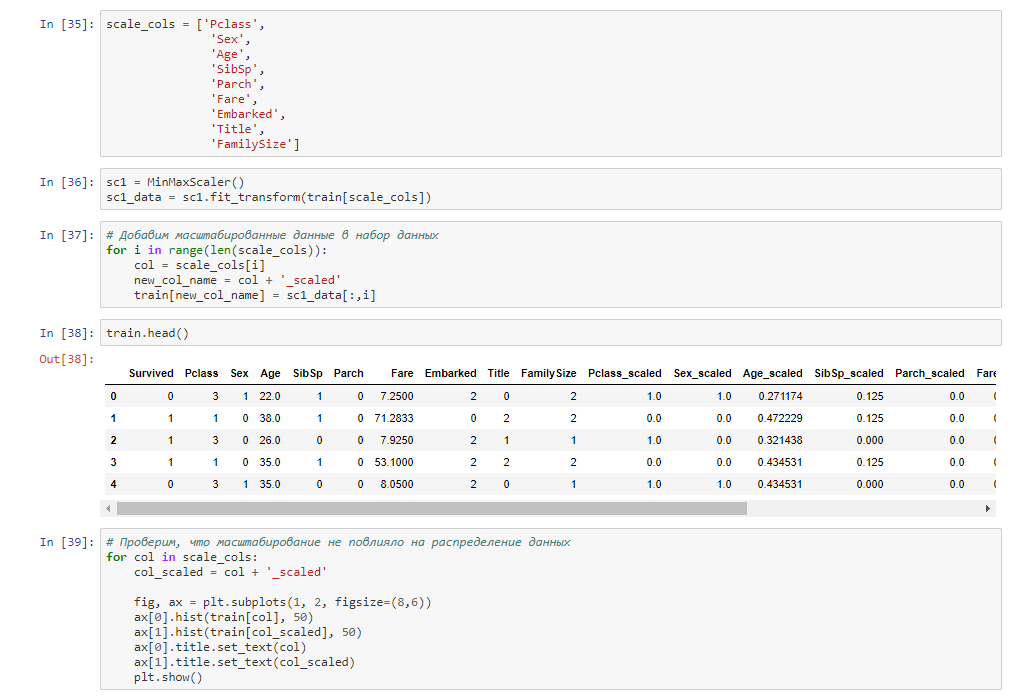
Построим скрипичные диаграммы:

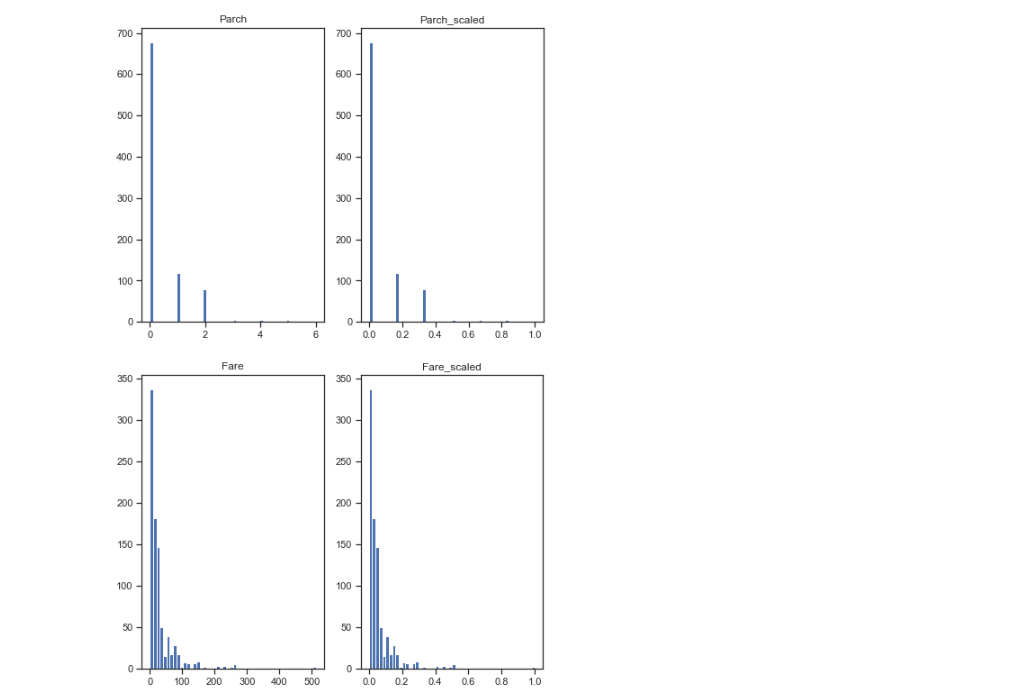


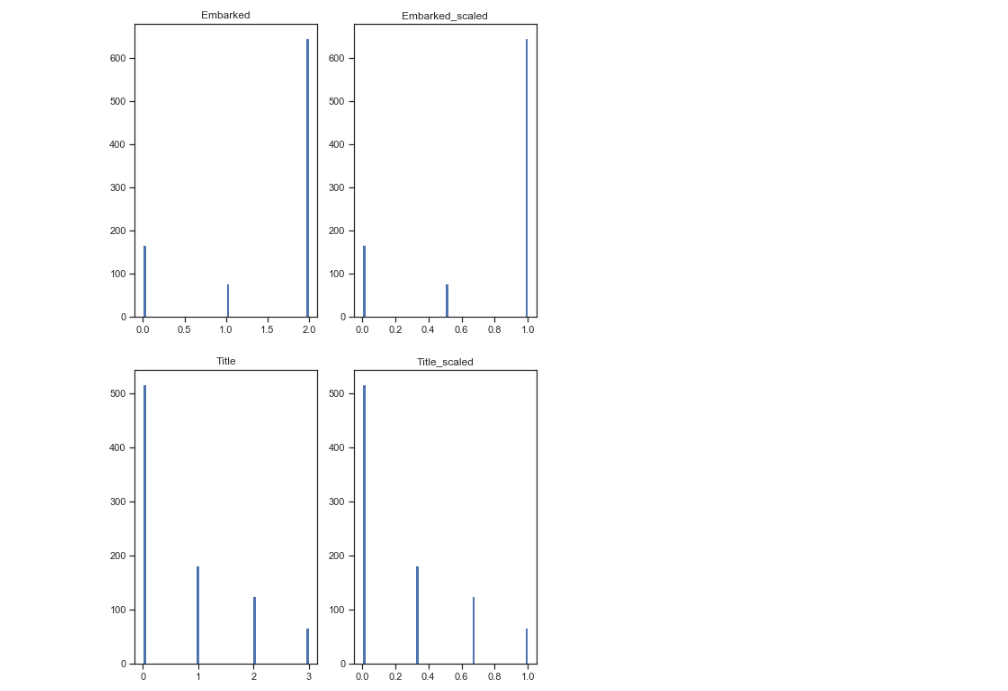


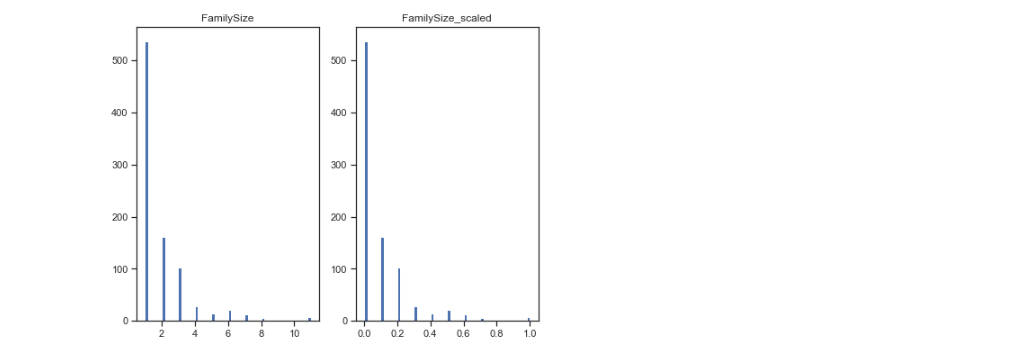
##### Выбор признаков, подходящих для построения моделей. Кодирование категориальных признаков. Масштабирование данных. Формирование вспомогательных признаков, улучшающих качество моделей

Для построения моделей будем использовать все признаки. Категориальные признаки отсутствуют, их кодирование не требуется. Выполним масштабирование данных.



