UNIVERIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA

EstudIA: herramienta web que analiza habitos personales para predecir el promedio academico

REPORTE DE PROYECTO

AUTORES:

JORGE JONATHAN DIAZ MARTINEZ

JHOVERAN CRISTIAN CUNO APAZA

PROFESOR: YURY OSCAR TELLO CANCHAPOMA

LIMA - PERU

8 DE DICIEMBRE DEL 2024

1. Introducción

1.1. Overview

La predicción de promedios académicos es crucial para los estudiantes universitarios, ya que les permite anticipar su rendimiento futuro y tomar decisiones informadas sobre su tiempo de estudio y otros aspectos relacionados con su rendimiento académico. Este proyecto tiene como objetivo desarrollar una plataforma web que utilice técnicas de aprendizaje automático, específicamente un modelo Random Forest, para predecir el promedio del próximo ciclo académico de los estudiantes universitarios. La plataforma toma en cuenta una variedad de parámetros, como el tiempo de estudio, el tiempo social, la edad, entre otros, y proporciona recomendaciones para optimizar el rendimiento académico.

1.2. Propósito

El propósito principal de este proyecto es ayudar a los estudiantes universitarios a predecir su promedio de ciclo, brindándoles información valiosa para mejorar su rendimiento. El modelo predice los promedios en función de datos previos, aunque el modelo se encuentra actualmente limitado por la base de datos, que contiene datos de estudiantes de colegio, lo que afecta la precisión de las predicciones para los estudiantes universitarios. No obstante, la plataforma ofrece un enfoque práctico y accesible, ya que permite a los usuarios interactuar con un formulario en la web, con el fin de obtener recomendaciones personalizadas.

2. Revisión Bibliográfica

2.1. Problema Existente

Tradicionalmente, los estudiantes universitarios no tienen acceso a herramientas fáciles de usar para predecir su rendimiento académico, lo que les dificulta planificar mejor su ciclo académico. Las predicciones de rendimiento suelen ser vagas y no personalizadas, basadas en estimaciones manuales y no en modelos automatizados. Este proyecto propone una solución mediante el uso de un modelo de aprendizaje automático, que proporciona predicciones personalizadas según diversos parámetros. Sin embargo, el modelo aún tiene ciertas limitaciones debido a que el dataset actual contiene datos de estudiantes de secundaria, lo que afecta la precisión de las predicciones en el contexto universitario.

2.2. Solución Propuesta

Este proyecto utiliza el algoritmo Random Forest para realizar predicciones sobre los promedios de los estudiantes universitarios, basándose en datos históricos de estudiantes de secundaria. El modelo está entrenado para identificar patrones en los datos y ofrecer recomendaciones personalizadas sobre el promedio esperado, aunque actualmente enfrenta desafíos debido a las diferencias en las bases de datos utilizadas. A pesar de esto, el modelo proporciona predicciones con una precisión significativa, lo que mejora la toma de decisiones para los estudiantes.

3. Análisis Teórico

3.1 Diagrama de Clases

El diagrama de clases ilustra la estructura del sistema y cómo las diferentes clases interactúan entre sí. A continuación se describen algunas de las clases más relevantes del proyecto:

1. **DataCollection**: Esta clase es responsable de la recolección de datos. Contiene métodos como validarData() para validar la información y sendToDataset() para enviar los datos procesados a otro componente del sistema.
2. **Preprocesamiento datos**: Se encarga de la limpieza y transformación de los datos. Los métodos principales de esta clase son cleanData() y transformData().

3. **Feature**: Define las características importantes para las predicciones. Cada característica tiene un nombre, un tipo y un valor de importancia. Los métodos calcularImportancia() y normalizar() permiten ajustar y validar los valores de las características.

4. **Random Forest**: Implementa el modelo de Random Forest utilizado para hacer las predicciones. Los métodos entrenarModelo() y predecir() permiten entrenar el modelo con los datos y luego realizar predicciones.

5. **SVM (Support Vector Machine)**: Otra técnica de aprendizaje automático que se utiliza en el sistema. Se configura con un kernel, C y gamma para entrenar el modelo y hacer predicciones mediante los métodos entrenarModelo() y predecir().

