Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования  
«Новосибирский государственный технический университет»

Кафедра теоретической и прикладной информатики

Проектная деятельность  
«Моделирование когнитивных процессов на основе машинного обучения и анализа психометрических и нейрофизиологических данных»

|  | Факультет: | ПМИ |
| --- | --- | --- |
| Группа:  Студент: | ПМ-92  Иванов В.В. |
| Преподаватель: | Авдеенко Т.В. |
|  |  |

Новосибирск

2021

# **Введение**

Цель исследования - выявление и анализ факторов, влияющих на успеваемость студентов и построение модели, способной классифицировать новые данные.

В качестве набора данных используются результаты психологического тестирования потока ПМ-8x, проведенные в 2020 году.

Мы рассмотрим пять целевых признаков - средний балл по математическим дисциплинам (**avgmath**), средний балл по прикладным дисциплинам (**avgprog**), средний балл по гуманитарным дисциплинам (**avghum**), средний балл по всем дисциплинам (**avg**) и категория успеваемости (**category**).

Для классификации будем использовать две модели, входящие в библиотеку **scikit-learn** - дерево решений (**tree.DecisionTreeClassifier**) и случайный лес (**ensemble.RandomForestClassifier**).

# **Анализ и обработка данных**

Для работы импортируем следующие модули, классы и методы:

**import pandas as pd**

**import numpy as np**

**import seaborn as sns**

**import warnings**

**from sklearn import tree**

**from matplotlib import pyplot as plt**

**from sklearn.tree import DecisionTreeClassifier**

**from sklearn.metrics import classification\_report**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.ensemble import RandomForestClassifier**

Первым шагом является чтение данных. Поместим их в два датафрейма - для предикторов и целевой переменной. Выберем в качестве целевой переменной средний балл по всем дисциплинам:

**df = pd.DataFrame(pd.read\_csv('intelligence\_tests.csv'))**

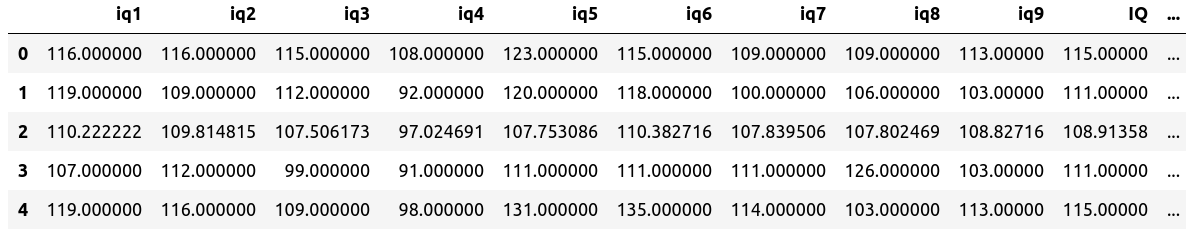
**X = df.copy().drop(['avg'], axis=1)**

**y = df.copy()['avg']**

Заменим все значения NaN средними по столбцу:

**X.fillna(np.mean(X), inplace=True)**

**y.fillna(np.mean(y), inplace=True)**



Изучим целевые переменные. Для этого изобразим данные в виде гистограммы:

**fig = plt.figure(figsize=(10, 8))**

**plt.hist(y, edgecolor='k', bins=8)**

**plt.axvline(y.median(), color='grey', linestyle='dashed', linewidth=4, label='Median')**

**plt.axvline(y.mean(), color='red', linewidth=4, label='Mean')**

**plt.legend(loc='upper left', prop={'size': 20})**

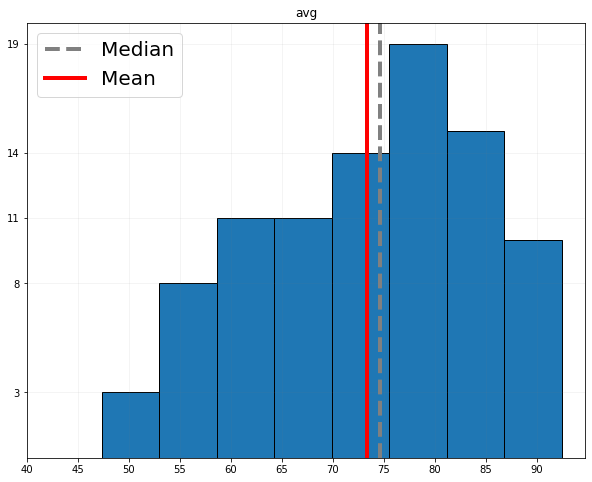
**plt.xticks([40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90])**

