KALINGA INSTITUTE
OF INDUSTRIAL TECHNOLOGY
Deemed to be University U/S 3 of UGC Act, 1956

REPORT

Final Grade of The The Student

Submitted by
Group-5

2005784	Anish Kumar Goyal
2005828	Sayan Hazra
2005847	Alok Kumar Patel
2005927	Arpan Porel
2005939	Dyutiman Saha

TOOLS AND TECHNIQUES LAB
-Under The Guidance of Dr. Jayanti Dansana-

Table of Contents

TOPIC	PAGE NUMBER
Problem Statement	
Description of the Dataset	
Source	
Attribute Information	
Methodology	
Experimental Results -Gender proportion and their study times -Family size and Grades of the students -Rural or Urban? -Does parent's job have any effect on the students' grades? -Machine Learning -Heatmap -Models Used -Conclusion	

Problem Statement

The problem addressed in this project is the accurate prediction of final grades based on student performance data using machine learning algorithms. The goal is to develop a model that can accurately predict the final grade of a student given their performance indicators. This will involve working with a dataset that contains the attributes of 396 Portuguese students, and defining classification algorithms to identify whether a student will secure a good final grade. The project will also evaluate different machine learning models to determine the best approach for predicting final grades based on the available dataset.

Description of the Dataset

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school-related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). In [Cortez and Silva, 2008], the two data sets were modeled under binary/five-level classification and regression tasks. Important note: the target attribute G3 has a strong correlation with attributes G2 and G1.

This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. It is more difficult to predict G3 without G2 and G1, but such prediction is much more useful (see paper source for more details).

Sample Snapshot of the Dataset-

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10

Source

P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7.

Web Link : http://www3.dsi.uminho.pt/pcortez

Attribute Information

- school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
- sex - student's sex (binary: 'F' - female or 'M' - male)
- age - student's age (numeric: from 15 to 22)
- address - student's home address type (binary: 'U' - urban or 'R' - rural)
- famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
- Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
- Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - “ 5th to 9th grade, 3 - “ secondary education or 4 - “ higher education)
- Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - “ 5th to 9th grade, 3 - “ secondary education or 4 - “ higher education)
- Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
- Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
- reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
- guardian - student's guardian (nominal: 'mother', 'father' or 'other')
- traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
- studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
- failures - number of past class failures (numeric: n if 1<=n<3, else 4)
- schoolsup - extra educational support (binary: yes or no)
- famsup - family educational support (binary: yes or no)
- paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- activities - extra-curricular activities (binary: yes or no)
- nursery - attended nursery school (binary: yes or no)
- higher - wants to take higher education (binary: yes or no)
- internet - Internet access at home (binary: yes or no)
- romantic - with a romantic relationship (binary: yes or no)
- famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
- freetime - free time after school (numeric: from 1 - very low to 5 - very high)
- goout - going out with friends (numeric: from 1 - very low to 5 - very high)
- Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
- health - current health status (numeric: from 1 - very bad to 5 - very good)
- absences - number of school absences (numeric: from 0 to 93)

Methodology

Since universities are prestigious places of higher education, students' retention in these universities is a matter of high concern. It has been found that most of the students' drop-out from the universities during their first year is due to lack of proper support in undergraduate courses. Due to this reason, the first year of the undergraduate student is referred as a "make or break" year. Without getting any support on the course domain and its complexity, it may demotivate a student and can be the cause to withdraw the course.

There is a great need to develop an appropriate solution to assist students retention at higher education institutions. Early grade prediction is one of the solutions that have a tendency to monitor students' progress in the degree courses at the University and will lead to improving the students' learning process based on predicted grades.

Using machine learning with Educational Data Mining can improve the learning process of students. Different models can be developed to predict students' grades in the enrolled courses, which provide valuable information to facilitate students' retention in those courses. This information can be used to early identify students at-risk based on which a system can 1 suggest the instructors to provide special attention to those students. This information can also help in predicting the students' grades in different courses to monitor their performance in a better way that can enhance the students' retention rate of the universities.

Using various packages such as cufflinks, seaborn & matplotlib to represent the data along with different attributes graphically or pictorially to analyse the dataset for predicting the Final Grade.