6. **KNN (K-Nearest Neighbors)**: Implementa el algoritmo KNN para predicciones basadas en los vecinos más cercanos. Esta clase también tiene métodos entrenarModelo() y predecir().

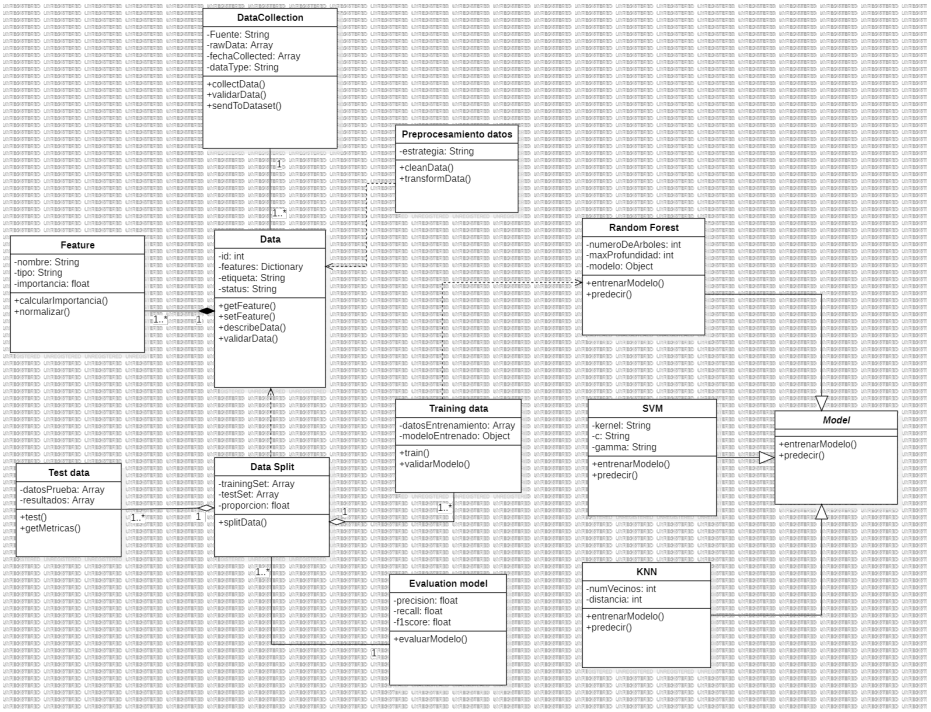
7. **Model**: La clase base para los modelos de predicción, que tiene métodos comunes para entrenar y predecir con cualquier tipo de modelo.

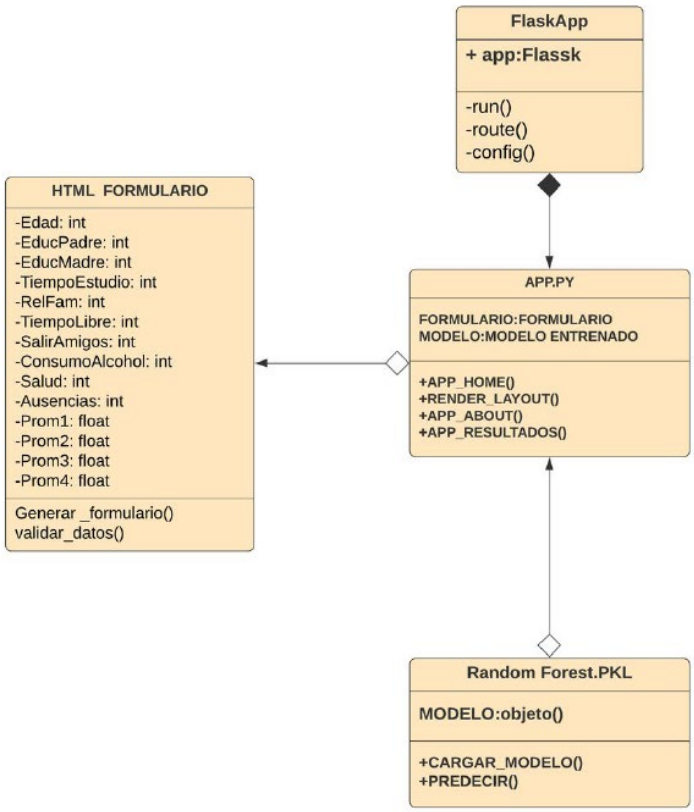
8. **Evaluation**: Se encarga de la evaluación del modelo utilizando métricas como la precisión (precision), el recall (recall) y la puntuación F1 (f1score). El método evaluarModelo() permite obtener un análisis de la calidad del modelo.

