Facultad de Estudios Superiores

Acatlán

ANÁLISIS ESTADÍSTICO EN STARBUCKS

TRABAJO FINAL
ANALISIS MULTIVARIADO

Equipo de trabajo:
Ávila Cruz Hector Manuel
Bolaños Macía Santiago
Pineda Rodríguez Lorena
Tapia Peñaloza Jorge Luis
Vazquez Cortes Carolina

Mayo del 2018

Profesor: JOSÉ GUSTAVO FUENTES CABRERA

Índice

1. Reporte Ejecutivo **1**

2. Introducción **3**

3. Objetivos y alcances **4**
3.1. Objetivo General 4
3.2. Objetivos Específicos 4

4. Estado del Arte **5**
4.1. Marco teórico 11
4.1.1. Machine Learning 11
4.1.2. Aprendizaje no supervisado 12
4.1.3. Aprendizaje supervisado 15

5. El conjunto de datos y sus variables **20**

6. Análisis descriptivo de las variables **21**
6.1. Análisis Univariante 21
6.1.1. Variables discretas: 21
6.1.2. Variables continuas: 22
6.2. Análisis Multivariado 23
6.3. Valores Extremos 25
6.4. Valores Categoricos 26

7. Modelación no supervisada **28**
7.1. Análisis de Componentes Principales 30
7.2. Escalamiento Multidimensional 31
7.3. Análisis Estadístico de los Grupos 32

8. Modelación Supervisada **36**
8.1. Modelo Predictivo (Comida) 36

9. Estrategia de uso **37**

10. Conclusiones **38**

11. Referencias **39**

Índice de figuras

1. Frecuencia en los datos de las Variables Discretas 21
2. Histograma de todos los datos continuos 22
3. Correlacion. 23
4. Horas. 24
5. Días. 24
6. Productos. 25
7. Gráfica de Inercia 28
8. Visualización de Cluster por PCA 30
9. Visualización de Cluster por MDS 31
10. Porcentaje total de clientes por Grupo 32
11. Preferencia de productos por cluster 33
12. Gasto promedio por cluster 34
13. Porcentaje de llegada de los grupos por hora del día 34
14. El izquierdo la Real y El lado derecho es el Prediccion. 37

1. Reporte Ejecutivo

Es común entre los establecimientos comerciales y demás empresas dedicadas a prestar servicios experimentar cambios y transiciones tanto en sus clientes,como en el tipo de artículos y servicios que hay que ofrecerle a éstos, pues la demanda varía y hay que tomar en cuenta aspectos de cantidad de personal, localización del establecimiento y horarios en los que se presta el servicio;ante los ya mencionados cambios, las empresas deben tomar medidas y actuar para contrarrestar las posibles consecuencias que traigan consigo los cambios como puede ser la acumulación de productos sin vender o la falta de éstos para abastecer la demanda, perder interés por parte del público y disminuir por ende las ventas; además de que es importante recalcar que el fin último de toda compañía es brindar un servicio de alta calidad a sus clientes.

Cuando un comprador entra en una institución de este tipo, deja ver ciertas pautas en su comportamiento, dichas pautas pueden ser influenciadas por una gran cantidad de factores, si estos factores son analizados correctamente,podríamos convertir los datos en información relevante para la toma de decisiones e incluso poder predecir comportamientos y situaciones del negocio.

Mediante el análisis multivariado en los agentes que influyen en las compras de la compañía Starbucks Coffee y haciendo uso de algunas herramientas que nos provee la rama de la inteligencia artificial como lo es el machine learning, hemos clasificado a los clientes en diversos grupos, el comportamiento y características de los 4 tipos de clientes que fueron encontrados, llevó a los involucrados en este proyecto a desarrollar más aún, de acuerdo a la información recabada anteriormente y las cualidades del grupo será posible estimar también la bebida o bocadillo que el cliente va a pedir.

Los beneficios son bastos, permite al negocio prevenir los recursos que necesitará a lo largo del día, además de que le dará al empresario una idea de cuáles son los productos que más éxito están teniendo entre los comensales, y permitiría establecer estrategias para invitar al consumo a los grupos que menos afluencia han tenido, o por el contrario premiar a aquellos grupos cuya asistencia es frecuente.

El mercado en el cuál sería posible ver los resultados de la aplicación de éstas técnicas avanzadas, actuales y de alta presición es abundante, podría ser empleada en cualquier empresa o negocio que quiera organizar de manera adecuada los descansos de sus trabajadores, maximizar su productividad, conocer en mayor medida a sus compradores y que estos elementos le permitan un crecimiento, tanto en ganancias como en reconocimiento.

El propósito de los desarrolladores de este proyecto, estudiantes de la carrera de actuaría,fue aplicar los conocimientos de la materia de análisis multivariado, la creación de clusters y la desarrollo de modelos predictivos, para poder comunicar datos de interés, no detectables a simple vista, de manera que el trabajador y el ejecutivo de la empresa centro de esta investigación pueda ver los beneficios de la innovación y los avances del uso de las herramientas de la inteligencia artificial para conocer a sus clientes y mejorar su situación financiera.

