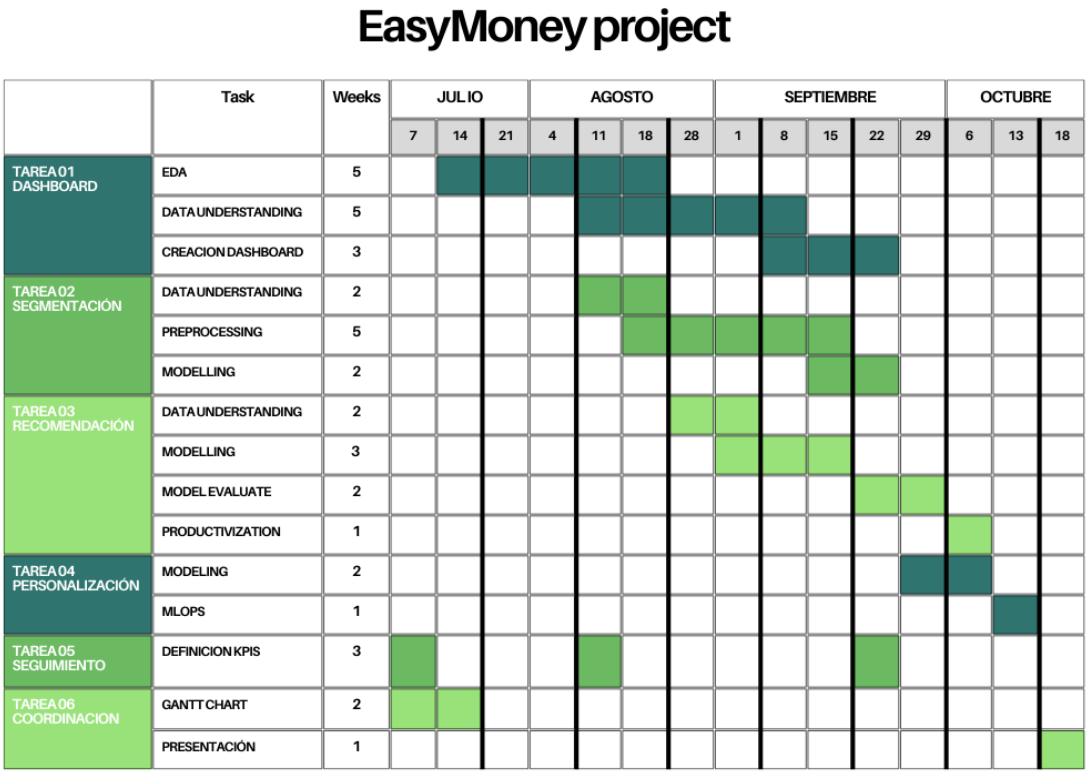
Tarea 06 Coordinación de proyecto EASYMONEY

Introducción

El objetivo principal de esta tarea es establecer un proceso robusto de coordinación y validación de los cambios y mejoras que se implementen tanto en los modelos operativos como en el dashboard. Es crucial que exista un proceso de validación para los informes y análisis, así como un plan de proyecto claro sobre la gestión y coordinación de las tareas asignadas. Esto garantiza la calidad y la alineación de los resultados con las expectativas de las partes interesadas.

Diagrama de Gantt

El diagrama de Gantt presentado ilustra el cronograma de las tareas del proyecto desde julio hasta octubre. A continuación, se describen las tareas y sus respectivas fechas de inicio y duración:



Descripción de las Tareas

TAREA 01: DASHBOARD

- **Objetivo:** Realizar un análisis exploratorio de los datos (EDA) y desarrollar un dashboard para la visualización de todos los departamentos.
- **Duración:** 5 semanas.
- **Actividades Clave:** EDA y creación del dashboard.

TAREA 02: SEGMENTACIÓN

- **Objetivo:** Realizar un entendimiento de los datos y aplicar técnicas de segmentación, para generar una segmentación general de los clientes.
- **Duración:** 5 semanas.
- **Actividades Clave:** Preprocesamiento y modelado.

TAREA 03: RECOMENDACIÓN

- **Objetivo:** Desarrollar un modelo recomendación en este caso de nuestro producto credit card para una campaña de 10.000 clientes por email.
- **Duración:** 3 semanas.
- **Actividades Clave:** Entendimiento de datos y modelado.