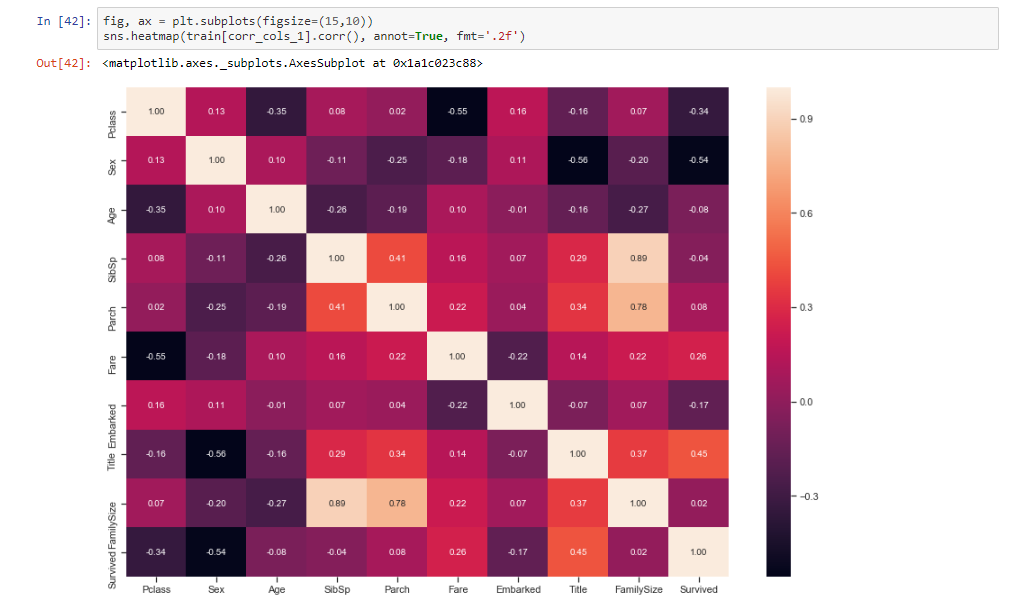


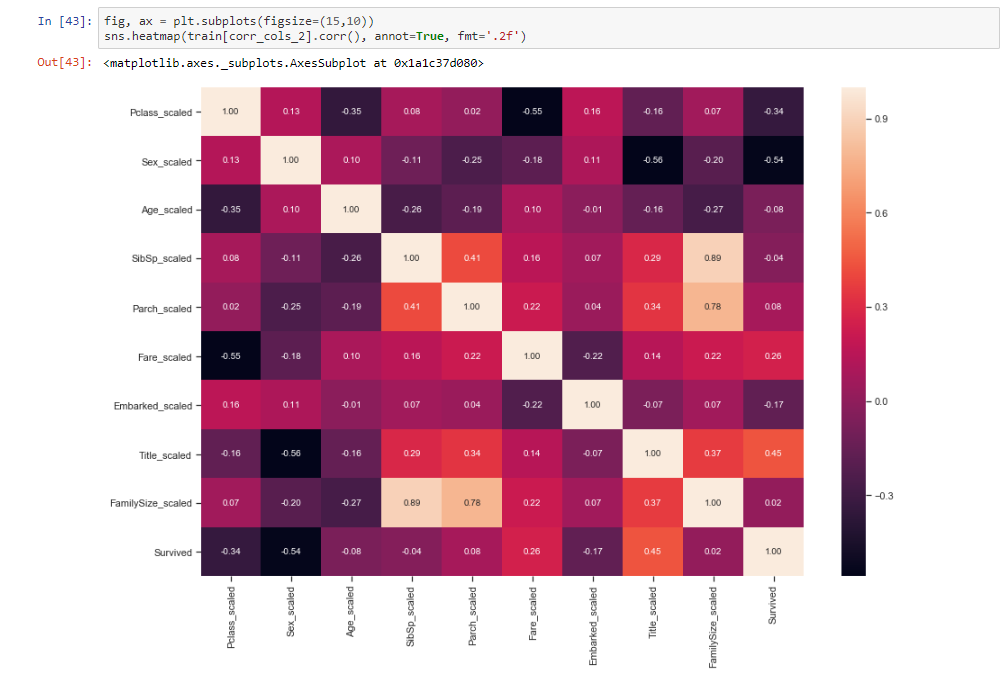




##### Проведение корреляционного анализа данных. Формирование промежуточных выводов о возможности построения моделей машинного обучения







На основе корреляционной матрицы можно сделать следующие выводы:

Корреляционные матрицы для исходных и масштабированных данных совпадают.

Целевой признак классификации "Survived" наиболее сильно коррелирует с "Title" (0.450) , "Fare" (0.260) и “Prach”. Эти признаки обязательно следует оставить в модели классификации.

Большие по модулю значения коэффициентов корреляции свидетельствуют о значимой корреляции между исходными признаками и целевым признаком. На основании корреляционной матрицы можно сделать вывод о том, что данные позволяют построить модель машинного обучения.

##### Выбор метрик для последующей оценки качества моделей

**В качестве метрик для решения задачи классификации будем использовать:**

Precision - доля верно предсказанных классификатором положительных объектов, из всех объектов, которые классификатор верно или неверно определил как положительные.

Recall - доля верно предсказанных классификатором положительных объектов, из всех действительно положительных объектов. ROC AUC. Основана на вычислении следующих характеристик:

True Positive Rate, откладывается по оси ординат. Совпадает с recall.

False Positive Rate, откладывается по оси абсцисс. Показывает какую долю из объектов отрицательного класса алгоритм предсказал неверно.

F1-мера - вычисляется как среднее гармоническое от precision и recall.

Accuracy - вычисляет процент (долю в диапазоне от 0 до 1) правильно определенных классов. Метрика "Accuracy" показывает точность по всем классам, но точность может быть различной для различных классов. Так как в данном наборе данных присутствует дисбаланс классов, будем использовать balanced\_accuracy\_score.

In [24]:

*# Отрисовка ROC-кривой*

**def** draw\_roc\_curve(y\_true, y\_score, pos\_label=1, average='micro'): fpr, tpr, thresholds = roc\_curve(y\_true, y\_score,

pos\_label=pos\_label) roc\_auc\_value = roc\_auc\_score(y\_true, y\_score, average=average) plt.figure()

lw = 2

plt.plot(fpr, tpr, color='darkorange',

lw=lw, label='ROC curve (area = **%0.2f**)' % roc\_auc\_value) plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05]) plt.xlabel('False Positive Rate') plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic') plt.legend(loc="lower right")

plt.show()

Разработаем класс, который позволит сохранять метрики качества построенных моделей и реализует визуализацию метрик качества.

In [25]:

**class MetricLogger**:

**def** init (self): self.df = pd.DataFrame(

{'metric': pd.Series([], dtype='str'),

'alg': pd.Series([], dtype='str'),

'value': pd.Series([], dtype='float')})

**def** add(self, metric, alg, value):

*"""*

*Добавление значения*

*"""*

*# Удаление значения если оно уже было ранее добавлено* self.df.drop(self.df[(self.df['metric']==metric)&(self.df['alg']==alg)].index, inplace = **True**) *# Добавление нового значения*

temp = [{'metric':metric, 'alg':alg, 'value':value}] self.df = self.df.append(temp, ignore\_index=**True**)

**def** get\_data\_for\_metric(self, metric, ascending=**True**):

*"""*

*Формирование данных с фильтром по метрике*

*"""*

temp\_data = self.df[self.df['metric']==metric]

temp\_data\_2 = temp\_data.sort\_values(by='value', ascending=ascending)

**return** temp\_data\_2['alg'].values, temp\_data\_2['value'].values

**def** plot(self, str\_header, metric, ascending=**True**, figsize=(5, 5)):

*"""*

*Вывод графика*

*"""*

array\_labels, array\_metric = self.get\_data\_for\_metric(metric, ascending) fig, ax1 = plt.subplots(figsize=figsize)

pos = np.arange(len(array\_metric)) rects = ax1.barh(pos, array\_metric,

align='center', height=0.5, tick\_label=array\_labels)

ax1.set\_title(str\_header)

**for** a,b **in** zip(pos, array\_metric):

plt.text(0.5, a-0.05, str(round(b,3)), color='white') plt.show()