**plt.yticks([3, 8, 11, 14, 19])**

**plt.grid(color='grey', alpha=0.1, zorder=1)**

**plt.title('avg')**

**plt.show()**

****

Аналогично создадим переменные для признаков **avgmath**, **avgprog**, **avghum** и заполним пропущенные значения:

**y1 = df.copy()['avgmath']**

**y2 = df.copy()['avgprog']**

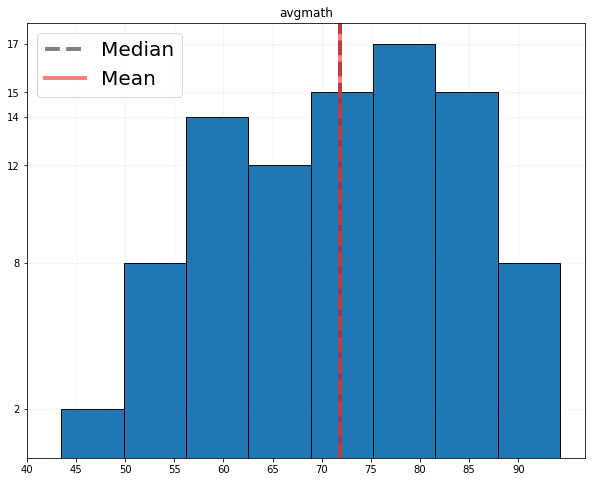
**y3 = df.copy()['avghum']**

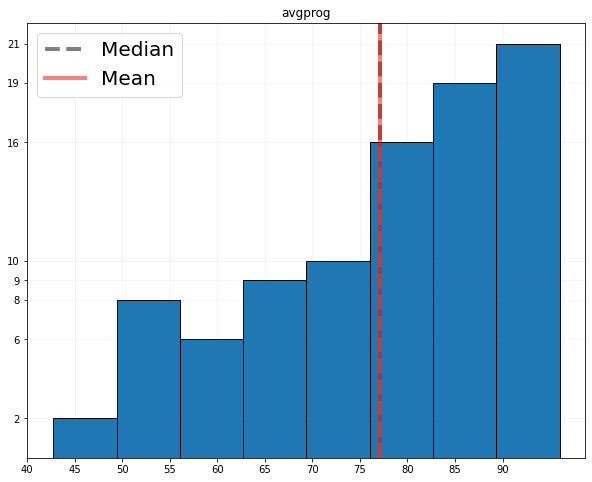
**y1.fillna(np.mean(y1), inplace=True)**

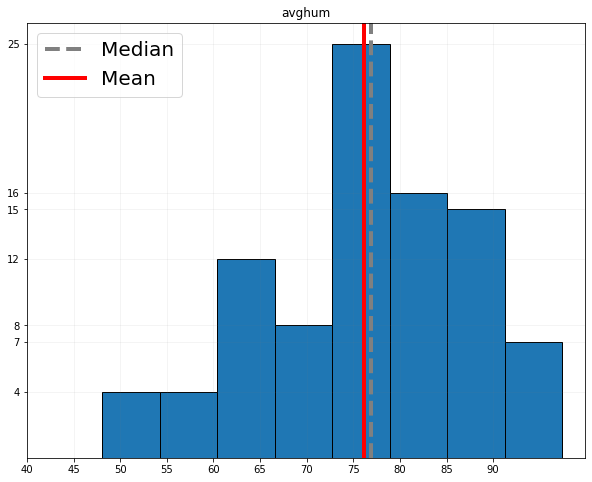
**y2.fillna(np.mean(y2), inplace=True)**

**y3.fillna(np.mean(y3), inplace=True)**

Изобразим гистограммы:







Примечательны следующие наблюдения. Во-первых, самый низкий средний балл наблюдается на первом графике, что говорит, вероятно, об относительной сложности математических дисциплин.

