

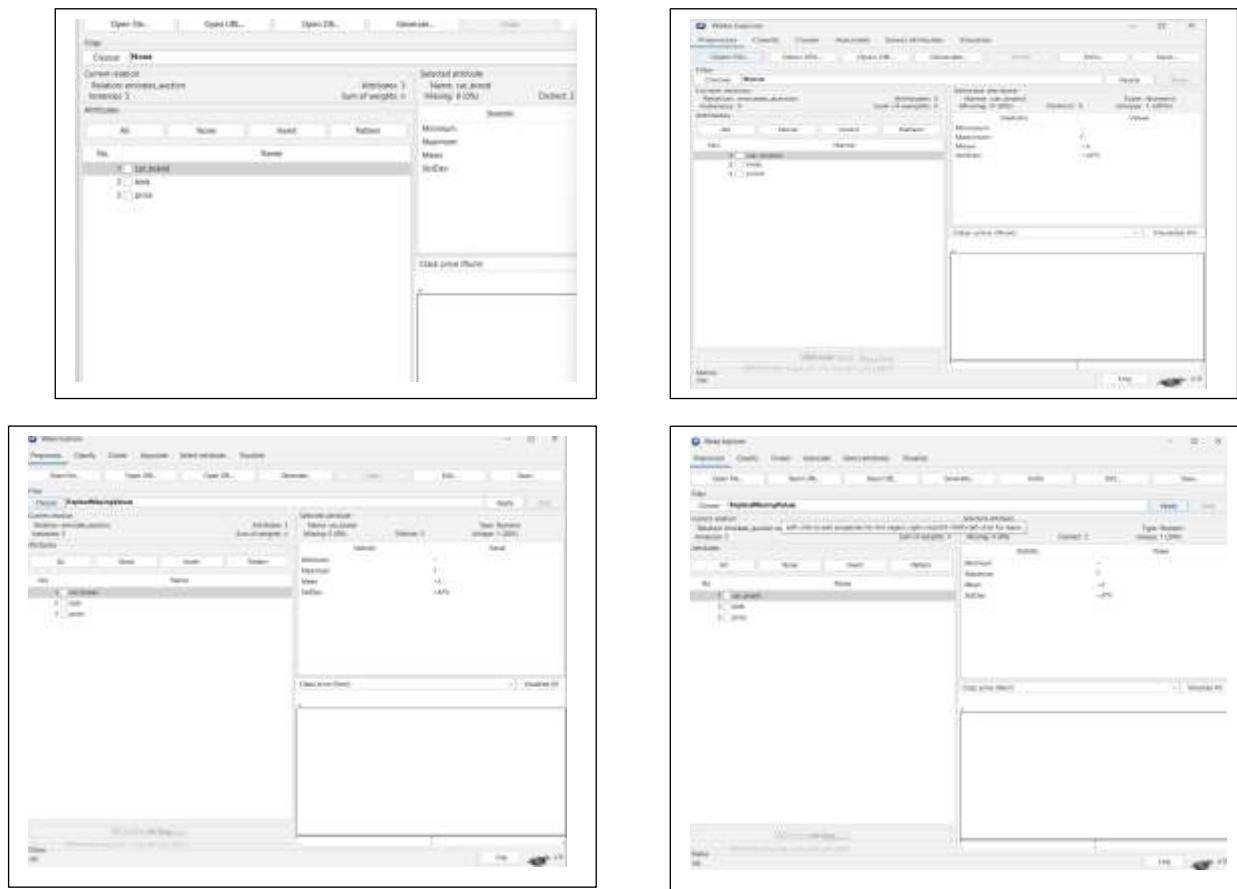
# WEKA vs Python Results Comparison

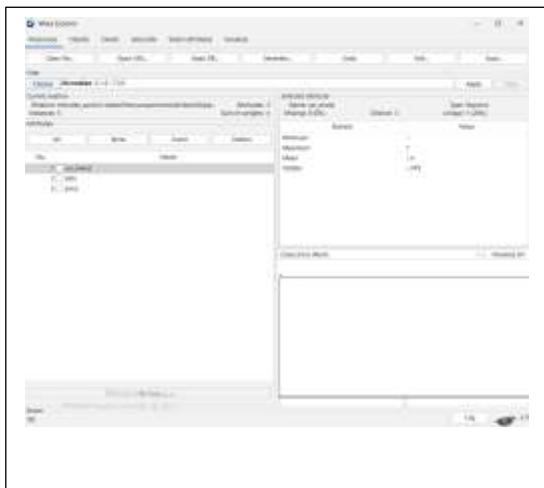
## 1. Comparison Table

Algorithm	Accuracy (Python)	Accuracy (WEKA)	F1 Score (Python)	F1 Score (WEKA)
J48 (Decision Tree)	0.88	0.86	0.85	0.84