##### Выбор наиболее подходящих моделей для решения задачи классификации или регрессии

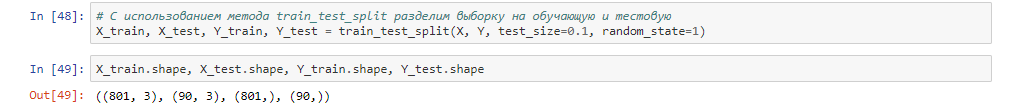
Для задачи классификации будем использовать следующие модели:

Логистическая регрессия Метод ближайших соседей Машина опорных векторов Решающее дерево Случайный лес Градиентный бустинг

##### Формирование обучающей и тестовой выборок на основе исходного набора данных



17



##### Построение базового решения (baseline) для выбранных моделей без подбора гиперпараметров. Производится обучение моделей на основе обучающей выборки и оценка качества моделей на основе тестовой выборки

In [30]:

*# Модели*

clas\_models = {'LogR': LogisticRegression(),

'KNN\_4':KNeighborsClassifier(n\_neighbors=4), 'SVC':SVC(),

'Tree':DecisionTreeClassifier(), 'RF':RandomForestClassifier(), 'GB':GradientBoostingClassifier()}

In [31]:

*# Сохранение метрик*

clasMetricLogger = MetricLogger()

In [32]:

**def** train\_model(model\_name, model, MetricLogger): model.fit(X\_train, Y\_train)

Y\_pred = model.predict(X\_test)

precision = precision\_score(Y\_test.values, Y\_pred) recall = recall\_score(Y\_test.values, Y\_pred) roc\_auc = roc\_auc\_score(Y\_test.values, Y\_pred)

f1 = f1\_score(Y\_test.values, Y\_pred)

bal\_accuracy = balanced\_accuracy\_score(Y\_test.values, Y\_pred)

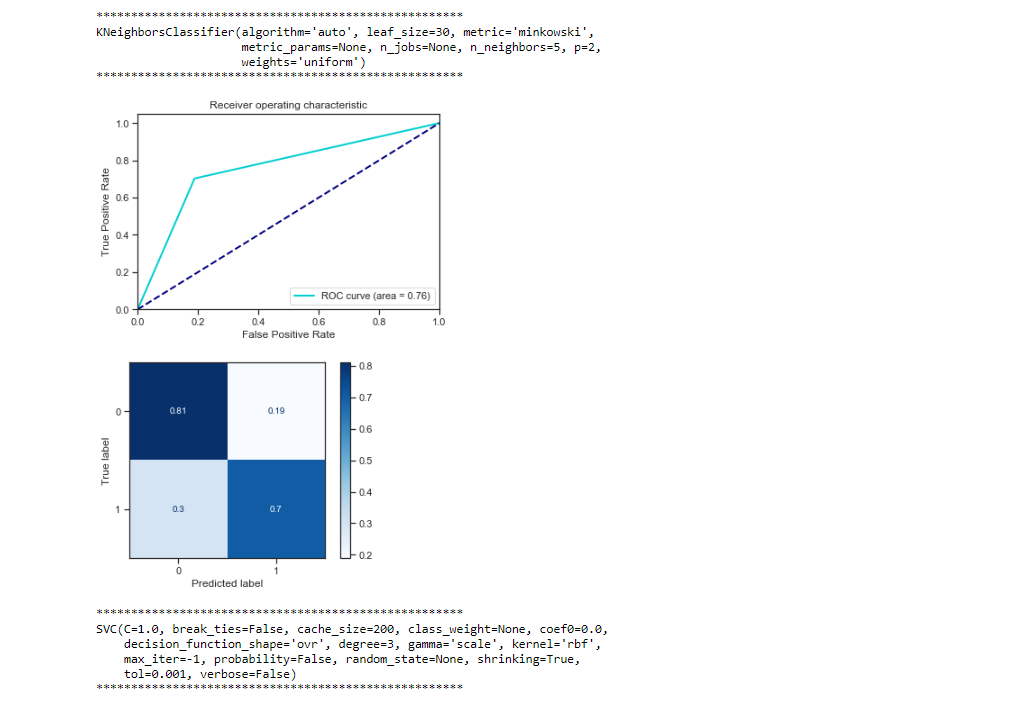
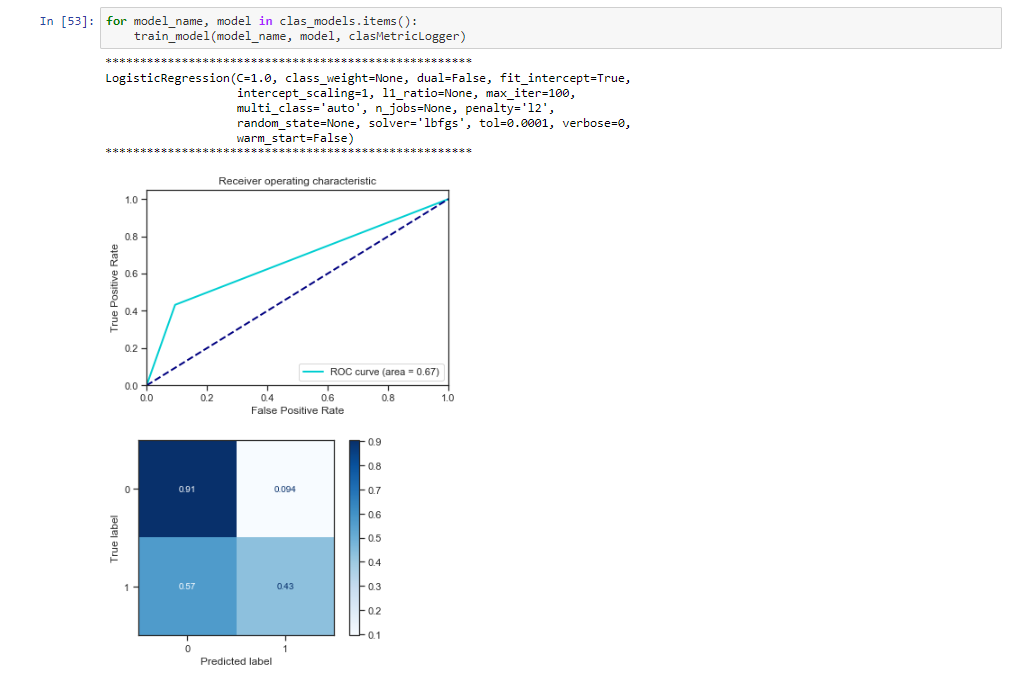
MetricLogger.add('precision', model\_name, precision) MetricLogger.add('recall', model\_name, recall) MetricLogger.add('roc\_auc', model\_name, roc\_auc) MetricLogger.add('f1', model\_name, f1) MetricLogger.add('bal\_accuracy', model\_name, bal\_accuracy)

