

Evaluation Student

شركاء النجاح

الشركاء الاستراتيجيين

الشركاء الذهبيين

الشريك الماسي

Evaluation Student

فريق العمل

سارة الناصر

ندى النجار

ريم الغامدي

أسماء الرضيان

شركاء النجاح

الشركاء الاستراتيجيين

الشركاء الذهبيين

الشريك الماسي

جدول المحتويات :

1- المشكلة

2- الأهداف

3- التساؤلات

4- خطوات عمل المشروع

- دراسة البيانات

- تحديد العوامل المهمة

- استخدام الخوارزميات للتنبؤ بنتائج الطلاب.

5- النتيجة والخاتمة

المشكلة:

لاحظت المدرسة من زيادة نسبة الرسوب في مادتين الرياضيات والانقلش، لذلك تم جمع البيانات تتعلق بالطلاب وأولياء الأمور، لتحديد أسباب الرسوب .

الأهداف:

- تحديد العوامل المهمة التي تحدد مستوى الطلاب.
- بناء نماذج توقعية لنتائج الطلاب بداية العام ليتم مساعدة من يتوقع رسوبهم.

التساؤلات:

هل كان لنسبة الذكور والإناث علاقة بدرجات الطلبة؟

هل هناك علاقة بتعليم الوالدين على درجات الطلبة؟

هل يؤثر غياب الطلبة على نسبة الرسوب؟

الإناث



الذكور



خطوات عمل المشروع:

- دراسة البيانات المقدمة من المدرسة.
- تحديد العوامل الأكثر تأثير في رسوب الطلاب ونجاحهم.
- استخدام نماذج للتنبؤ بنتائج الطلاب.

دراسة البيانات:

```
df.groupby(['freetime'])['G3'].describe()
```

	count	mean	std	min	25%	50%	75%	max
freetime								
1	64.0	11.171875	3.401935	0.0	10.0	11.0	13.0	19.0
2	171.0	12.280702	3.790488	0.0	11.0	13.0	15.0	20.0
3	408.0	11.183824	3.965209	0.0	10.0	11.0	14.0	19.0
4	293.0	11.208191	3.662238	0.0	10.0	11.0	14.0	19.0
5	108.0	10.916667	4.218617	0.0	9.0	11.0	14.0	19.0

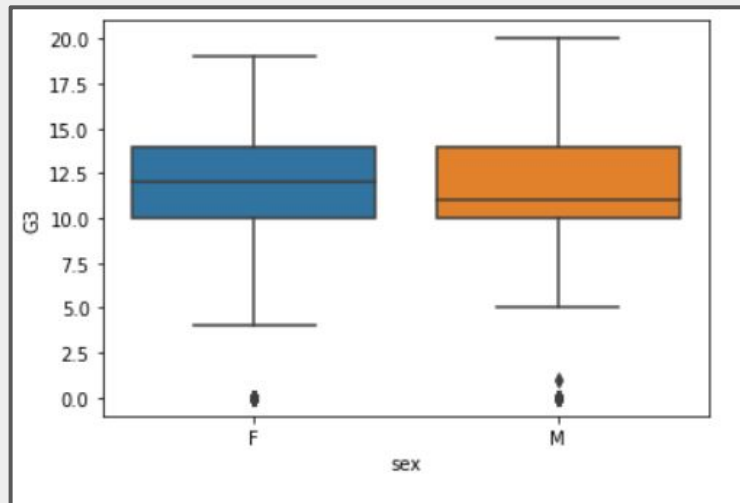
علاقة الدرجات مقارنة بالغيابات لدى الطلبة

BoxPlot:

دراسة البيانات:



```
sns.boxplot(x='sex',y='G3',data=df)
```



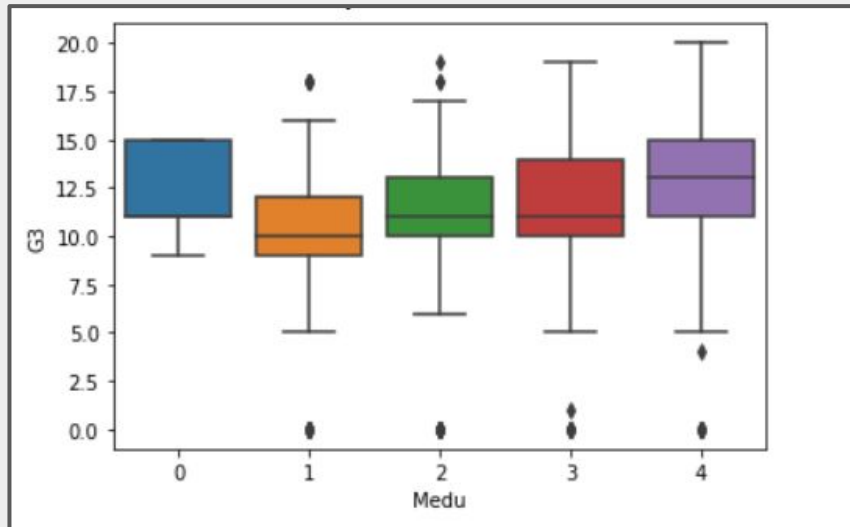
علاقة الدرجات مقارنة بالذكور والإناث

BoxPlot:

دراسة البيانات:



```
sns.boxplot(x='Medu', y='G3', data=df)
```



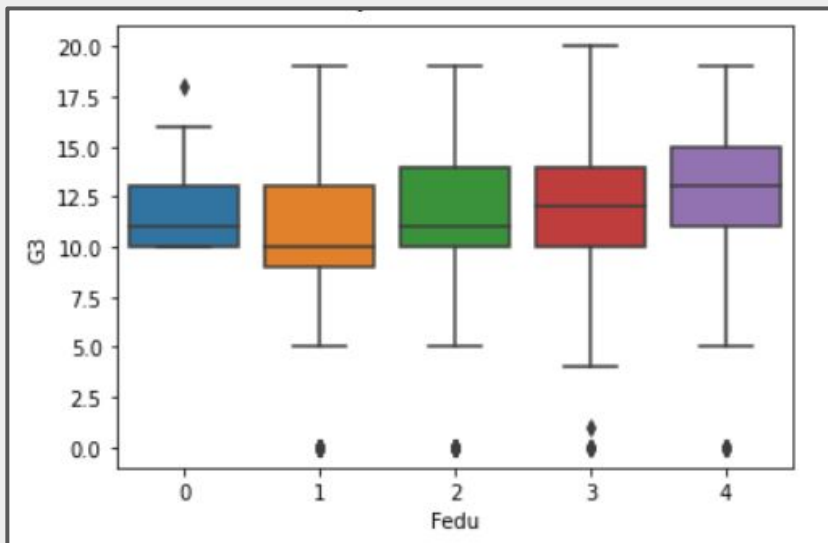
علاقة الدرجات بتعليم الأم

BoxPlot:

دراسة البيانات:



```
sns.boxplot(x='Fedu', y='G3', data=df)
```



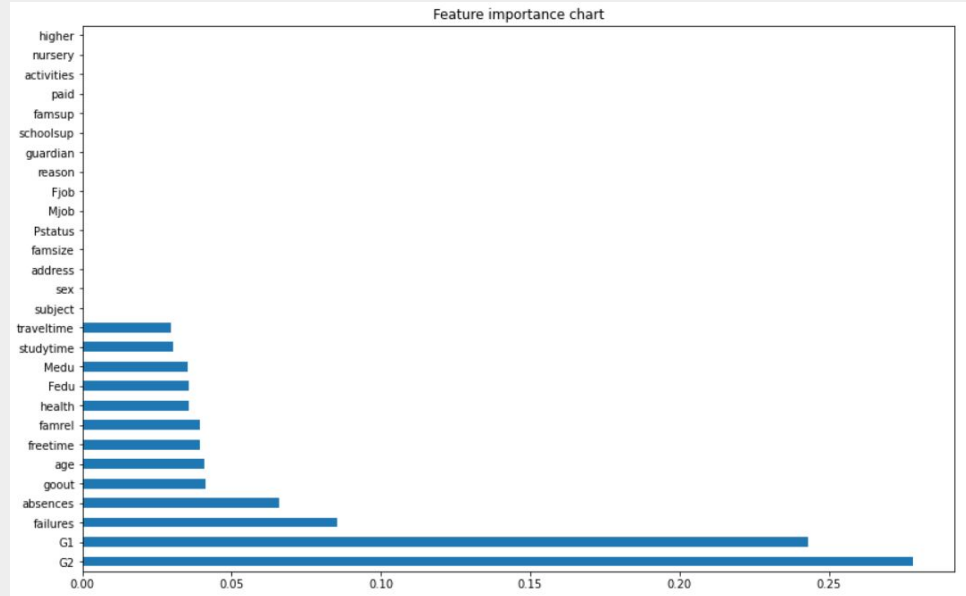
علاقة الدرجات بتعليم الاب

تحديد العوامل المهمة:

```
vif_data = pd.DataFrame()
vif_data["feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                    for i in range(len(X.columns))]
```

feature	VIF
subject	0.000000
sex	0.000000
age	1.137050
address	0.000000
famsize	0.000000
Pstatus	0.000000
Medu	1.792043
Fedu	1.739565
Mjob	0.000000
Fjob	0.000000
reason	0.000000
guardian	0.000000
traveltime	1.080244
studytime	1.077151
failures	1.300958
schoolsup	0.000000
famsup	0.000000
paid	0.000000
activities	0.000000
nursery	0.000000
higher	0.000000
internet	0.000000
famrel	1.041758
freetime	1.162428
goout	1.159298
health	1.037526
absences	1.055896
G1	3.920179
G2	3.919086

```
model = ExtraTreesClassifier()
model.fit(X,y)
plt.figure(figsize=(14,9))
plt.title("Feature importance chart")
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(28).plot(kind='barh')
```



Linear Regression:

- بعد دراسة البيانات تم تحديد أهم الحقول المؤثرة على تقييم النموذج
مثل : الدرجات ،العمر تعليم الام والاب الصحة وقت الفراغ الغياب ..

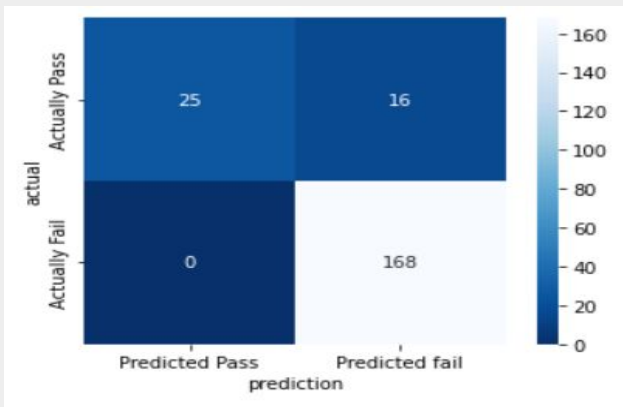
```
regr.fit(x_train, y_train)
print(regr.score(x_train, y_train))
regr.fit(x_val, y_val)
print(regr.score(x_val, y_val))
regr.fit(x_test, y_test)
print(regr.score(x_test, y_test))
```

```
0.8429240559060498
0.8542122636691802
0.821815089059527
```