Во-вторых, плотность распределения на втором графике непрерывно возрастает в зависимости от среднего балла по прикладным дисциплинам. Это может быть связано с двумя факторами: характером оценивания подобных дисциплин (как правило, такие курсы разделены на большое количество практических работ, благодаря чему преподаватель имеет возможность оценить студента довольно точно) и доступности (подавляющее большинство подобных дисциплин не требует математических навыков, поэтому большое количество студентов имеет высокий балл).

Наконец, плотность распределения на третьем графике обладает заметно меньшей дисперсией, что может указывать на относительную простоту гуманитарных дисциплин (график плотности смещается вправо к центру) и в то же время на пренебрежительное отношение гуманитарными дисциплинами среди тех, кто учится на высокий балл (график плотности смещается влево к центру). Последний факт можно связать с тем, что студенты стараются уделять больше времени более важным, профильным предметам, а также с тем, что большинство гуманитарных дисциплин на факультете прикладной математики и информатики оцениваются без выставления баллов (недифференцированный зачёт) и, следовательно, не влияют на рейтинг студента и наличие стипендии.

Теперь во всех целевых переменных разделим средний балл на категории, соответствующие следующим оценкам:

≥ 93 - **A**

≥ 83 - **B**

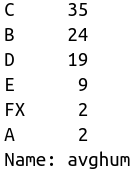
≥ 73 - **C**

≥ 63 - **D**

≥ 50 - **E**

< 50 - **FX**

Распределение студентов по категориям для разных целевых переменных:

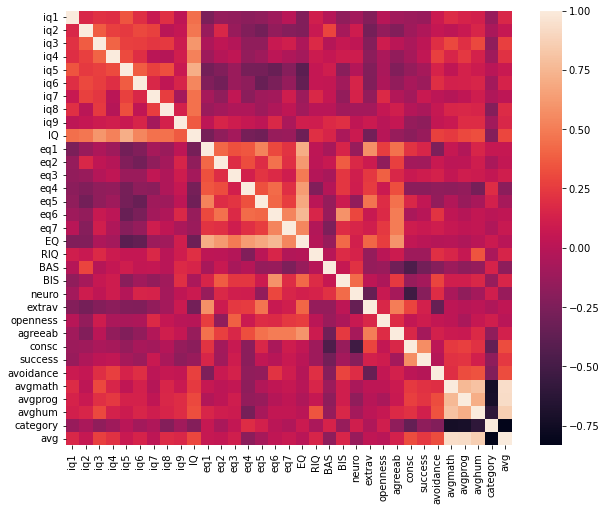




Далее необходимо проанализировать предикторы. Для выявления зависимостей довольно удобным способом является построение таблицы корреляции Пирсона с её последующей визуализацией:

**fig = plt.figure(figsize=(10, 8))**

**sns.heatmap((pd.concat([X, df['avg']], axis=1)).corr())**



Результат визуализации показывает, в частности, наличие относительно сильной корреляции между значением коэффициента интеллекта (**IQ**) и средним баллом (**avgmath**, **avgprog**, **avghum**, **avg**). Однако особый интерес вызывает категориальный признак **category**, содержащий информацию о наличии академических задолженностей у студента:

**1** - студент не имеет академических задолженностей

**2** - студент имел не более двух академических задолженностей и ликвидировал их в первые две недели

**3** - студент имеет академическую задолженность или имел более двух академических задолженностей

Рассмотрим данный признак подробнее. Для этого изобразим график, на котором по осям X отложим значения признака **category**, а по осям Y - средний балл (**avg**):