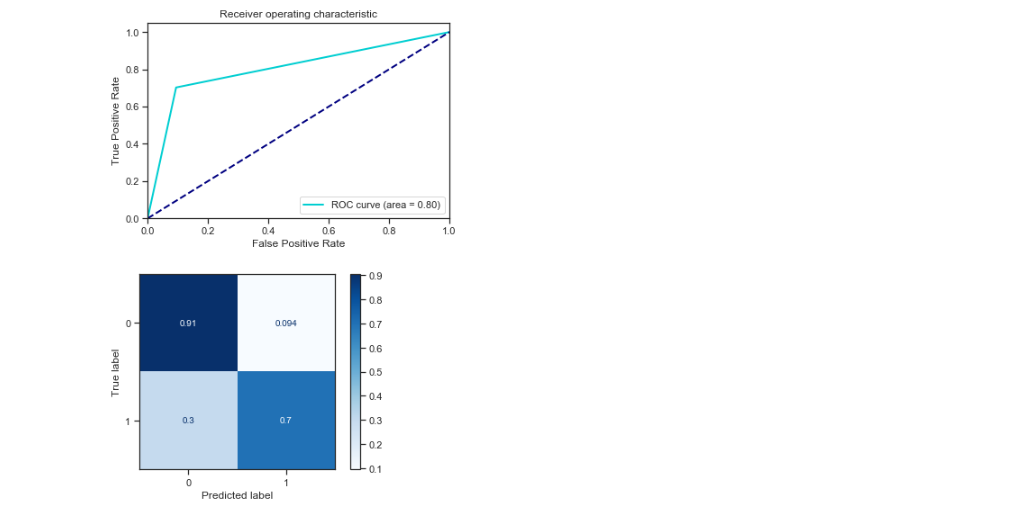
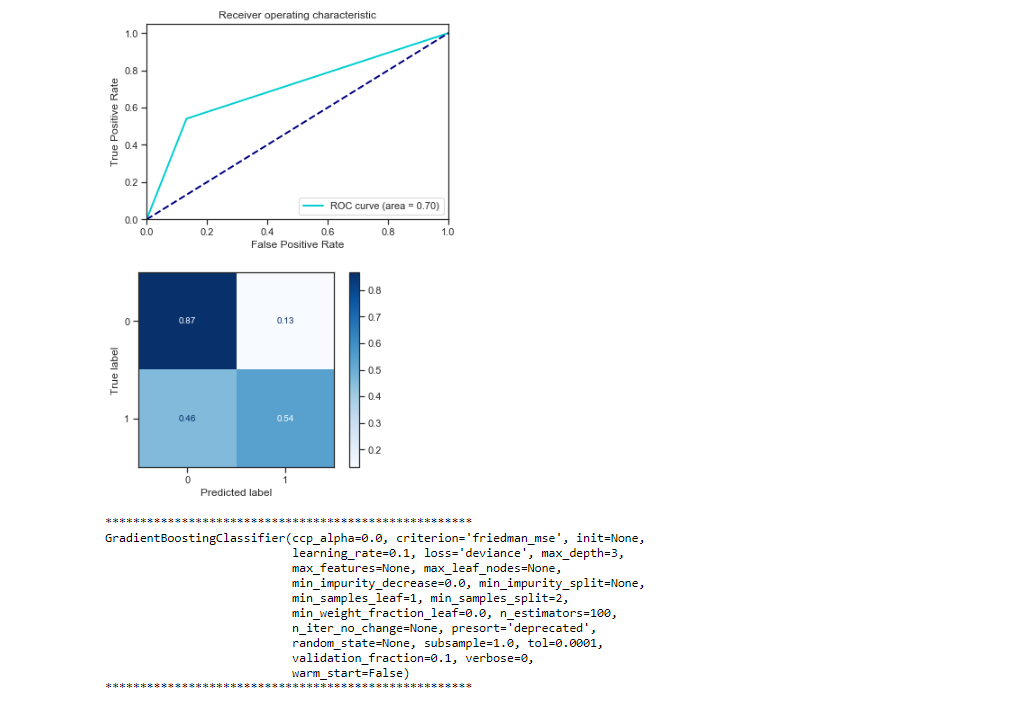
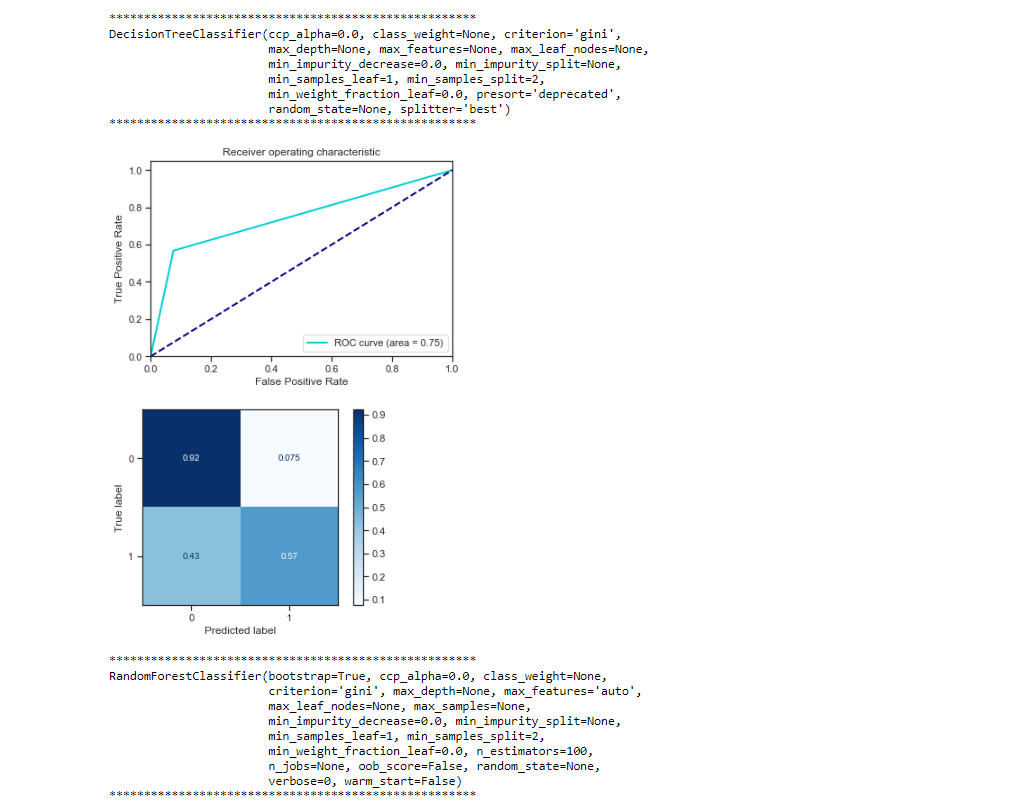
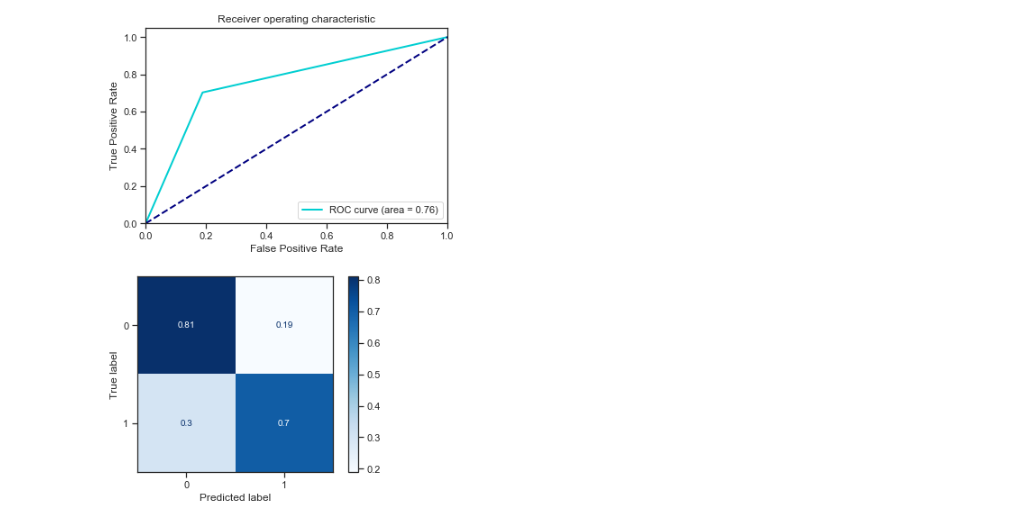
print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') print(model) print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') draw\_roc\_curve(Y\_test.values, Y\_pred)

plot\_confusion\_matrix(model, X\_test, Y\_test.values,

display\_labels=['0','1'], cmap=plt.cm.Blues, normalize='true')