In [33]:
```
import pandas as pd
df = pd.read_csv(r"https://raw.githubusercontent.com/mrsayan/final-grade-prediction/main/student-mat.csv")
df.head()
```

Out[33]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10

5 rows × 33 columns

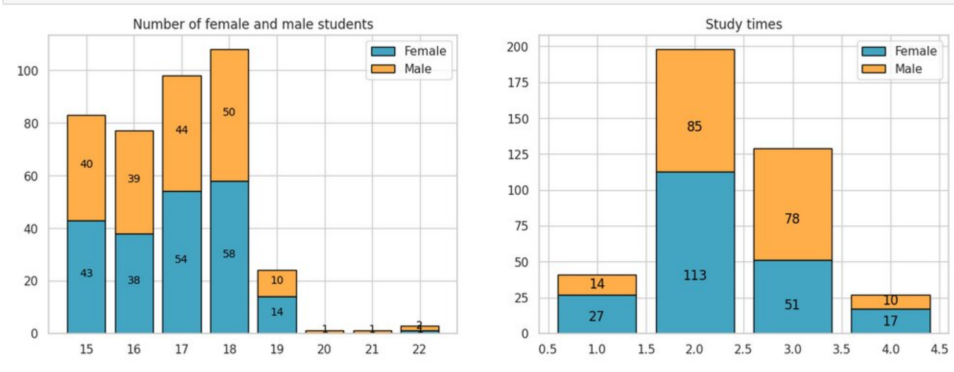
NaN Values

In [34]:
```
nan_info = dict(df.isna().sum())
def find_nan():
    for key in nan_info:
        if nan_info[key] > 0:
            return ('NaN values are present')

    return 'No NaN'
find_nan()
```

Out[34]: 'No NaN'

Gender proportion and their study times

In [35]:
```
import matplotlib.pyplot as plt

#Proportion of female and male students
#Data
ages = df['age'].unique().tolist()
female = df.loc[df['sex'] == 'F']['age'].value_counts().tolist()
female = female + [0,0]
male = df.loc[df['sex'] == 'M']['age'].value_counts().tolist()

#Stacked Bar Chart
fig,axes = plt.subplots(1,2,figsize = (15,5))

axes[0].bar(ages,female, edgecolor = 'black', color = '#44a5c2', linewidth = 1, label = 'Female')
axes[0].bar(ages,male,bottom = female, edgecolor = 'black', color = "#ffae49", linewidth = 1, label = 'Male')

axes[0].set_title('Number of female and male students')
axes[0].legend()

for bar in axes[0].patches:
    if bar.get_height()>0:
        axes[0].text(bar.get_x() + bar.get_width()/2,
                     bar.get_y() + bar.get_height()/2,
                     bar.get_height(), ha = 'center',
                     color = 'black', weight = 'ultralight', size = 10)
    else:
        pass

#Study times between male and female students
#Data
study_times = df['studytime'].unique()

male = df.loc[df['sex'] == 'M']['studytime'].value_counts()
male = male.tolist()
female = df.loc[df['sex'] == 'F']['studytime'].value_counts()
female = female.tolist()

#Stacked Bar Chart
axes[1].bar(study_times, female, edgecolor = 'black', color = "#44a5c2", linewidth = 1, label = 'Female')
axes[1].bar(study_times, male, bottom = female, edgecolor = 'black', color = '#ffae49', linewidth = 1, label = 'Male
')

axes[1].set_title('Study times')
axes[1].legend()

for bar in axes[1].patches:
    if bar.get_height()>0:
        axes[1].text(bar.get_x() + bar.get_width()/2,
                     bar.get_y() + bar.get_height()/3,
                     bar.get_height(),
                     color = 'black', weight = 'ultralight', ha = 'center')
    else:
        pass

#plt show
plt.show()
```



The number of male and female students are relatively equal but as the age is over 18 years old, their is a significant drop which indicates most students graduate from high school before they are 19 yrs.