2. Introducción

Starbucks Coffee Company es una cadena internacional de café fundada en 1971,esta empresa ofrece al público una gran variedad de productos como café elaborado, bebidas calientes, bocadillos y algunos otros productos tales como tazas, termos y café en grano,además de poner a la venta productos como libros, CD's y películas.

Al inicio de los 70's cuando recién iniciaba el negocio,los fundadores jamás se hubieran imaginado el éxito que hoy ha traído como consecuencia más de 17000 instalaciones en 50 países, entre ellos México que cuenta con cerca de 600 sucursales de la franquicia de cafeterías.

Una de éstas se encuentra en la plaza La Cúspide Sky Mall y es en la que se concentrará el análisis del presente trabajo.

La Cúspide es una plaza que se encuentra entre la primera y la cuarta sección de Lomas Verdes en el municipio de Naucalpan de Juarez del Estado De México, la plaza se considera como un lugar al que la gente acude para llevar a cabo diversas actividades como el pago de impuestos y demás servicios por lo que el establecimiento de Starbucks que lleva el mismo nombre ha modificado su perfil de visitantes.

Debido a estos cambios se presenta la necesidad de hacer ajustes en el negocio, pues las características y necesidades de nuestros clientes se han transformado; como consecuencia de esto se realiza un análisis de los factores que inciden en la eficiencia de la sucursal; tales como el registro de las transacciones, la hora en la que éstas son realizadas, el tráfico interno de clientes y sus atributos y los productos que se consumen para encontrar relaciones que nos permitan clasificar en grupos a los consumidores, más aún se propondrá un modelo que permita predecir lo que consumen los clientes que llegan a la filial para conocer los recursos de los cuales debe disponerse para cubrir la demanda,mejorar el servicio y maximizar las ganancias, esto a través de implementar estrategias en los grupos principales que acuden al establecimiento .

Primero, se llevará a cabo una introducción y descripción de nuestras variables, mostraremos el sustento teórico de esta investigación, se expondrán las relaciones entre las variables y su dependencia, en relación a estas dependencias se crearán Clusters y se intentarán predecir las ventas llevado a cabo el análisis de las variables y grupos, para así poder ofrecer posibles soluciones a la situación presentada en Starbucks La Cúspide.

3. Objetivos y alcances

3.1. Objetivo General

Hacer uso de la modelación supervisada y no supervisada sobre un conjunto de datos obtenidos de las compras realizadas por los clientes de Starbucks "La Cúspide" para la clasificación e identificación de patrones entre los compradores que le permitan a los trabajadores de Starbucks hacer estimaciones sobre los productos a consumir por dichos grupos para su aplicación en estrategias de negocio.

3.2. Objetivos Específicos

- Valerse de las herramientas que brinda el machine learning para descubrir las conexiones que hay entre las variables de la información recabada en los registros de venta .

- Utilizar el análisis de los tipos de clientes que acuden a Starbucks " La Cúspide" para así implementar un modelo en el que se estime la cantidad de clientes que llegarán a la tienda, los artículos que son propensos a consumir y dependiendo el grupo al que pertenecen enfocar estrategias de venta para dichos grupos y concluir que dichas estrategias puedan maximizar las ganancias del negocio.

4. Estado del Arte

El análisis del estado del arte que aquí se agrupan Modelación supervisada y No supervisada artículos que se han realizado considerando la agrupación de patrones.

Investigaciones-Internacionales

Análisis de curvas ROC en estudios epidemiológico de psicopatologia infantil: aplicación al cuestionario CBCL

OBJETIVO:
Mediante el análisis de curvas ROC [1] se estudia la precisión diagnóstica del Child Behavior Checklist (CBCL) y se obtiene elpunto de corte Óptimo en una muestra de 196 niños y adolescentes procedentes de centros de consulta pediátrica y psiquiátrica. Se utilizaron como patrones de referencia el grupo de procedencia, el diagnóstico según la entrevista diagnóstica estructurada DICA-R y el diagnóstico del clínico. Los resultados indican que la capacidad del CBCL para discriminar entre sujetos con y sin psicopatología depende en gran medida del patrón de referencia utilizado, siendo mejor el rendimiento de la prueba cuando se contrasta con el grupo de procedencia o con la entrevista diagnóstica estructurada. Como prueba de cribado, el punto de corte de la puntuación total del CBCL se situada entre 10s valores 50 y 54 para optimizar la sensibilidad.
MUESTRA:
Los estudios epidemiológicos en psicopatologia infantil utilizan frecuentemente pruebas de cribado (screening) que proporcionan un diagnóstico binario del estado del sujeto: sano o patológico, aplicando un punto de corte al resultado cuantitativo que ofrece el algoritmo de corrección de la prueba. El análisis de curvas ROC (Receiver Operating Characteristics) proporciona una medida global de precisión diagnóstica, independiente del punto de corte y de la prevalencia, de gran interés para muchos estudios epidemiológicos en psicopatologia infantil, ya que permite comparar la eficacia de diferentes pruebas diagnósticas aplicadas a una misma muestra o de una misma prueba aplicada a diferentes muestras (Fombonne, 1991). La curva ROC se obtiene representando la sensibilidad (porcentaje de verdaderos positivos), en el eje de ordenadas, y la inespecificidad (porcentaje de falsos positivos), en el eje de abscisas, para diferentes puntos de corte aplicados sobre el resultado cuantitativo de un test. Así pues, la curva ROC es independiente del punto de corte y de la prevalencia, cumpliendo así las dos condiciones enunciada por Swets (1988) como imprescindibles en un indicador de precisión diagnóstica.