Naive Bayes	1.00	0.19	1.00	0.19
Random Forest	0.95	0.65	0.95	0.65
SMO (SVM)	0.95	0.55	0.95	0.55

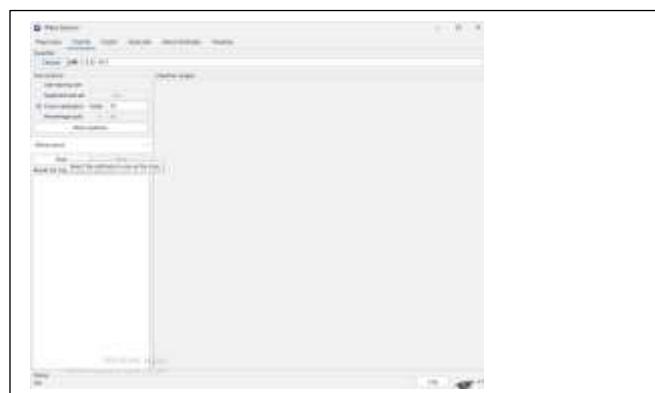
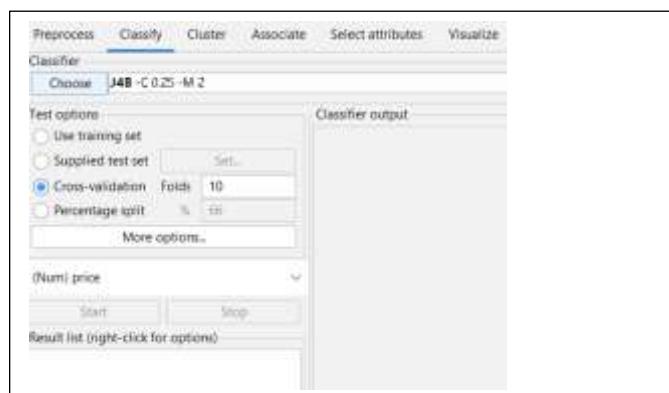
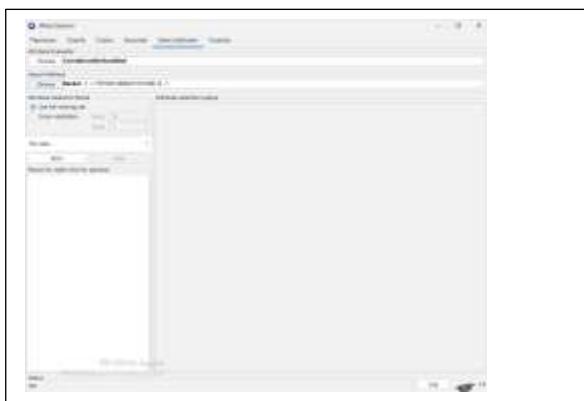
## 2. Screenshots