**fig = plt.figure(figsize=(10, 8))**

**plt.grid(color='grey', alpha=0.5, zorder=1)**

**plt.scatter(df['category'], df['avg'], c=df['category'], s=300, edgecolor='k', zorder=2)**

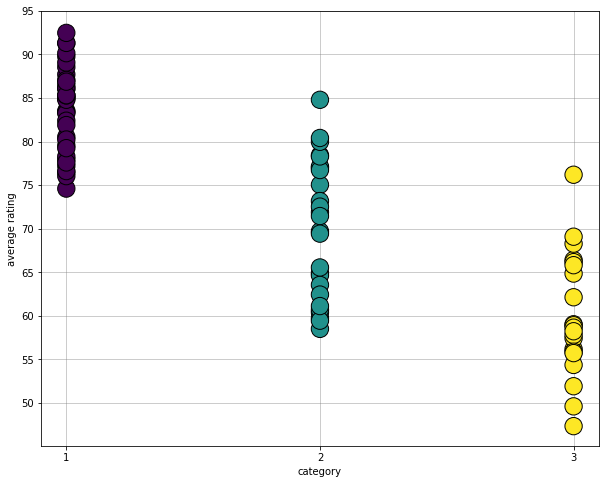
**plt.xticks([1, 2, 3])**

**plt.yticks([50, 55, 60, 65, 70, 75, 80, 85, 90, 95])**

**plt.xlabel('category')**

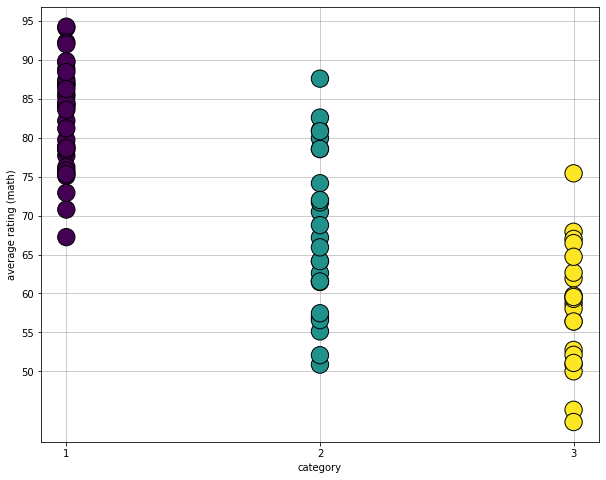
**plt.ylabel('grade')**

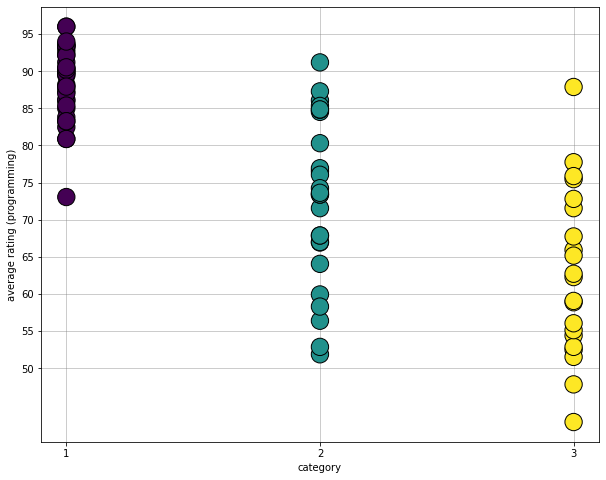
**plt.show()**

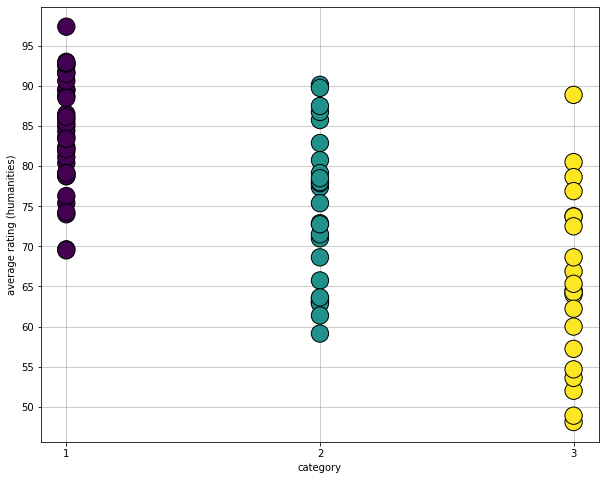


Из графика становится очевидной взаимосвязь между наличием академических задолженностей и средним баллом. Заметим, что в категорию студентов без задолженностей попали все, кто имеет среднюю оценку “хорошо” или выше.