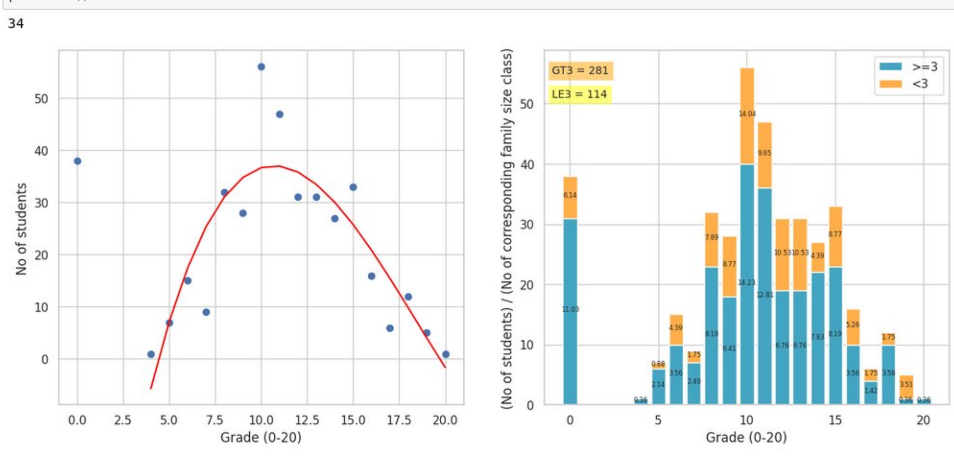
The second plot shows that female students tend to study slighty more that their male counterparts

Family size and Grades of the students

We are also intersted to know does family size and the place of living the students could have any relationship with their last year exams score which do lead to their graduation

In [36]:
```
scs = df['G3'].value_counts()
scs = dict(sorted(scs.items(), key = lambda x:x[0]))

import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures


def linear_model(ax, X,y,degree = 2):
    X, y = np.asarray(list(X)).reshape(-1,1), list(y)

    poly = PolynomialFeatures(degree = degree)
    X_poly = poly.fit_transform(X)

    lr = LinearRegression()
    lr.fit(X_poly,y)

    ax.plot(X,lr.predict(X_poly), color = 'red')
    return


fig, axes = plt.subplots(1,2,figsize = (15,6))
axes[0].scatter(scs.keys(),scs.values())

X, y = list(scs.keys())[1:], list(scs.values())[1:]
linear_model(axes[0],X, y , degree = 3)

axes[0].set_xlabel('Grade (0-20)')
axes[0].set_ylabel('No of students')


#second axis
g3_unique = df['G3'].unique()
g3_unique.sort()
g3_unique

def sort_dict(lst):
    for num in g3_unique:
        if num not in lst:
            lst[num] = 0

    lst = dict(sorted(lst.items(), key = lambda x:x[0]))
#     print(lst)
    return list(lst.values())

info_gt3 = df.loc[df['famsize'] == 'GT3']['G3'].value_counts()
info_gt3 = sort_dict(info_gt3)

info_lt3 = df.loc[df['famsize'] == 'LE3']['G3'].value_counts()
info_lt3 = sort_dict(info_lt3)

axes[1].bar(g3_unique, info_gt3, label = '>=3',color = '#44a5c2', linewidth = 1)
axes[1].bar(g3_unique, info_lt3, bottom = info_gt3, label = '<3', color = '#ffae49', linewidth = 1)

axes[1].legend()


i = 0
for bar in axes[1].patches:
    if bar.get_height()>0:
        if i <=17:
            axes[1].text(bar.get_x() + bar.get_width()/2,
                         bar.get_y() + bar.get_height()/2,
                         round((bar.get_height()/281)*100,2),ha = 'center',fontsize = 6)
        else:
            axes[1].text(bar.get_x() + bar.get_width()/2,
                         bar.get_y() + bar.get_height()/2,
                         round((bar.get_height()/114)*100,2), ha = 'center',fontsize = 6)

        i+=1
    else:
        pass
print(i)


axes[1].set_xlabel('Grade (0-20)')
axes[1].set_ylabel('(No of students) / (No of corresponding family size class)')

axes[1].text(-1, 55, 'GT3 = 281', fontsize = 10,
         bbox = dict(facecolor = 'orange', alpha = 0.5))
axes[1].text(-1, 51, 'LE3 = 114', fontsize = 10,
         bbox = dict(facecolor = 'yellow', alpha = 0.5))

plt.show()
```