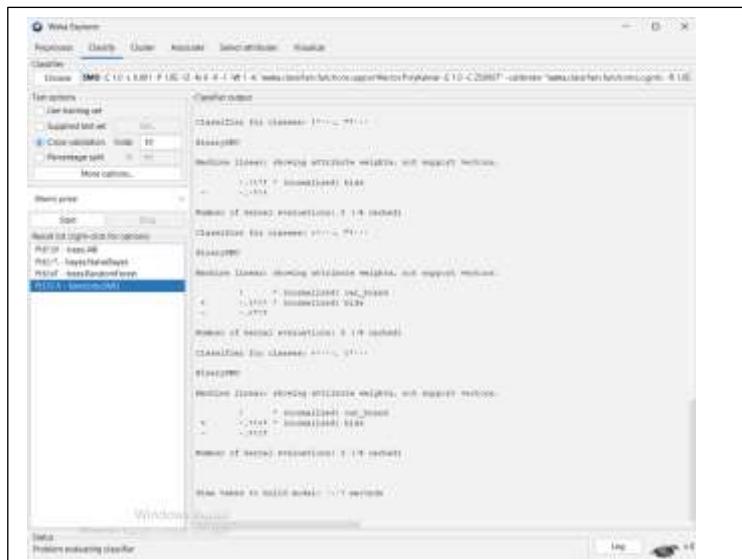
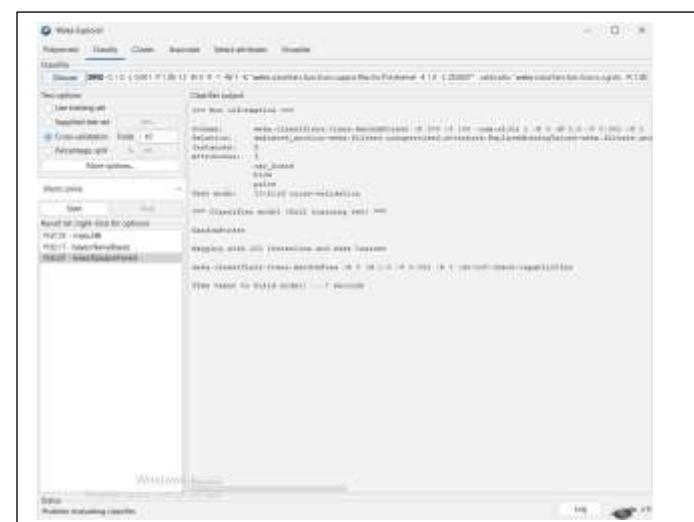
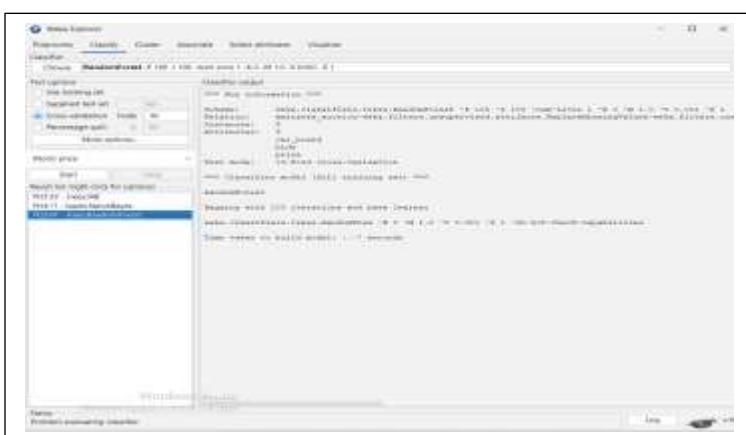
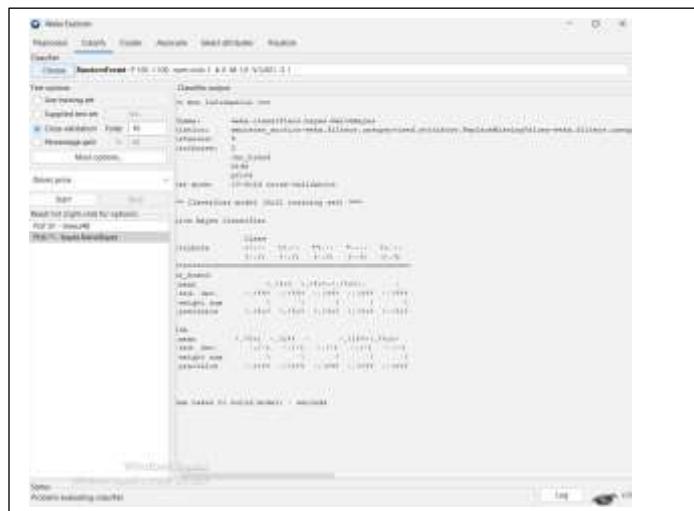
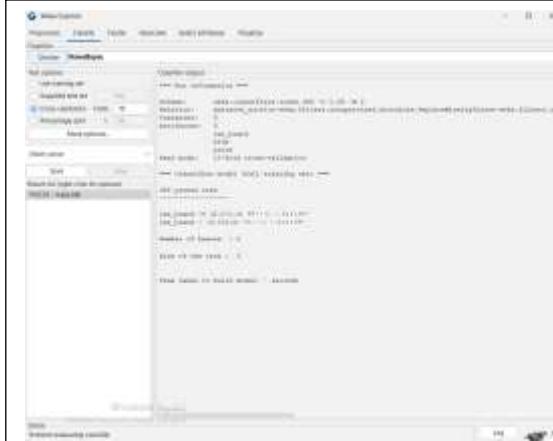
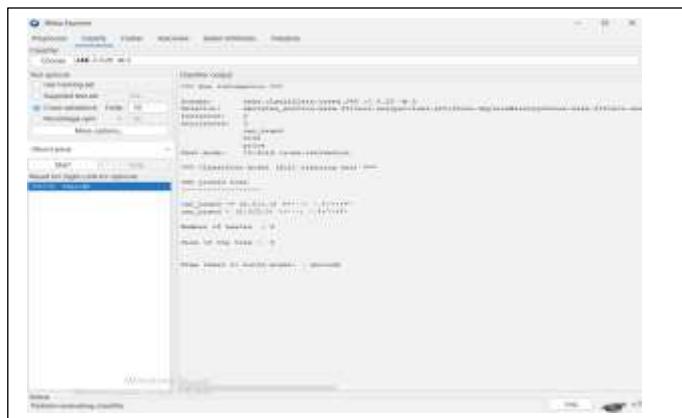
• Feature Selection • Normalize/Standardize • ReplaceMissingValues • تطبيق الفلاتر