Аналогичным образом построим графики для целевых переменных **avgmath**, **avgprog** и **avghum** соответственно:





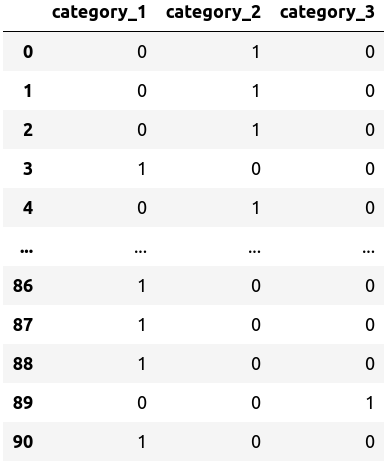


Вполне ожидаемым результатом визуализации является больший разброс объектов выборки.

# **Построение модели**

Обучим дерево решений, используя в качестве предиктора признак **category**. Для представления категориальных признаков в численном виде воспользуемся аналогом **One-Hot** кодирования в библиотеке **pandas**:

**X1 = pd.get\_dummies(X['category'], prefix='category')**



Разобьем выборку на тренировочную (для обучения модели) и тестовую (для оценивания модели), обучим дерево решений и сделаем на основе этой модели предсказания:

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X1, y, train\_size=0.7, test\_size=0.3)**

**model = DecisionTreeClassifier().fit(X\_train, y\_train)**

**y\_pred = model.predict(X\_test)**

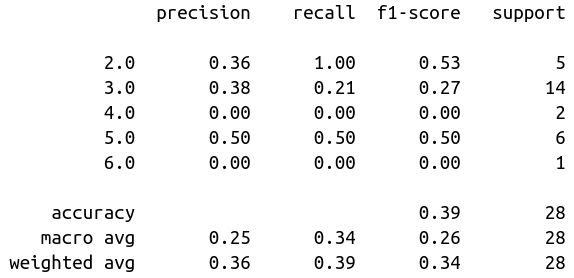
Выведем результат:

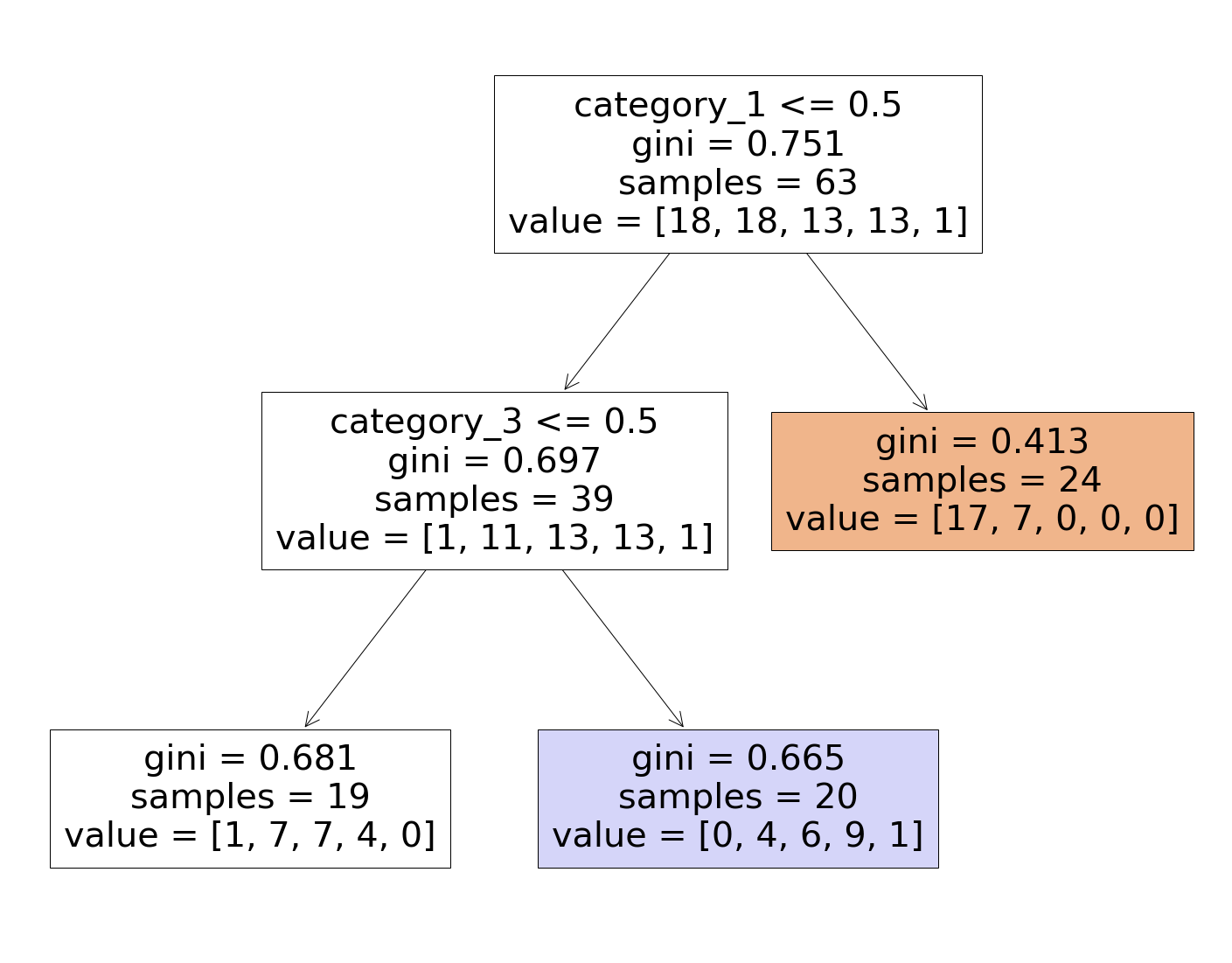
**print(classification\_report(y\_test, y\_pred))**

**fig = plt.figure(figsize=(25,20))**

**tree.plot\_tree(model, feature\_names=X1.columns, filled=True)**

**plt.show()**





Модель показала довольно плохую точность на тестовой выборке - всего **39%** объектов были классифицированы правильно.

Добавим теперь уже упомянутый предиктор **IQ** и аналогично оценим новую модель:

**X2 = pd.get\_dummies(X[['category', 'IQ']], prefix='category')**

**def modelVis(model):**

**plt.figure(figsize=(12, 8))**

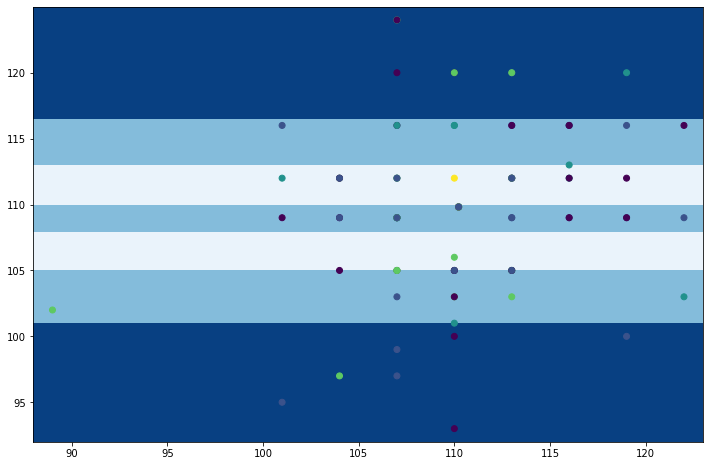
**x\_min, x\_max, y\_min, y\_max = X.iloc[:, 0].min() - 1, X.iloc[:, 0].max() + 1, X.iloc[:, 1].min() - 1, X.iloc[:, 1].max() + 1**