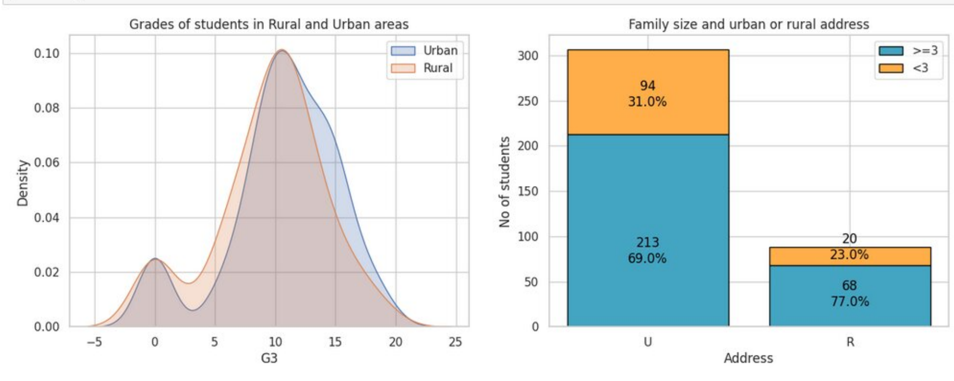
34



We can conclude the less the family size is, the better the grades are. Because as it is plotted, the students with lower family size have better grades compared to other with higher family size and it becomes more obvious if we look at better and higher grades when higher percentage of *LE3 Famsize* have received better grades.

Rural or Urban?

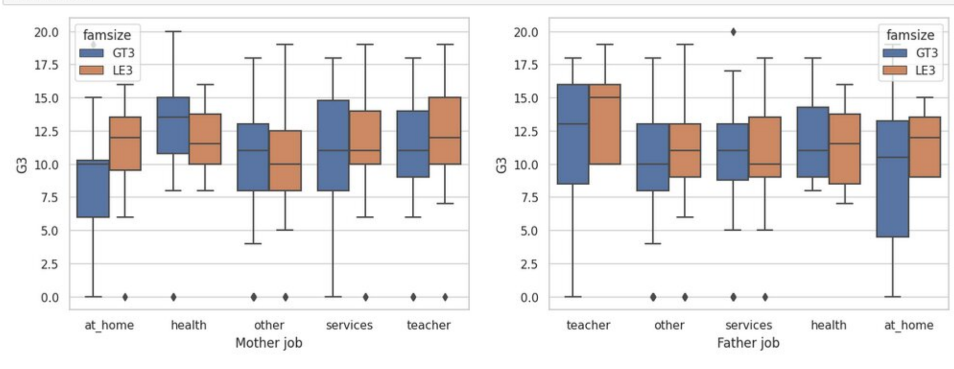
In [37]:
```
import seaborn as sns

fig, axes = plt.subplots(1,2,figsize = (15,5))

#First ax
sns.kdeplot(data = df.loc[df['address'] == 'U'], x = 'G3', shade = True, label = 'Urban', ax = axes[0])
sns.kdeplot(data = df.loc[df['address'] == 'R'], x = 'G3', shade = True, label = 'Rural',ax = axes[0])

axes[0].legend()
axes[0].set_title('Grades of students in Rural and Urban areas')

#second ax
ads = df['address'].unique().tolist()

rural_gt3 = df.loc[(df['address'] == 'R') & (df['famsize'] == 'GT3')]['famsize'].size
urban_gt3 =  df.loc[(df['address'] == 'U') & (df['famsize'] == 'GT3')]['famsize'].size

rural_le3 = df.loc[(df['address'] == 'R') & (df['famsize'] == 'LE3')]['famsize'].size
urban_le3 =  df.loc[(df['address'] == 'U') & (df['famsize'] == 'LE3')]['famsize'].size

axes[1].bar(ads,[urban_gt3, rural_gt3], label = '>=3',color = '#44a5c2',
            linewidth = 1, edgecolor = 'black')
axes[1].bar(ads,[urban_le3, rural_le3],bottom = [urban_gt3, rural_gt3],label = "<3",
            color = '#ffae49', linewidth = 1, edgecolor = 'black')
axes[1].legend()

info = dict(df['address'].value_counts())
i = 0
for bar in axes[1].patches:
    if i%2==0:
        txt = "{}\n{}%".format(bar.get_height(),round(bar.get_height()/info['U'],2)*100)
        axes[1].text(bar.get_x() + bar.get_width()/2,
                     bar.get_y() + bar.get_height()/3,
                    txt, ha = 'center', color = 'black')
    else:
        txt = "{}\n{}%".format(bar.get_height(),round(bar.get_height()/info['R'],2)*100)
        axes[1].text(bar.get_x() + bar.get_width()/2,
                     bar.get_y() + bar.get_height()/3,
                     txt, ha = 'center', color = 'black')
    i+=1

axes[1].set_title('Family size and urban or rural address')
axes[1].set_xlabel('Address')
axes[1].set_ylabel('No of students')

plt.show()
```