9. **Training data** y **Test data**: Se encargan de gestionar los conjuntos de datos de entrenamiento y prueba. Los métodos permiten dividir los datos, entrenar el modelo y obtener las métricas del modelo.

10. **Data Split**: Esta clase gestiona la partición de los datos en subconjuntos de entrenamiento y prueba, utilizando el método splitData().

Este diagrama ayuda a entender cómo cada componente del sistema está diseñado para trabajar en conjunto y cómo se organiza el flujo de información dentro de la aplicación.





1. HTML_FORMULARIO

- **Descripción**: Representa el formulario que los usuarios llenan en la interfaz web. Este formulario recopila datos como edad, nivel educativo de los padres, tiempo de estudio, salud, ausencias, notas pasadas, entre otros.
- **Atributos**: Cada dato que el usuario ingresa (edad, tiempo libre, etc.) está definido como un atributo de tipo entero (int) o flotante (float).
- **Métodos**:
    - Generar_formulario(): Se encarga de construir el formulario HTML visible en la web.
    - validar_datos(): Verifica que los datos ingresados sean correctos antes de enviarlos.

2. APP.PY

- **Descripción**: Es el núcleo del proyecto, que conecta el formulario HTML con el modelo de predicción entrenado.
- **Relación**:
    - Recibe los datos enviados desde el HTML_FORMULARIO.
    - Llama al modelo de predicción almacenado en el archivo Random_Forest.pkl.
- **Atributos**:
    - FORMULARIO: Una instancia del formulario HTML que facilita la recolección de datos.
    - MODELO: El modelo de predicción entrenado previamente.

- **Métodos**:
    - APP_HOME(): Renderiza la página principal donde se encuentra el formulario.
    - RENDER_LAYOUT(): Organiza cómo se muestran los elementos en la página web.
    - APP_ABOUT(): Muestra información sobre el proyecto.
    - APP_RESULTADOS(): Procesa los datos del formulario, llama al modelo y muestra las predicciones.

3. Random_Forest.PKL

- **Descripción**: Este archivo contiene el modelo de predicción (Random Forest) previamente entrenado. Está almacenado como un objeto para ser utilizado por la aplicación.
- **Atributos**:
    - MODELO: Representa el modelo de predicción cargado.
- **Métodos**:
    - CARGAR_MODELO(): Carga el modelo desde el archivo .pkl para ser usado por la aplicación.
    - PREDECIR(): Toma los datos ingresados por el usuario y genera una predicción del promedio futuro.

4. FlaskApp

- **Descripción**: Representa el framework Flask, que se encarga de ejecutar la aplicación web.
- **Atributos**:
    - app: Instancia de Flask utilizada para definir rutas y configurar la aplicación.
- **Métodos**:
    - run(): Inicia el servidor web.
    - route(): Define las rutas disponibles en la aplicación (e.g., /, /resultados).
    - config(): Configura las propiedades generales del servidor.

3.2. Hardware/Software Diseñado

Los requisitos de hardware para este proyecto son mínimos, ya que se centra principalmente en el desarrollo de software. Los requisitos de software incluyen:

- Lenguaje de programación Python.
- Librerías como pandas para manipulación de datos y scikit-learn para algoritmos de aprendizaje automático.
- El marco Flask para la creación de la aplicación web.
- HTML, CSS y JavaScript para la interfaz de usuario.