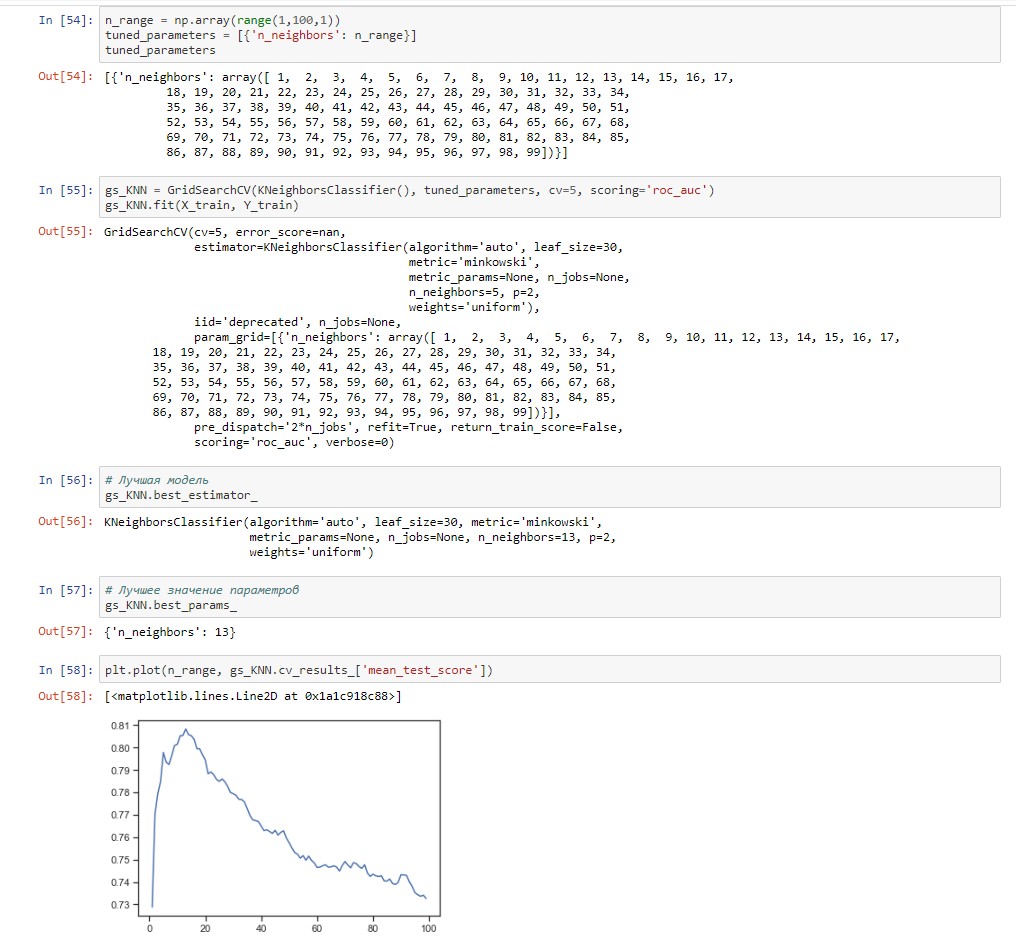
plt.show()



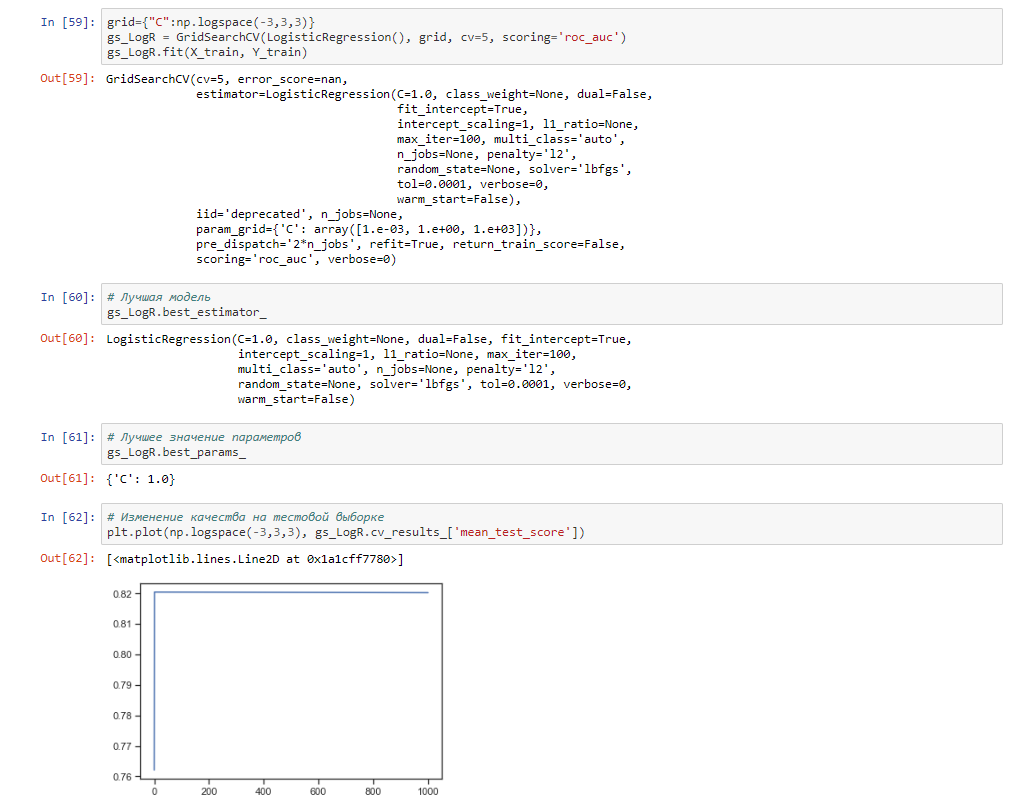


##### Подбор гиперпараметров для выбранных моделей. Рекомендуется использовать методы кросс- валидации. В зависимости от используемой библиотеки можно применять функцию GridSearchCV, использовать перебор параметров в цикле, или использовать другие методы

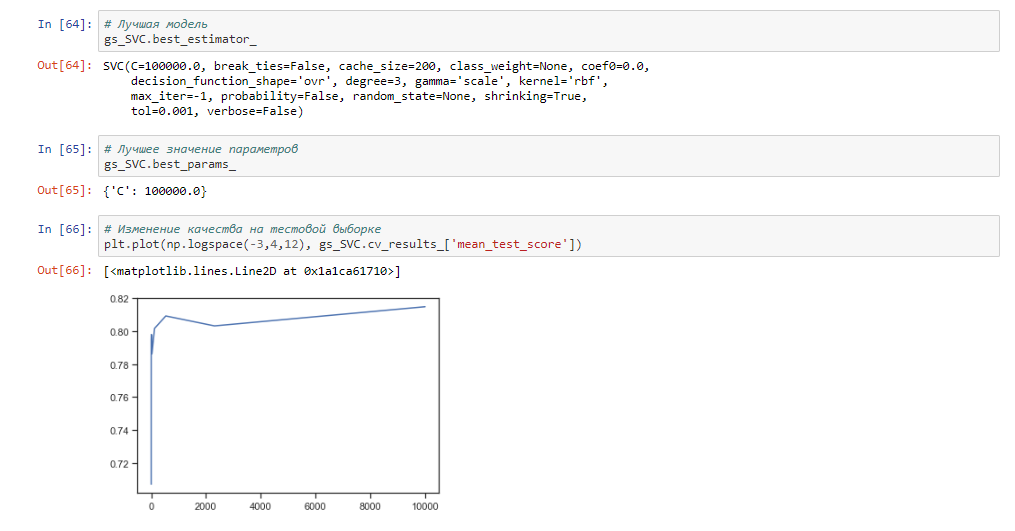
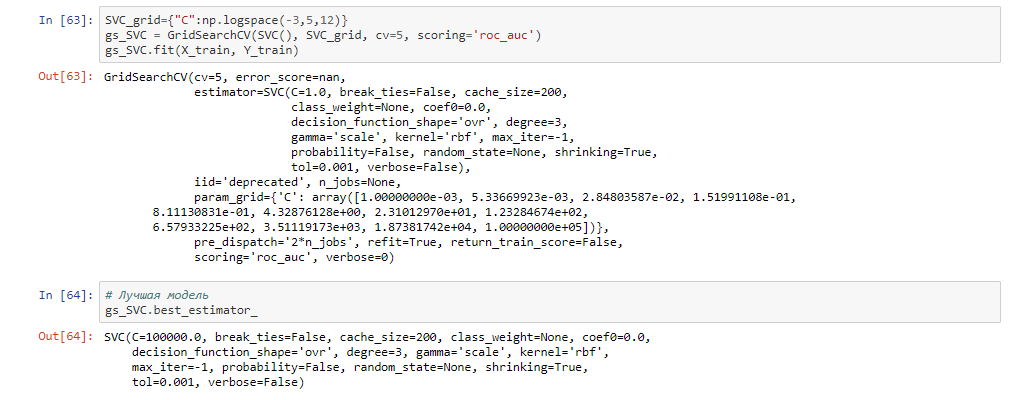
###### Метод ближайших соседей