Train	84
Test	82

Logistic Regression:

```
Logistic.fit(X_train , Y_train)
print('f1_score for train: ',f1_score(Logistic.predict(X_train), Y_train))
print('f1_score for validation: ',f1_score(Y_val, Logistic.predict(X_val)))
print('f1_score for test: ',f1_score(Logistic.predict(X_test), Y_test))
print("Logistic confusion matrix: \n\n", confusion_matrix(Y_train, Logistic.predict(X_train)))
print("Logstic precision: \n\n", precision_score(Y_train, Logistic.predict(X_train)))
print("Logstic recall: \n\n", recall_score(Y_train, Logistic.predict(X_train)))
print("Logistic confusion matrix: \n\n", confusion_matrix(Y_val, Logistic.predict(X_val)))
print("Logstic precision: \n\n", precision_score(Y_val, Logistic.predict(X_val)))
print("Logstic recall: \n\n", recall_score(Y_val, Logistic.predict(X_val)))
print("Logistic confusion matrix: \n\n", confusion_matrix(Y_test, Logistic.predict(X_test)))
print("Logstic precision: \n\n", precision_score(Y_test, Logistic.predict(X_test)))
print("Logstic recall: \n\n", recall_score(Y_test, Logistic.predict(X_test)))
```

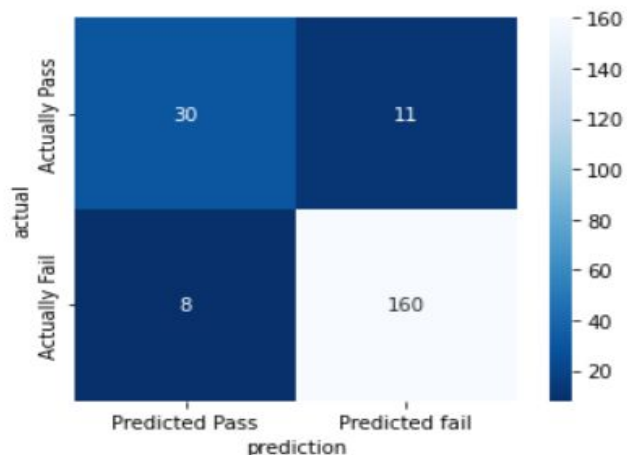


92%	Accuracy
60%	Sensitivity
100%	Specificity

:Random Forest

```
print('f1_score for training: ',f1_score(Y_train, Random_Forest.predict(X_train)))
print('f1_score for validation: ',f1_score(Random_Forest.predict(X_val), Y_val))
print('f1_score for test: ',f1_score(Random_Forest.predict(X_test), Y_test))
print("Random_Forest precision: \n\n", precision_score(Y_train,
Random_Forest.predict(X_train)))
print("Random_Forest precision: \n\n", precision_score(Y_val, Random_Forest.predict(X_val)))
print("Random_Forest precision: \n\n", precision_score(Y_test, Random_Forest.predict(X_test)))

print("Random_Forest recall: \n\n", recall_score(Y_train, Random_Forest.predict(X_train)))
print("Random_Forest recall: \n\n", recall_score(Y_val, Random_Forest.predict(X_val)))
print("Random_Forest recall: \n\n", recall_score(Y_test, Random_Forest.predict(X_test)))
```