4. Investigación Experimental

Durante el desarrollo de la plataforma, se llevaron a cabo experimentos para garantizar su efectividad y precisión:

- **Preprocesamiento de Datos**: Limpieza de datos, manejo de valores faltantes y partición de los datos en conjuntos de entrenamiento y prueba.
- **Selección y Entrenamiento del Modelo**: Evaluación de diferentes algoritmos, eligiendo Random Forest debido a su robustez.
- **Evaluación del Modelo**: Uso de métricas como la precisión, la exactitud y la recuperación para evaluar el rendimiento del modelo.

5. Diagrama de Flujo

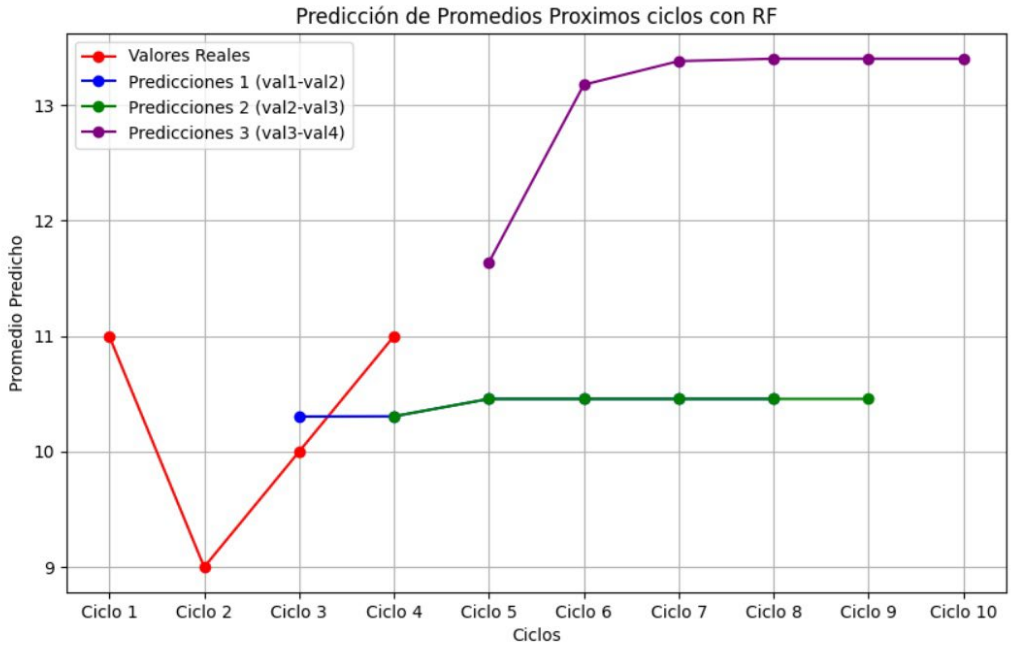
El diagrama de flujo describe el proceso de interacción con la plataforma: desde la entrada de datos por parte del usuario hasta la generación de las recomendaciones de promedio, mostrando de manera visual cómo se gestionan los datos y las predicciones.

6. Resultados

El modelo de predicción desarrollado ha mostrado un rendimiento decente, alcanzando una precisión significativa en las predicciones. Sin embargo, las predicciones no son del todo precisas debido a que el dataset contiene información de estudiantes de secundaria, y los promedios de los estudiantes universitarios son generalmente más altos. En promedio, los estudiantes universitarios tienen un rendimiento entre 8 y 13, lo que genera una ligera discrepancia en las predicciones del modelo.

6.1 Grafico de Prediccion

Una característica destacada del proyecto es la generación de gráficas para visualizar la evolución del rendimiento académico proyectado hasta el ciclo 10.



A continuación, se muestra un gráfico generado por el modelo de predicción, donde se visualizan los promedios predichos para los próximos ciclos.

Puntos rojos: Estos puntos representan los valores de los promedios de los últimos cuatro ciclos académicos, que el usuario ingresa como input en el formulario. Estos valores se utilizan para generar las predicciones para los ciclos futuros.

Puntos verdes, morados y azules: Estos puntos representan las predicciones generadas para los ciclos siguientes, utilizando los promedios de los ciclos anteriores. Cada color corresponde a una serie de predicciones basadas en los dos últimos promedios ingresados, con el fin de estimar cómo evolucionará el rendimiento del estudiante en los próximos ciclos.