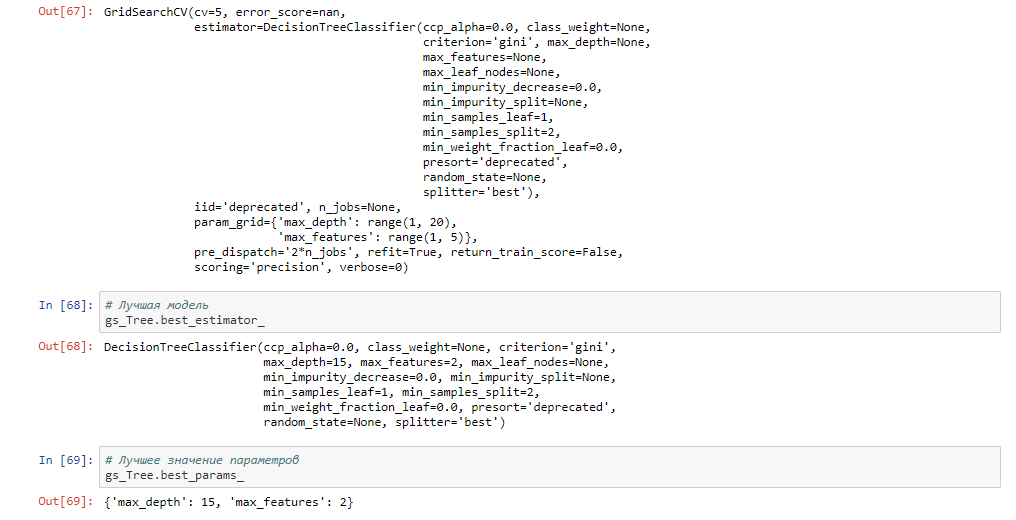
###### Логистическая регрессия



###### Машина опорных векторов

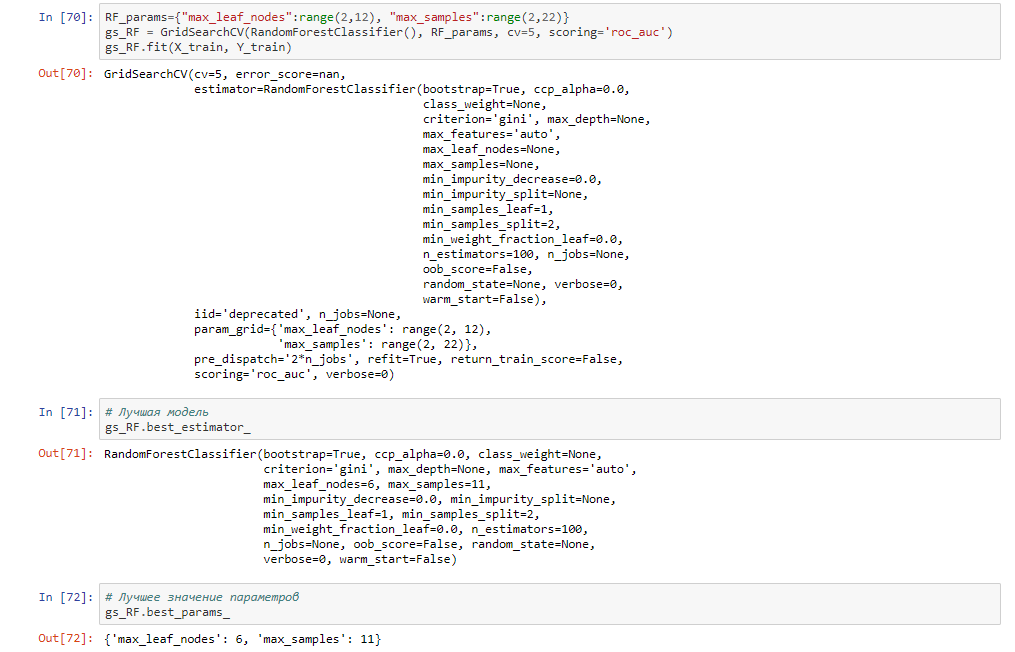


###### Решающее дерево



###### Случайный лес

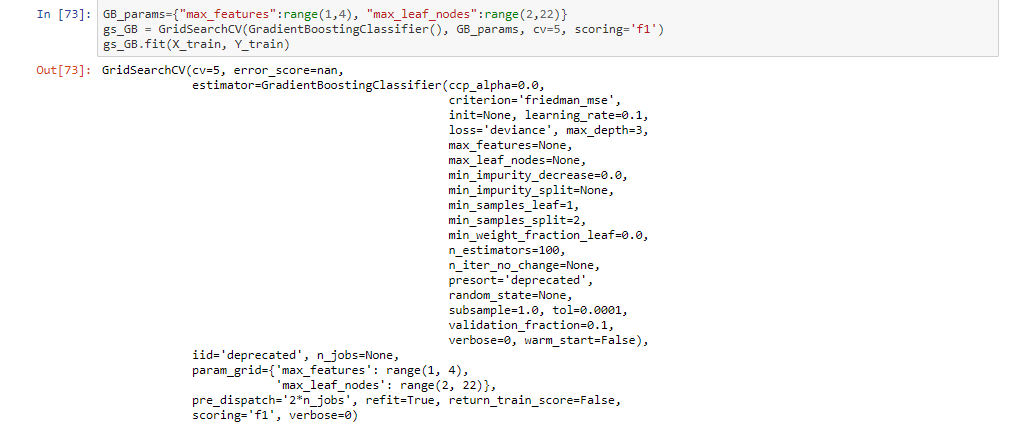
Будем использовать случайный поиск, т.к. полный решетчатый поиск работает слишком долго.

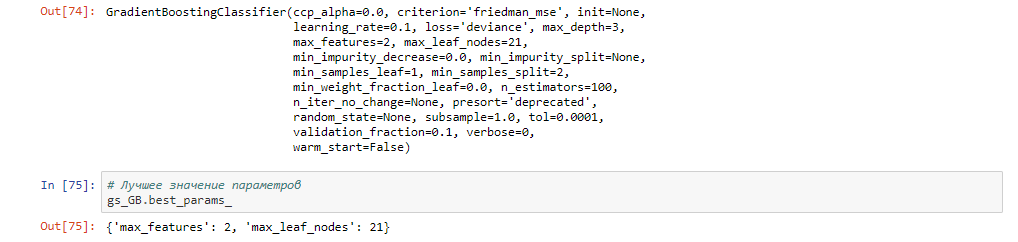


RF\_params={"max\_depth":range(1,11), "n\_estimators":range(2,400)}

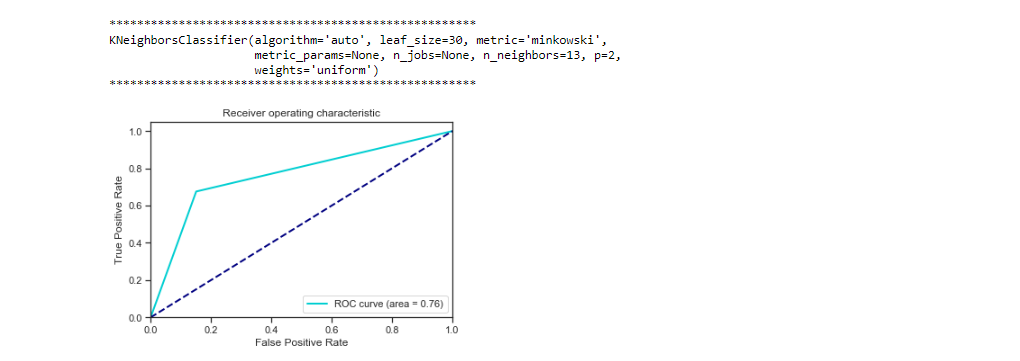
gs\_RF = RandomizedSearchCV(RandomForestClassifier(), RF\_params, cv=5, scoring='balanced\_accuracy') gs\_RF.fit(X\_train, Y\_train)