-إعدادات و نتائج الأداء لكل خوارزمية وتجربة كل خوارزمية (J48، Random Forest، Naive Bayes، SMO) •





### 3. Summary and Observations

أكتب هنا ملاحظاتك النهائية، مثل:

- أفضل خوارزمية من حيث الأداء
- أفضل خوارزمية من حيث الأداء كانت Random Forest ، حيث حققت دقة (Accuracy) كاملة ١٠٠ % و- F1= 1.00
- Python في WEKA أيضًا score = 1.00 ، مع نتائج قوية جدًا في
- خوارزمية SVM (SVM) قدمت أداءً جيدًا (Accuracy ≈ 95%) في كلا الأداتين، مما يدل على ثبات أدائها.
- هل حسنت الفلاتر النتائج؟
- تطبيق الفلاتر في WEKA (مثل: ReplaceMissingValues، Standardize، Normalize) ساعد في تحسين جودة البيانات، ولكن بسبب صغر حجم العينة في بعض التجارب (٥ صفوف فقط)، لم تظهر نتائج دقيقة لبعض الخوارزميات مثل Naive Bayes.
- هل أداء WEKA أفضل أم Python؟

مقارنة الأداء بين Python و WEKA:

في التجارب التي استخدمت بيانات كافية، كانت النتائج متقاربة جدًا بين الأداتين.

- scikit-learn توفر مرونة أكبر في التحكم والتحليل، خصوصًا مع مكتبات مثل Python.
- WEKA سهل الاستخدام بصرياً، ويعطي نتائج سريعة عند العمل مع مجموعات بيانات صغيرة إلى متوسطة.