There is no clear relationship that shows being urban and rural means higher or lower grades.
Also ost of the families in both rural and urban parts have more than 3 person in their family but when it comes to less than 3 persons, urban people have less family count as it is shown, 23% of rural people have family size less than 3 people but 30% urban counterparts have less than 3 people in their family and as shown before, the students coming from lower family size have achieved better grades than the ones coming from higher family size

Does parent's job have any effect on the students' grades?

In [38]:
```
fig, axes = plt.subplots(1,2, figsize = (15,5))

sns.set(style="whitegrid")

sns.boxplot(data = df, x = 'Mjob', y = 'G3',ax = axes[0],hue = 'famsize')
axes[0].set_xlabel('Mother job')

sns.boxplot(data = df, x = 'Fjob', y = 'G3',ax = axes[1], hue = 'famsize')
axes[1].set_xlabel('Father job')

plt.show()
```



As expected, students with lower family size tend to have higher scores at their last year exams but more interestingly, the students whose father job is teacher have scored better and even higher when the family size is less than 3. Also students whose mother job is in health sector have came up with better grades.

It is worth to notice that in both cases when either the father or mother job in remote(at home) or maybe the parent is jobless , students have achieved lower grades. It suggests than parent's job has a significant meaning for the student performance.

Machine learning

In [39]:
```
from sklearn.preprocessing import LabelEncoder

categorical = df.select_dtypes(include = 'object').columns
encoders = []
for column in categorical:
    le = LabelEncoder()
    le.fit(df[column])
    df[column] = le.transform(df[column])
    encoders.append(le)

df.head()
```

Out[39]:

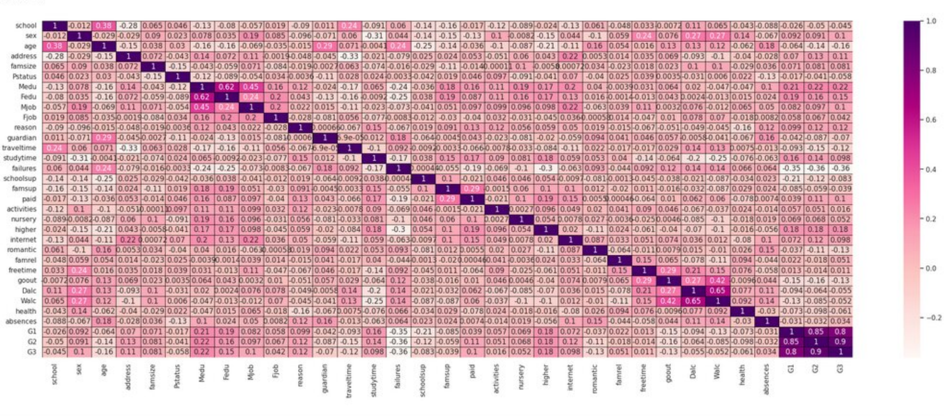
	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	0	0	18	1	0	0	4	4	0	4	...	4	3	4	1	1	3	6	5	6	6
1	0	0	17	1	0	1	1	1	0	2	...	5	3	3	1	1	3	4	5	5	6
2	0	0	15	1	1	1	1	1	0	2	...	4	3	2	2	3	3	10	7	8	10
3	0	0	15	1	0	1	4	2	1	3	...	3	2	2	1	1	5	2	15	14	15
4	0	0	16	1	0	1	3	3	2	2	...	4	3	2	1	2	5	4	6	10	10