###### Градиентный бустинг

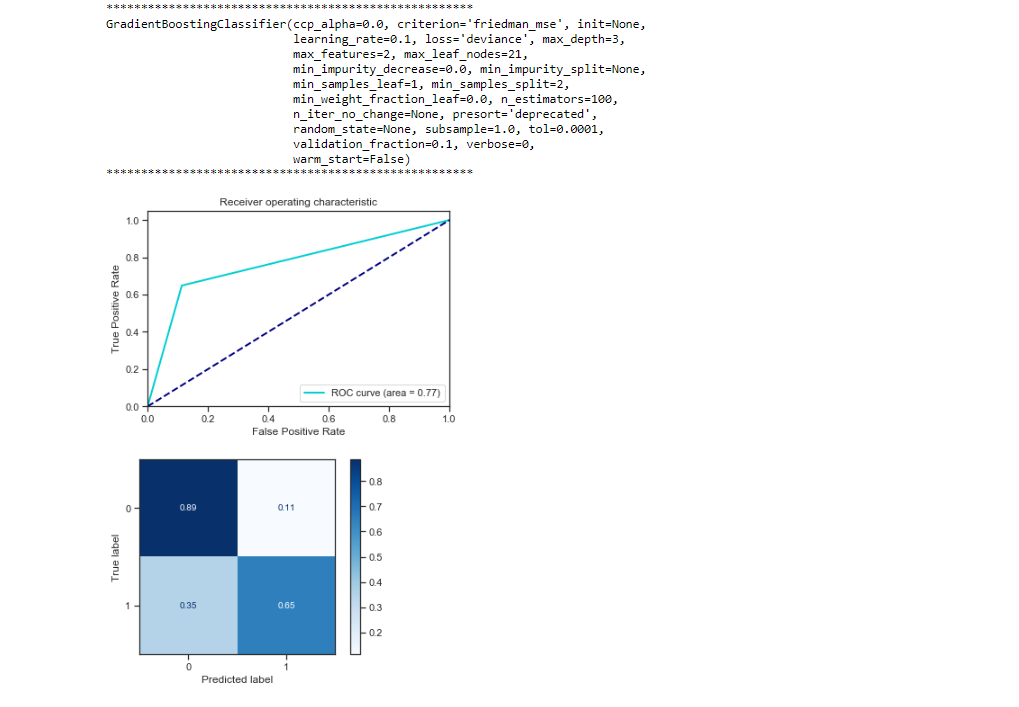
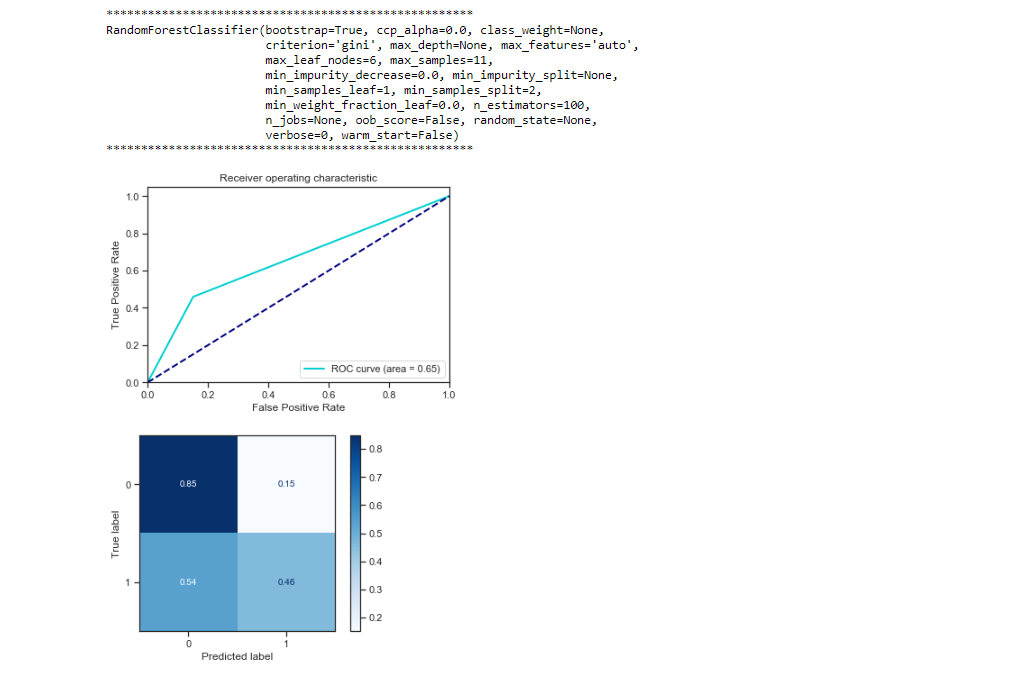
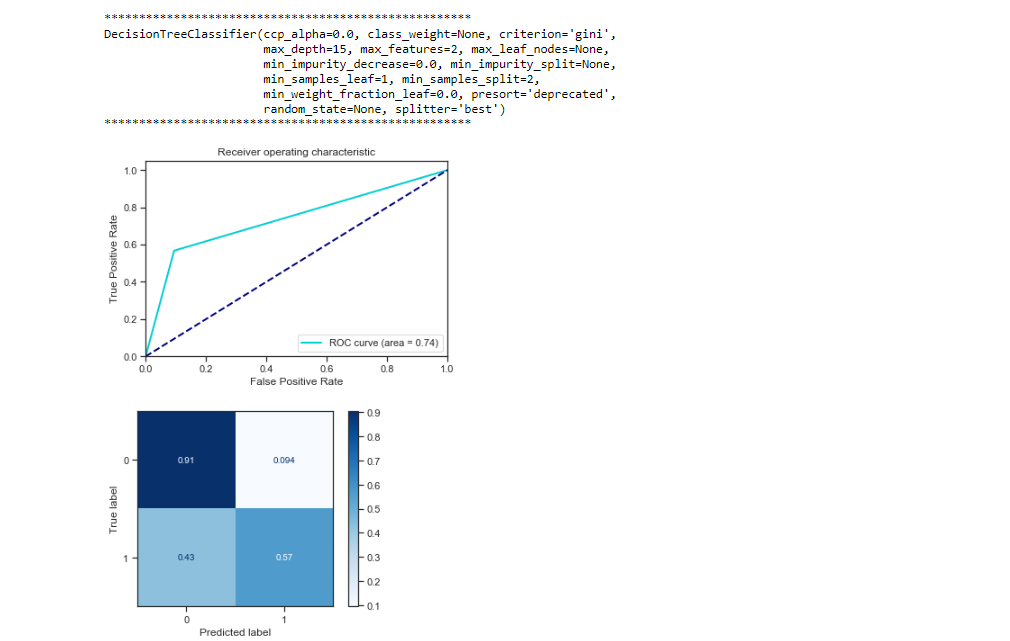
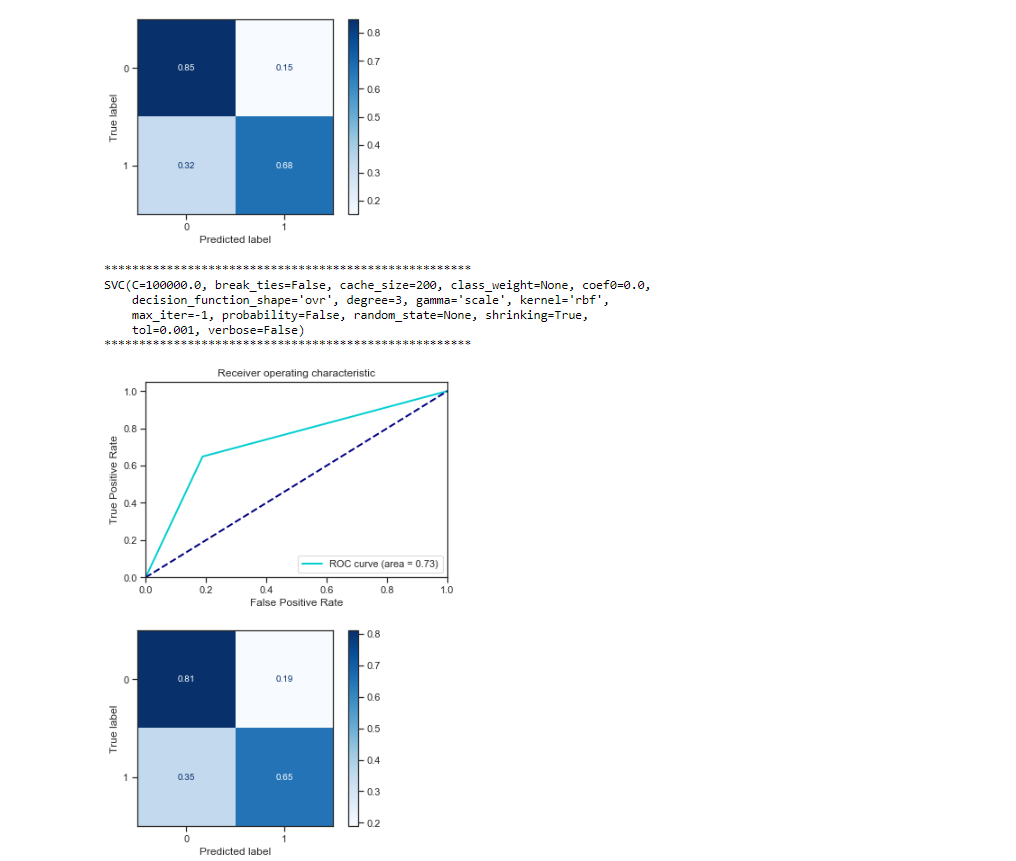