TAREA 04: PERSONALIZACIÓN

- **Objetivo:** Segmentar esos 10.000 clientes para generar una personalización de las creatividades de la campaña por mail.
- **Duración:** 3 semanas.
- **Actividades Clave:** Segmentacion y MLOps.

TAREA 05: SEGUIMIENTO

- **Objetivo:** Definir KPIs y realizar un seguimiento de los modelos implementados.
- **Duración:** 3 semanas.
- **Actividades Clave:** Definición de KPIs.

TAREA 06: COORDINACIÓN

- **Objetivo:** Asegurar la correcta comunicación y coordinación entre los miembros del equipo.
- **Duración:** 2 semanas.
- **Actividades Clave:** Gantt chart y presentación final (22 de Octubre) .

Gestión del Código con Git

La gestión del código es un aspecto esencial para el éxito del proyecto. Para ello, se utilizará Git, específicamente GitHub, como plataforma de control de versiones.

Ventajas de Git y GitHub

- **Control de Versiones:** Permite a los desarrolladores mantener un registro detallado de los cambios realizados en el código, facilitando el seguimiento y la recuperación de versiones anteriores si es necesario.
- **Colaboración Efectiva:** Múltiples desarrolladores pueden trabajar en diferentes características simultáneamente sin interferir en el trabajo de los demás gracias al uso de ramas.
- **Proceso de Aprobación:** A través de pull requests, los cambios pueden ser revisados y aprobados por un manager antes de ser fusionados a la rama principal, asegurando la estabilidad del código.

Flujo de Trabajo

- **Rama Principal:** La rama master contendrá el código que se lleva a producción.
- **Rama de Desarrollo:** La rama develop se utilizará para introducir cambios nuevos y características en desarrollo.
- **Revisiones:** Antes de realizar un merge a master, se debe realizar una revisión exhaustiva del código, y este debe ser aprobado por el manager del proyecto.

Documentación del Código

La documentación es vital para garantizar que el código sea comprensible y fácil de mantener. Deberá abarcar tanto una perspectiva técnica como una no técnica.

Recomendaciones de Documentación

- **Descripción Funcional:** Cada función debe tener una breve descripción de su propósito, los parámetros de entrada y salida, y la última revisión realizada.
- **Código Limpio:** Es importante seguir principios de **Clean Code**, asegurando que el código sea fácil de entender, mantener y escalar. Las funciones deben ser cortas y realizar una única acción.
- **Documentación Técnica y General:** Se debe crear un manual técnico para desarrolladores y otro más accesible para el personal no técnico. Esto facilita la incorporación de nuevos miembros al equipo y asegura que otros departamentos comprendan el funcionamiento del modelo.

Creación de un Paquete

Si el proyecto crece en complejidad, se recomienda construir un paquete propio en Python o R que contenga todas las funciones y estructuras utilizadas, lo que facilitará su reutilización y gestión.

Metodología de Trabajo

Se optará por la metodología **Agile**, específicamente por **Scrum**, que ha demostrado ser efectiva en proyectos de desarrollo de software.

Beneficios de Scrum

- **Iteraciones Rápidas:** Los proyectos se dividen en sprints de dos semanas, permitiendo ajustes y mejoras constantes.
- **Involucramiento del Cliente:** Los clientes y partes interesadas tienen la oportunidad de dar feedback continuo, lo que mejora la alineación del producto final con sus expectativas.

Estructura de Scrum

1. **Backlog:** Todas las tareas se recopilan en un backlog, que se prioriza y se detalla con historias que describen los objetivos de cada tarea.
2. **Planificación de Sprints:** Al inicio de cada sprint, el equipo selecciona las tareas del backlog que se abordarán, asignándolas a los desarrolladores
3. **Scrum Board:** Se utilizará una scrum board para visualizar el progreso de las tareas, con columnas para To Do, Doing, Review, y Done.
4. **Reuniones Diarias:** Se llevarán a cabo reuniones diarias de Scrum para revisar el avance y los bloqueos

Conclusiones

La planificación y ejecución de este proyecto de análisis de datos y desarrollo de modelos está diseñada para maximizar la colaboración y la adaptabilidad mediante el uso de metodología Agile y herramientas de gestión de código. El uso de un diagrama de Gantt permite tener una visión clara del progreso del proyecto y asegura que todos los miembros del equipo estén alineados con los objetivos y plazos establecidos.