**xx, yy = np.meshgrid(np.arange(x\_min, x\_max, 0.01), np.arange(y\_min, y\_max, 0.01))**

**Z = model.predict(np.c\_[xx.ravel(), yy.ravel()]).reshape(xx.shape)**

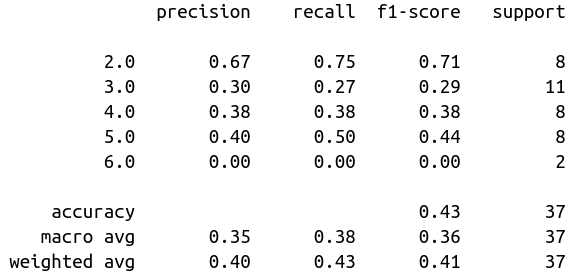
**cs = plt.contourf(xx, yy, Z, cmap='Blues')**

**plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=y)**



На изображении показано, как дерево решений отделяет объекты выборки при обучении модели.

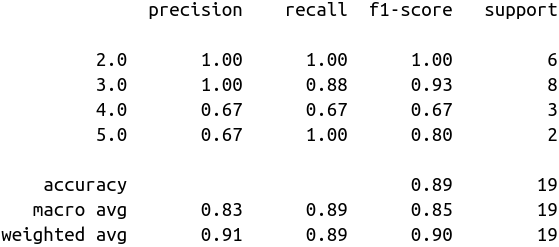
Эта модель всё ещё остается довольно примитивной. Её точность на тестовой выборке составила **43%**:



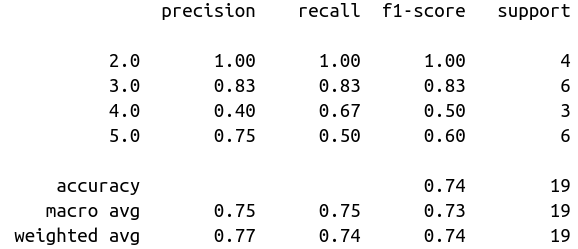
Теперь воспользуемся ансамблевым методом, называемым случайным лесом. Для начала обучим модели на признаках **avgmath**, **avgprog** и **avghum**:

**model = RandomForestClassifier(n\_estimators=100, criterion='entropy').fit(X\_train, y\_train)**

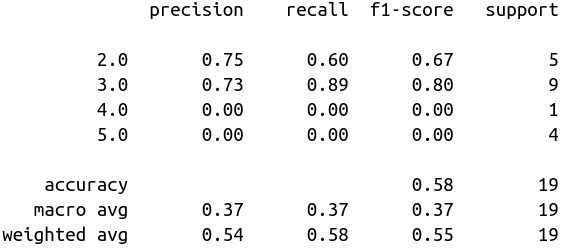
Для **avgmath**:



Для **avgprog**:



Для **avghum**:



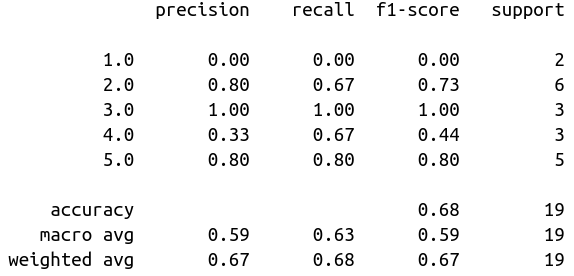
По результатам можно убедиться, что наибольший вклад в успеваемость студента вносят математические дисциплины, наименьший - гуманитарные.