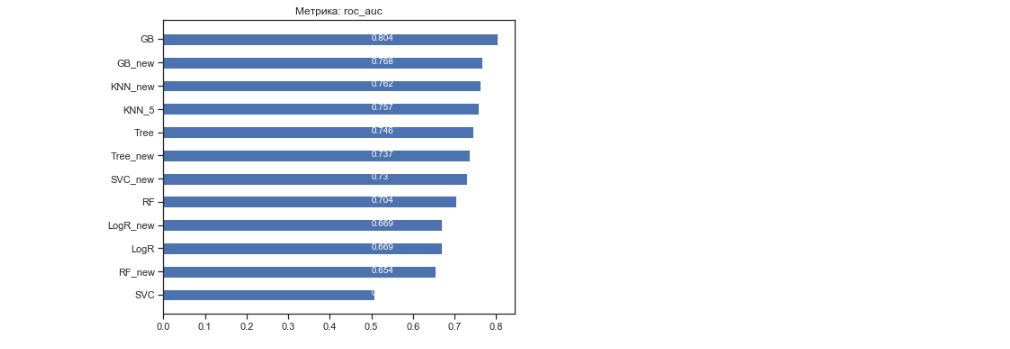
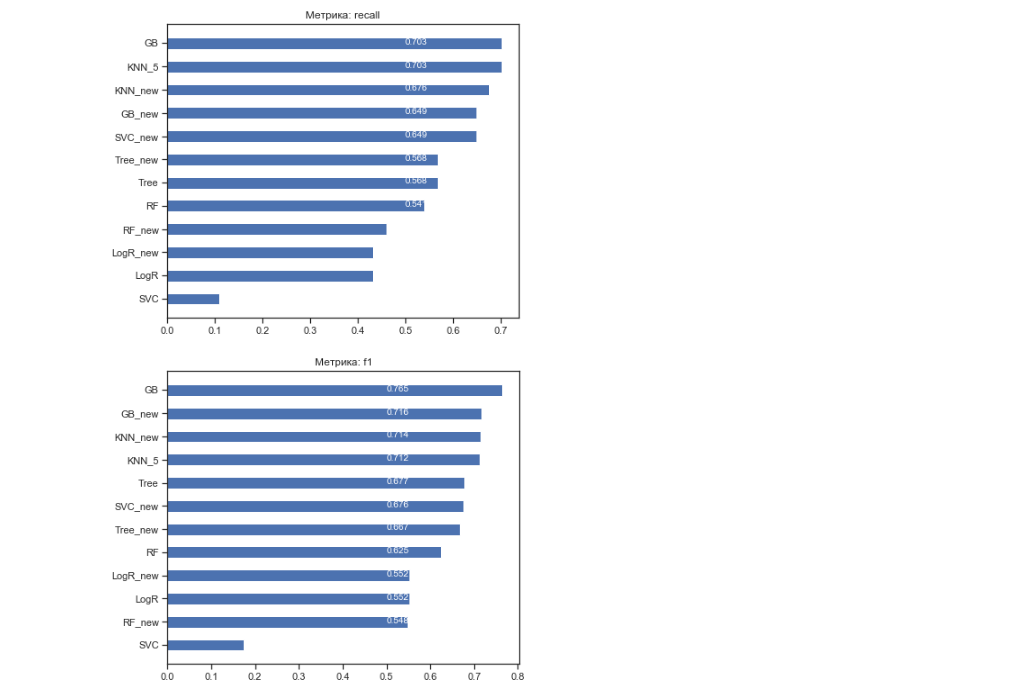
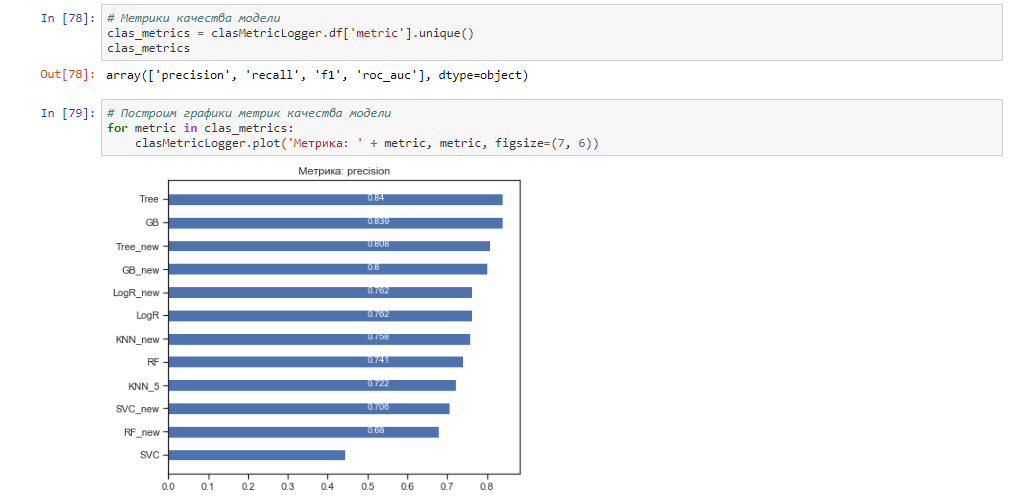
##### Повторение пункта 8 для найденных оптимальных значений гиперпараметров. Сравнение качества полученных моделей с качеством baseline-моделей



27



##### Формирование выводов о качестве построенных моделей на основе выбранных метрик. Результаты сравнения качества рекомендуется отобразить в виде графиков и сделать выводы в форме текстового описания. Рекомендуется построение графиков обучения и валидации, влияния значений гиперпарметров на качество моделей и т.д.



Таким образом, градиентный бустинг и метод ближайших соседей показывают наилучший результат.

# ЗАКЛЮЧЕНИЕ

### В данном курсовом проекте мы выполнили типовую задачу машинного обучения. В первую очередь, выбрали набор данных для построения моделей машинного обучения, провели разведочный анализ данных и построили графики, необходимые для понимания структуры данных. Далее выбрали признаки, подходящие для построения моделей, масштабировали данные и провели корреляционный анализ данных. Это позволило нам сформировать промежуточные выводы о возможности построения моделей машинного обучения.

На следующем этапе мы выбрали метрики для последующей оценки качества моделей и наиболее подходящие модели для решения задачи классификации. Затем сформировали обучающую и тестовую выборки на основе исходного набора данных и построили базовое решение для выбранных моделей без подбора гиперпараметров.

Следующим шагом был подбор гиперпараметров для выбранных моделей, после чего мы смогли сравнить качество полученных моделей с качеством baseline-моделей. Большинство моделей, для которых были подобраны оптимальные значения гиперпараметров, показали лучший результат.

В заключение, мы сформировали выводы о качестве построенных моделей на основе выбранных метрик. Для наглядности результаты сравнения качества отобразили в виде графиков, а также сделали выводы в форме текстового описания. Четыре метрики из пяти показали, что для выбранного набора данных лучшими моделями оказались случайный лес и логистическая регрессия.

# ЛИТЕРАТУРА

* 1. Ю.Е. Гапанюк, Лекции по курсу «Технологии машинного обучения», 2019 - 2020 учебный год
  2. Электронный ресурс <https://scikit-learn.org/stable/>
  3. Дж. Вандер Плас «Python для сложных задач. Наука о данных и машинное обучение»