75.06/95.58 Organización de Datos - 2C 2019

Trabajo Práctico 2

Competencia de Machine Learning

Grupo 34: "DataTravellers"

Integrantes:

- Andrés Pablo Silvestri: 85881 (silvestri.andres@gmail.com)
- Juan Manuel González: 79979 (juanmg0511@gmail.com)
- Patricio Pizzini: 97524 (pizzinipatricio@yahoo.com.ar)

Link a repositorio de GitHub:
https://github.com/patopizzini/Organizacion_Datos_2C2019/tree/master/TP2

Fecha de entrega: 05/12/2019

Contenido

Introducción

En este informe se describe el trabajo realizado para poder llegar a los distintos resultados que se fueron entregando a la competencia de Kaggle: Inmuebles24.

La competencia consistía en estimar cúal es el valor de mercado de las propiedades.

A partir de los datos proporcionados, se fueron extrayendo distintos features y probando distintos algoritmos de machine learning para obtener las mejores predicciones posibles, el proceso para obtener este resultado se describe a continuación.

Durante el trabajo, se utilizo el lenguaje de programación Python y algunas librerías extras como sklearn y pandas para facilitar el manejo de datos.

Transformación de datos y armado de features

Primeramente se trabajo sobre que decision tomabamos sobre los valores que estaban nulos, una vez hecho eso, se pasó a tipo 'category' todas las columnas que resultan acordes para ser categorizadas por tener una cantidad limitada de valores, lo que permite mejorar el rendimiento de las diferentes operaciones que involucren estas columnas.

Se describirán a continuación el armado de distintos features que se fueron probando, si bien no todos quedaron como definitivos para obtener el resultado final, fueron parte del proceso, coimo así también el formatear cierta información de modo que nos quede todo numérico y poder utilizar algoritmos de tipo árbol que son los que elegimos para plantear la solución de este problema.

Features utilizados

Luego de muchas pruebas, se fueron descartando/sumando features al modelo, con lo cual vamos a dividir en dos secciones esta parte, por un lado los features que finalmente quedaron en la solución final brindada y por otro lado los features que fueron descartados por no brindarnos valor a la hora del análisis, vale aclarar que estos siendo bastantes y quedando algunos en el camino, solo vamos a mencionar aquellos que a priori nos parecían más relevantes y que para nuestra sorpresa no nos dieron un valor significativo o positivo.

Algoritmos utilizados

Random Forest

Como primer medida se decidió iniciar el trabajo para la predicción usando Random Forest, entre otros motivos podemos decir que la elección está basada en el simple hecho de que cuando un atributo es un buen predictor sus árboles van a tener mejores resultados que aquellos que usan un conjunto de atributos que no son buenos predictores, como así también sabemos que son invariantes a la escala de los atributos es decir que no necesitamos normalizarlos, lo cual entre otras cosas nos facilita la operatoria, con todo esto partimos con la idea inicial de utilizarlo para ir descubriendo qué features pueden resultar positivos y cuáles no.

KNN

XGBoost

Tomando como base los features que ya teníamos evaluados con Random Forest decidimos aplicar XGBoost como la primer alternativa a sabiendas que ya no podíamos encontrar muchas mejoras sobre lo que se obtenía con Random Forest, es por eso que aunque sabiendo que con este otro algoritmo no siempre es el mejor al menos casi siempre da buenos resultados y muchas competencias se ganan usándolo como base.

En un primer momento no le pusimos mayor importancia a los hiper parámetros (no los trabajamos más que tomando algunos a base de pruebas) y probamos simplemente la ejecución del entrenamiento y la predicción.

A partir de ese momento comenzamos a trabajar tanto con los hiper parámetros que se verá más en detalle más adelante como así también buscando otros features que nos pudiesen traer una mejora sustancial.

Ensamble de algoritmos

Luego tener los algoritmos empleados por separado y a sabiendas que de momento teníamos a Random Forest como la alternativa más precisa decidimos emplear una combinatoria de los algoritmos para ver si esto nos generaba una mejora y es en base a estos features hicimos entrenamientos. Esto no nos generó una mejora, por lo que decidimos no incluir estas alternativas puesto que todas terminaban empeorando lo que conseguimos.

Hiperparametros

Conclusiones

Durante el desarrollo de este trabajo, se utilizaron distintas herramientas de machine learning adquiridas durante la cursada, probando diferente algoritmos como así también la construcción de distintos features para ir mejorando el score de predicción.

Al probar distintos algoritmos de regresión, verificamos que en este problema de ML, Random Forest nos dio un mejor score en relación a XGBoost, lo que nos llamo la atencion ya que este ultimo se utiliza más para competencias de ML.

Finalmente, notamos que si bien se obtenían mejoras al tunear los hiperparametros, las mejoras más significativas ocurrieron al agregar features que aportaron valor al modelo.