

Voorspelling, aplicación de escritorio para aprendizaje de máquina supervisado

Autores:

Iván-David Rey-Rueda

German-Francisco Diaz-Figueredo

Ingeniería de software
Universidad de Santander UDES
Bucaramanga, Santander
2020

Voorspelling, aplicación de escritorio para aprendizaje de máquina supervisado

Autores:

Iván-David Rey-Rueda
German-Francisco Diaz-Figueredo

Trabajo de grado para optar por título de Ingeniero de Software

Directora proyecto:

Paola Rondón-Villarreal

Ingeniería de software
Universidad de Santander UDES
Bucaramanga, Santander
2020

Índice general

1. Introducción	4
2. Objetivo	5
2.1. Objetivo general	5
2.2. Objetivo específico	5
3. Problema	6
3.1. Descripción problema	6
3.2. Justificación	6
3.3. Planteamiento problema	6
4. Marco teórico	7
5. Metodología	8
6. Conclusión	9

Abstract

Keywords: Supervised Machine Learning, Desktop app, Scikitlearn.

Resumen

Palabras clave: Aprendizaje de Máquina Supervisado, Aplicación de escritorio, Scikitlearn.

Capítulo 1

Introducción

Capítulo 2

Objetivo

2.1. Objetivo general

Desarrollar una aplicación nativa de escritorio para aprendizaje de máquina supervisado en el sistema operativo *Windows 10*, que permita a los usuarios entrenar y realizar predicciones con base a un conjunto de datos inicial suministrado en formato *CSV* o *TSV*.

2.2. Objetivo específico

- Diseñar el prototipo de la interfaz gráfica utilizando *PhotoShop* y la herramienta web Figma, respetando los estándares ISO 13406-2, 11581-10 y 9241-210.
- Implementar el diseño de la interfaz gráfica en *Windows Forms* en el *IDE Visual Studio 2019*, a partir del diseño construido en la herramienta web de prototipado *Figma*.
- Realizar el diseño de alto nivel del back-end de la aplicación, utilizando diagramas *UML* con base a herramientas o extensiones como: *PLantUML*, *LucidChart* y el diseñador de clases de *Visual Studio 2019*.
- Desarrollar el *back-end* de la aplicación con base al diseño previamente establecido y las librerías disponibles de *ScikitLearn* del lenguaje *Python 3.8.2*.
- Unificar el *front-end* implementado en el *Framework Windows Forms* con el *back-end* de *Python 3.82* elaborado en el *IDE Visual Studio 2019* y *Visual Studio Code*.
- Desplegar la aplicación posteriormente tras haber completado sus fases de requerimientos, diseño, codificación y pruebas, que permita seleccionar un conjunto de datos, ajustar hiperparámetros, entrenar clasificadores y finalmente regresar los resultados de las predicciones.

Capítulo 3

Problema

3.1. Descripción problema

Una parte de los analistas de datos carecen de los conocimientos de programación en *Python* para generar sus propios algoritmos, y de igual forma entusiastas y estudiantes que desean desarrollar sus propios modelos de aprendizaje de máquina no precisan de las habilidades necesarias para escribir su propio código, de modo que se ven en la necesidad de recurrir a software o aplicaciones que les permita realizar este tipo de tareas. La mayoría de aplicaciones con este fin son de pago o en algunos casos con pruebas gratuitas muy limitadas, por lo que después de unos pocos usos se hace necesaria una suscripción, a causa de que es necesario como mínimo generar los ingresos necesarios para mantener en servicio el aplicativo web para el uso del público.

3.2. Justificación

Las aplicaciones para aprendizaje de máquina supervisado disponibles en la web presentan varias desventajas, como lo son: (1) la integridad de los datos al utilizar aplicaciones web no se garantiza, ya que hay intercambio de información entre cliente-servidor. Esta situación se presenta en el caso de los *datasets*, los cuales pueden contener datos sensibles como información personal. (2) Los tiempos de ejecución, debido a que los algoritmos de búsqueda de hiper-parámetros exhaustivos tienen un tiempo de ejecución del orden de $O(n^k)$, es decir tiempo polinómico, y por lo general utilizar algoritmos alternativos no se encuentra en los intereses de quienes ofrecen servicios web para aprendizaje de máquina supervisado, por lo que los usuarios se ven obligados a utilizar los métodos regularmente implementados en aplicaciones de este tipo. (3) Los altos costes de ejecutar entrenamiento y pruebas de aprendizaje de máquina supervisado en ambientes web, ya sea utilizando arquitecturas *serverless* o basadas en micro-servicios, se requiere obligatoriamente del uso de recursos físicos en los servidores donde se aloja la aplicación, lo que supone costos monetarios por mantener disponible el servicio, más el propio uso de la aplicación por parte de sus clientes.

3.3. Planteamiento problema

Capítulo 4

Marco teórico

Capítulo 5

Metodología

Capítulo 6

Conclusión