5 rows × 33 columns

Heatmap

In [40]:
```
plt.figure(figsize = (30,10))
sns.heatmap(df.corr(),cmap = 'RdPu',annot = True,linewidth = 0.5,linecolor = 'black')
```

Out[40]: <Axes: >



In [41]:
```
columns = []
info = dict(df.corr()['G3'])
for column in info:
    if abs(info[column]) > 0.05 and column != 'G3':
        columns.append(column)

from sklearn.model_selection import train_test_split

X,y = df[columns], df['G3']
xtrain, xtest, ytrain, ytest = train_test_split(X,y, test_size = 0.25, random_state = 25)
```

Models

In [42]:
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import ElasticNet
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor, ExtraTreesRegressor
from sklearn.svm import SVR

models = [LinearRegression(), ElasticNet(), RandomForestRegressor(),
          ExtraTreesRegressor(),GradientBoostingRegressor(),SVR()]

test_scores = {}
train_scores = dict()
for model in models:
    model.fit(xtrain, ytrain)

    test_score = round(model.score(xtest,ytest),3)
    train_score = round(model.score(xtrain,ytrain),3)

    test_scores[type(model).__name__] = test_score
    train_scores[type(model).__name__] = train_score
```

In [43]:
```
#Cross Validation
from sklearn.model_selection import KFold, cross_val_score
from statistics import mean

k_folds = KFold(n_splits = 5)
cross_val_scores = []
for model in models:
    scores_kfold = cross_val_score(model,X,y, cv = k_folds)
    scores_kfold = mean(scores_kfold)
    cross_val_scores.append(round(scores_kfold,3))

print(cross_val_scores)
```

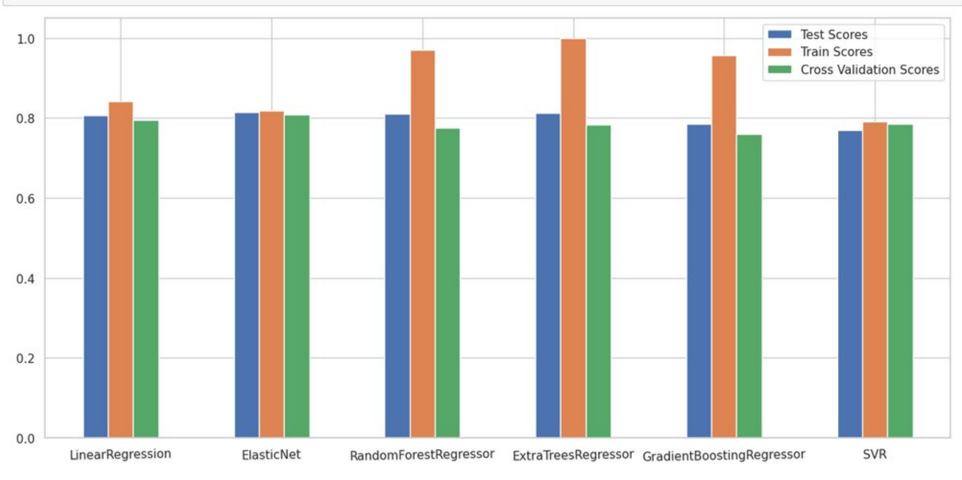
```
[0.796, 0.809, 0.776, 0.783, 0.76, 0.785]
```

In [44]:
```
scores = {'Test Scores':list(test_scores.values()),
          'Train Scores':list(train_scores.values()),
          'Cross Validation Scores':cross_val_scores}

scores_df = pd.DataFrame(scores,index = [type(model).__name__ for model in models])
scores_df
```

Out[44]:

	Test Scores	Train Scores	Cross Validation Scores
LinearRegression	0.807	0.841	0.796
ElasticNet	0.814	0.818	0.809
RandomForestRegressor	0.811	0.971	0.776
ExtraTreesRegressor	0.813	1.000	0.783
GradientBoostingRegressor	0.786	0.957	0.760
SVR	0.769	0.791	0.785

In [45]:
```
scores_df.plot.bar(figsize = (15,7),rot = True)
plt.show()
```



Considering the cross validation scores into account, the best models are ***ElasticNet*** and ***Linear Regression***