Теперь обучим модели на предикторе **category**, используя в качестве целевых переменных **avgmath**, **avgprog** и **avghum**:

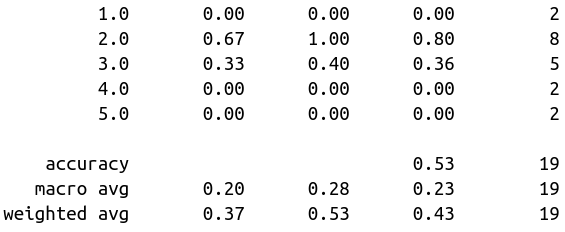
Для **avgmath**:

**X6 = pd.get\_dummies(X[['category']])**

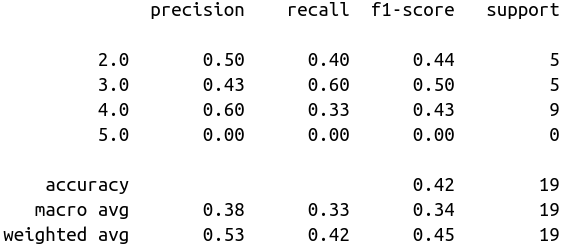
**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X6, y1, train\_size=0.8, test\_size=0.2)**



Для **avgprog**:



Для **avghum**:



Нам осталось рассмотреть в качестве целевого признака переменную **category**:

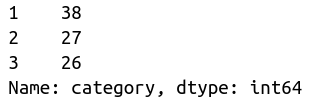
**X = df.copy().drop(['category'], axis=1)**

**y = df.copy()['category']**

**X.fillna(np.mean(X), inplace=True)**

**y.fillna(np.mean(y), inplace=True)**

**y.value\_counts()**



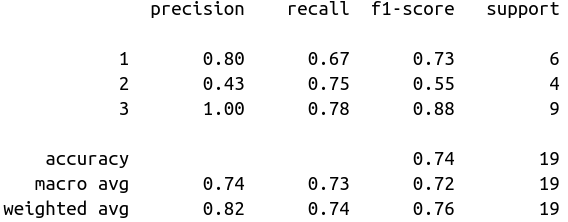
В качестве предикторов будем использовать **avg**, **avgmath**, **avgprog** и **avghum**:

Для **avg**:

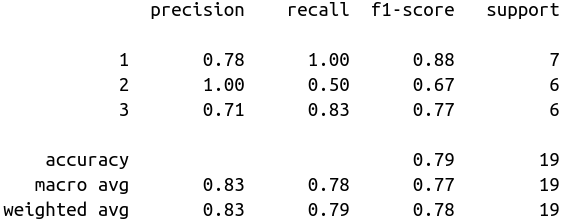
**X7 = X[['avg']]**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X1, y, train\_size=0.8, test\_size=0.2)**

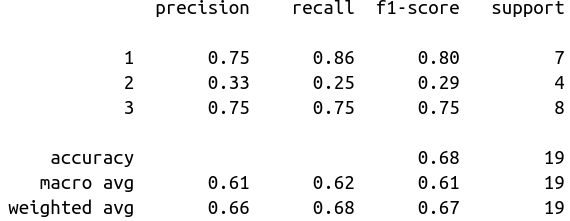
**model = RandomForestClassifier(n\_estimators=100).fit(X\_train, y\_train)**



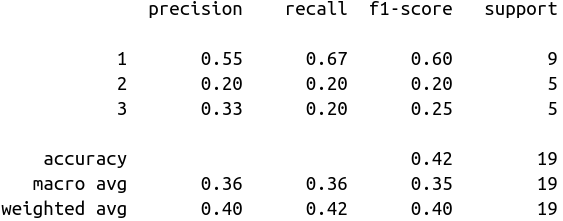
Для **avgmath**:



Для **avgprog**:



Для **avghum**:

****

# **Выводы**

В ходе исследования были получены следующие результаты:

* Выявлена зависимость между наличием академических задолженностей и средней оценкой
* Обнаружено, что наибольший вклад в средний рейтинг студента вносят математические дисциплины, в то время как наименьший - гуманитарные дисциплины
* Обнаружено, что наиболее вероятным фактором наличия академических задолженностей является низкая успеваемость по математическим дисциплинам, в том время как наименее вероятным - низкая успеваемость по гуманитарным дисциплинам
* Построена простейшая модель классификации, способная предсказывать ответы на основе новых данных