90%

Accuracy

73%

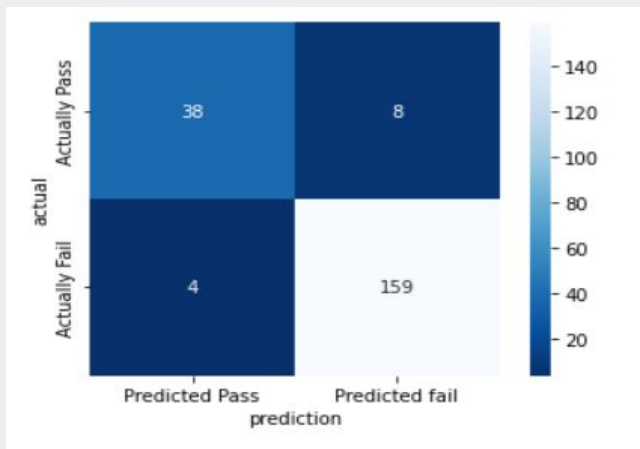
Sensitivity

95%

Specificity

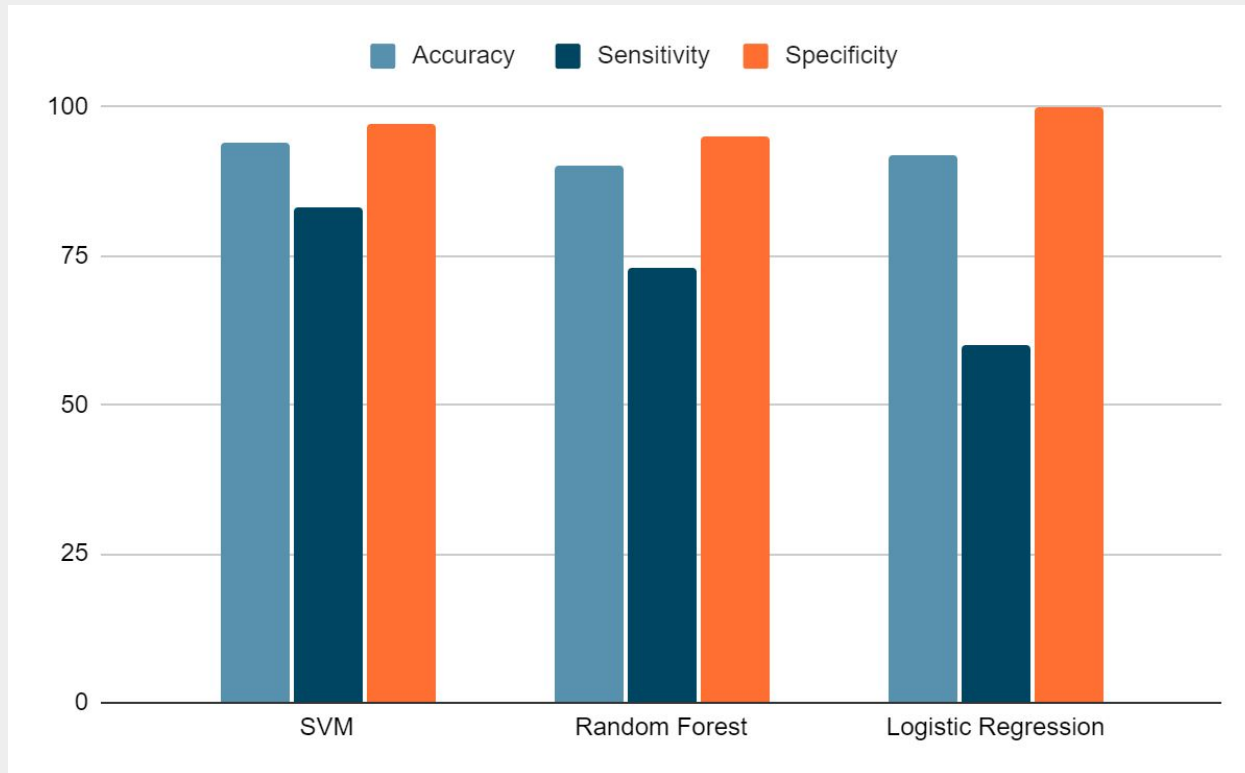
:Support Vector Machine

```
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
svc = SVC(C=1.0, random_state=1, kernel='linear')
svc.fit(X_train_std, y_train)
svc = SVC(C=1.0, random_state=1, kernel='linear')
svc.fit(X_train_std, y_train)
y_predict = svc.predict(X_test_std)
print("Accuracy score %.3f" %metrics.accuracy_score(y_test, y_predict))
```



94%	Accuracy
83%	Sensitivity
97%	Specificity

النتائج:



التوصيات والخاتمة:

- إضافة بيانات حول معلمي المواد ومستوى تعليمهم .
- إضافة بيانات حول طرق التدريس للمواد و الواجبات المدرسية والأنشطة الدراسية .
- إضافة بيانات حول طرق التدريس للمواد .

شكرا لكم

شركاء النجاح

الشركاء الاستراتيجيين

الشركاء الذهبيين

الشريك الماسي