- **Puntos verdes**: Predicción basada en los promedios del ciclo 1 y 2.
- **Puntos azules**: Predicción basada en los promedios del ciclo 2 y 3.
- **Puntos morados**: Predicción basada en los promedios del ciclo 3 y 4.

De esta manera, el modelo crea hasta tres series de predicciones, lo que ayuda a visualizar cómo podría cambiar el rendimiento académico del estudiante a lo largo de los siguientes ciclos.

7. Ventajas y Desventajas

Ventajas:

1. **Precisión en las predicciones**: La plataforma web de predicción utiliza un modelo de Random Forest, el cual es conocido por su alta precisión al manejar datos complejos y no lineales. Este modelo es ideal para predecir el rendimiento académico de los estudiantes en base a varios parámetros como el tiempo de estudio, las actividades extracurriculares y otros factores. El entrenamiento con un conjunto de datos histórico permite al modelo aprender patrones y realizar predicciones robustas. A pesar de las limitaciones del dataset, el modelo sigue demostrando una buena capacidad para predecir con un alto nivel de precisión.
2. **Interfaz amigable y accesibilidad**: La integración del modelo en una plataforma web facilita el acceso a los usuarios, incluso aquellos sin experiencia técnica avanzada. La interfaz de usuario es intuitiva, lo que permite a los estudiantes y académicos interactuar de manera sencilla con el sistema, proporcionando una herramienta valiosa para predecir promedios sin requerir habilidades técnicas complejas.
3. **Personalización y flexibilidad**: El modelo no solo predice el promedio final, sino que también ofrece la opción de personalizar los parámetros de entrada, como el tiempo de estudio y las actividades extracurriculares. Esto permite a cada usuario adaptar el sistema a su contexto personal, mejorando la utilidad de las predicciones.
4. **Optimización del rendimiento académico**: Al contar con recomendaciones sobre los posibles resultados y promedios, los estudiantes pueden ajustar sus hábitos de estudio y otras variables que afectan su desempeño académico. La plataforma puede ser una herramienta crucial para la mejora continua, brindando a los estudiantes una visión clara de sus puntos fuertes y áreas de mejora.

Desventajas:

1. **Dependencia de un dataset limitado**: Uno de los principales inconvenientes de la plataforma es que el modelo está entrenado con un conjunto de datos que incluye información principalmente de estudiantes de colegio. Debido a que el enfoque es trabajar con estudiantes universitarios, las predicciones pueden no ser del todo precisas. Las notas promedio de los

estudiantes universitarios son generalmente más altas que las de los estudiantes de colegio, lo que provoca que el modelo sobreestime las predicciones para los usuarios universitarios.

2. **Limitaciones en el rango de predicción**: Al estar basado en un modelo de clasificación como Random Forest, el sistema hace predicciones basadas en los parámetros de entrada proporcionados. Sin embargo, no tiene la capacidad de predecir con exactitud comportamientos atípicos o nuevos patrones de estudiantes que se desvíen de los datos históricos. Esto puede ser un inconveniente cuando se enfrenta a cambios significativos en los comportamientos de los estudiantes o en el currículo académico.

3. **Necesidad de actualizaciones periódicas**: Para mantener la precisión de las predicciones, el modelo requiere actualizaciones regulares del dataset, especialmente para incorporar datos de estudiantes universitarios y sus características. Este proceso de actualización puede ser costoso y llevar tiempo, lo que limita la flexibilidad del sistema en situaciones de rápido cambio, como ajustes en los sistemas educativos o cambios en los estilos de aprendizaje.

4. **Posible sobreajuste a patrones antiguos**: Aunque el Random Forest es menos propenso al sobreajuste debido a su naturaleza de combinación de múltiples árboles, el modelo aún puede verse afectado si los datos de entrenamiento no son lo suficientemente variados o si el contexto cambia significativamente. Esto es particularmente relevante en un entorno educativo donde los métodos de enseñanza y los enfoques de evaluación están en constante evolución.

