PROYECTO

Análisis de Notas de Estudiantes

Objetivo: Desarrollar un programa en Python que importe datos desde un archivo Excel, realice análisis estadísticos sobre las notas de los estudiantes y genere un informe detallado.

Datos de Entrada:

* Un archivo Excel (notas\_estudiantes.xlsx) con dos columnas:
  + "Nombre del estudiante" (texto)
  + "Nota" (número entero)

Requisitos del Programa:

1. Importación de Datos:
   * Utilizar biblioteca para leer el archivo Excel y almacenar los datos en un DataFrame.
2. Análisis Estadístico:
   * Calcular el número total de estudiantes.
   * Determinar el número y porcentaje de estudiantes aprobados (nota >= 70).
   * Determinar el número y porcentaje de estudiantes reprobados (nota < 70).
   * Determinar el número y porcentaje de estudiantes reprobados con notas entre 60 y 69.
   * Determinar la media de las notas.
   * Determinar la desviación estandar de las notas.
3. Generación de Informe:
   * Mostrar los resultados de manera clara y concisa en la consola.
   * El informe debe incluir:
     + Número total de estudiantes.
     + Número y porcentaje de estudiantes aprobados.
     + Número y porcentaje de estudiantes reprobados.
     + Número y porcentaje de estudiantes reprobados entre 60 y 69.
     + La media de las notas.
     + La desviación estandar de las notas.

Estructuras de Datos y Algoritmos:

* Utilizar DataFrames para almacenar y manipular los datos.
* Implementar funciones para realizar los cálculos estadísticos.

Requisitos Adicionales:

* El código debe estar bien documentado con comentarios explicativos.
* Se deben manejar posibles errores, como archivos no encontrados o datos inválidos.
* El código debe ser lo más eficiente posible.

Ejemplo de Salida:

Informe de Análisis de Notas

Número total de estudiantes: 25

Estudiantes aprobados: 15 (60.00%)

Estudiantes reprobados: 10 (40.00%)

Estudiantes reprobados (60-69): 3 (12.00%)

Media de las notas: 75.5

Desviación estandar de las notas: 12.34

Recomendaciones:

* Dividir el programa en funciones para mejorar la organización y la legibilidad.
* Utilizar formatos de cadena para presentar los resultados con la precisión adecuada.
* Agregar validaciones para que el código sea robusto.

Aquí tienes la rúbrica ajustada a un total de 40 puntos y su respectiva lista de cotejo.

**Rúbrica de Evaluación**

**Proyecto: Análisis de Notas de Estudiantes**

| **Criterio** | **Excelente (5 pts)** | **Bueno (4 pts)** | **Regular (3-2 pts)** | **Deficiente (1-0 pts)** | **Puntaje** |
| --- | --- | --- | --- | --- | --- |
| **Importación de Datos** | Se usa correctamente una biblioteca para leer el archivo Excel y los datos se almacenan sin errores. | Se usa una biblioteca adecuada, pero hay errores menores en la importación. | Se usa la biblioteca, pero hay fallos en la carga de datos. | No se logra importar correctamente el archivo. | /5 |
| **Análisis Estadístico** | Se calculan correctamente todos los valores requeridos (total, aprobados, reprobados, media, desviación estándar). | Se calculan la mayoría de los valores, pero con algunos errores. | Se omiten algunos valores importantes o hay errores graves en los cálculos. | No se logra realizar el análisis correctamente. | /5 |
| **Generación de Informe** | El informe presenta todos los resultados de manera clara y bien organizada. | El informe es comprensible, pero con errores de formato. | Presenta problemas de legibilidad o falta información. | No se genera un informe claro o completo. | /5 |
| **Estructura y Organización del Código** | Código bien organizado en funciones, fácil de entender. | Código funcional, pero con estructura mejorable. | Código poco modular y difícil de seguir. | Código desordenado o sin funciones claras. | /5 |
| **Documentación y Comentarios** | Código bien comentado en cada parte importante. | Código con comentarios, pero insuficientes. | Pocos comentarios o poco explicativos. | No hay comentarios en el código. | /5 |
| **Manejo de Errores** | Se manejan adecuadamente errores como archivos inexistentes o datos inválidos. | Se manejan algunos errores, pero no todos. | Se intenta manejar errores, pero con fallas. | No hay manejo de errores. | /5 |
| **Eficiencia del Código** | Código optimizado, sin cálculos innecesarios ni redundancias. | Código funcional, pero con algunas ineficiencias. | Código poco optimizado, con procesos redundantes. | Código ineficiente o lento. | /5 |
| **Presentación de Resultados** | Se presentan los datos con formato adecuado y precisión correcta. | Resultados claros, pero con detalles mejorables en la presentación. | Resultados confusos o con errores en los valores mostrados. | No se muestran resultados correctamente. | /5 |

**Puntaje Total: /40**

**Lista de Cotejo**

✔ **Importación de datos**  
✔ Se usa una biblioteca para leer archivos Excel.  
✔ Los datos se almacenan correctamente en un DataFrame.

✔ **Análisis estadístico**  
✔ Se calcula el número total de estudiantes.  
✔ Se determina el número y porcentaje de estudiantes aprobados.  
✔ Se determina el número y porcentaje de estudiantes reprobados.  
✔ Se calcula la media y la desviación estándar de las notas.

✔ **Generación de informe**  
✔ Se muestran los resultados de manera clara en la consola.  
✔ Se incluyen todos los valores requeridos.

✔ **Estructura y legibilidad del código**  
✔ El código está dividido en funciones.  
✔ Se utilizan nombres de variables claros.  
✔ Se incluyen comentarios explicativos.

✔ **Manejo de errores**  
✔ Se verifica la existencia del archivo antes de procesarlo.  
✔ Se validan los datos del archivo para evitar fallos.

✔ **Eficiencia y buenas prácticas**  
✔ Se evitan redundancias en el código.  
✔ Se optimizan los cálculos.  
✔ Se presentan los resultados con formato adecuado.