8. Aplicaciones

La plataforma web de predicción de promedios tiene diversas aplicaciones tanto para estudiantes como para instituciones académicas. Algunas de las principales aplicaciones incluyen:

1. **Para estudiantes universitarios**: La plataforma es útil para los estudiantes que desean obtener una estimación precisa de su rendimiento académico. Al ingresar parámetros como el tiempo dedicado al estudio y las actividades extracurriculares, los estudiantes pueden obtener una predicción de su promedio final. Esta información les permite tomar decisiones informadas sobre sus hábitos de estudio y gestionar mejor su tiempo. Además, ofrece retroalimentación sobre áreas específicas en las que los estudiantes pueden mejorar, lo cual es fundamental para la autoevaluación y el desarrollo académico.

2. **Para instituciones educativas**: Las universidades pueden utilizar la plataforma como una herramienta complementaria para monitorear el rendimiento de sus estudiantes. Aunque el sistema está diseñado para estudiantes individuales, los administradores académicos pueden usar los datos recopilados para identificar patrones generales de rendimiento, como la relación entre el tiempo de estudio y el rendimiento. Esto puede ayudar a personalizar las estrategias de enseñanza y ofrecer apoyo adicional a los estudiantes con necesidades específicas. Además, la plataforma puede servir para predecir posibles cambios en el rendimiento general, lo que puede influir en las decisiones curriculares y de planificación educativa.

3. **Para programas de tutoría académica**: Los programas de tutoría pueden beneficiarse de las predicciones proporcionadas por el sistema. Al identificar a los estudiantes con un pronóstico de bajo rendimiento, los tutores pueden intervenir de manera temprana y proporcionar el apoyo necesario para mejorar los resultados. Esta capacidad de intervención temprana puede ser crucial para reducir la tasa de deserción y aumentar el éxito académico general.

4. **Investigación educativa**: La plataforma también puede ser útil para los investigadores en el campo de la educación. Al analizar cómo los diferentes factores, como el tiempo de estudio y las actividades extracurriculares, impactan en el rendimiento académico, los investigadores pueden obtener insights valiosos que ayuden a mejorar los métodos educativos y los enfoques

pedagógicos. Además, con la posibilidad de mejorar y actualizar los datos utilizados en el modelo, los investigadores pueden explorar nuevas formas de incorporar variables adicionales que influyan en el éxito académico.

5. **Desarrollo de políticas educativas**: A nivel gubernamental, las autoridades educativas pueden utilizar los insights obtenidos de esta plataforma para diseñar políticas más efectivas que aborden las necesidades de los estudiantes, optimicen los recursos educativos y mejoren el rendimiento académico a nivel macro. Los datos generados por la plataforma pueden proporcionar una base para la formulación de políticas orientadas a la mejora del sistema educativo.
6. **Plataforma para orientación profesional**: La herramienta también puede ser integrada en plataformas de orientación profesional, donde los estudiantes puedan usarla para obtener una idea de cómo sus hábitos y rendimiento académico pueden influir en su futura carrera profesional. Esto podría ayudar a los estudiantes a tomar decisiones informadas sobre su desarrollo académico y profesional.

9. Conclusión

El modelo de predicción ha demostrado ser una herramienta útil para la predicción del promedio académico, pero aún necesita mejoras, especialmente en lo que respecta a la precisión de las predicciones para estudiantes universitarios. El proyecto demuestra el valor del aprendizaje automático para personalizar la experiencia educativa y permite que los estudiantes tomen decisiones informadas sobre su rendimiento. En el futuro, se planea expandir el dataset y mejorar la precisión del modelo mediante técnicas más avanzadas.

10. Alcance Futuro

Se propone:

- **Ampliar el dataset** para incluir más datos relevantes de estudiantes universitarios.
- **Integrar técnicas avanzadas** de aprendizaje automático como redes neuronales para mejorar la precisión.
- **Desarrollar aplicaciones móviles** para mejorar el acceso a la plataforma.
- **Integrar dispositivos IoT** para monitoreo en tiempo real de datos educativos y comportamentales.