RESULTADO:
La curva ROC refleja el grado de solapamiento de las puntuaciones del test en 10s

[1] Navarro (1998)

grupos de sujetos sanos y patológicos (véase Figura 1). Cuando el solapamiento es total (test inútil) la curva ROC recorre la diagonal positiva del gráfico, ya que para cualquier punto de corte sensibilidad (S) es igual a inespecificidad (NE). Cuando el solapamiento es nu10 (test perfecto), la curva ROC recorre 10s bordes izquierdo y superior del gráfico, ya que para cualquier punto o bien S=l o bien NE=O, existiendo algún punto de corte en que ambos S=l y NE=O. En la práctica el solapamiento de puntuaciones entre sanos y enfermos ser5 parcial, generando curvas Roc intermedias entre las dos situaciones planteadas (Weinstein y Fineberg, 1980).

Análisis de Redes Criminales

OBJETIVO:
El presente trabajo describe un Proyecto de Minería de Datos en el ámbito de la información criminal, analizando los homicidios dolosos cometidos en la República Argentina e identificar las redes o bandas criminales, sus líderes o integrantes clave y como se relacionan entre sí.

RESULTADO:
El Proyecto COPLINK fue creado en el año 1997 en el Laboratorio de Inteligencia Artificial[2] de la Universidad de Arizona, en Tucson, con el objetivo de servir de modelo para ser llevado a nivel nacional. Recientemente se ha desarrollado la versión comercial, denominada COPLINK Solution Suite [Coplink, 2007]. Coplink está compuesto por dos sistemas integrados: Coplink Connect y Coplink Detect. El primero busca compartir información criminal entre distintos departamentos policiales, mediante un fácil acceso y una interfase sencilla, integrando distintas fuentes de información. El segundo está diseñado para detectar de forma automática distintos tipos de asociaciones entre las bases de datos mediante técnicas de minería de datos. Ambos sistemas presentan una interfase visual amigable [Chen et al., 2004]. Entre otras aplicaciones Coplink provee Análisis de Redes Criminales [Chen et al., 2004], la cual consiste en: identificar las redes o bandas criminales, sus líderes o integrantes clave y como se relacionan entre sí. En primer lugar se utiliza la técnica de concept space para extraer relaciones de los sumarios policiales y construir una posible red de sospechosos. La fuerza del vínculo entre dos sospechosos se mide en base a la frecuencia de hechos en los que participaron ambos. Luego se utiliza clustering jerárquico para partir la red en subgrupos y block modeling para identificar patrones de interacción entre los mismos. Finalmente se calcula el baricentro de cada subgrupo para determinar su miembro clave o líder.

Minería de Datos Educacional en Ambientes Virtuales de Aprendizaje

[2]Valenga(2007).

OBJETIVO:
La razón por la cual el uso de la Minería de Datos Educacional (MDE) es muy apropiada para descubrir información "escondida" en las bases de datos de un AVA. Los métodos pueden ser aplicados para explorar, visualizar, y analizar datos de e-learning con la finalidad de identificar patrones útiles aplicables a la evaluación de la actividad del usuario en la web y descubrir más profundamente como aprenden los estudiantes.
MUESTRA:
Construcción de un perfil cognitivo del estudiante basado en los datos filtrados por técnicas de Minería de Datos Educacional: la obtención de perfiles individuales y grupales permitirán la adaptación del material instruccional (Objetos de aprendizaje, tests, etc.)
• Análisis de las herramientas de Minería de Datos más apropiadas para los datos recolectados: se estudiarán las diversas técnicas de la Minería de Datos a fin de adaptarlas a nuestros objetivos.
• Adquisición de información desde un AVA[3] : decisión sobre las bases de datos a usar, generados por la acción del estudiante.

[3]Ambientes Virtuales de Aprendizaje

RESULTADO:
Se ha desarrollado un modelo de perfiles cognitivos considerando una variable lingüística de salida (nivel de conocimiento) y tres variables lingüísticas de entrada (progresión de notas, nivel de aprobación de las pruebas y nota final respecto a la media del curso). El modelo se completa con 27 reglas difusas que capturan la experticia de los profesores. Se espera un mejoramiento incremental de los perfiles con nuevas técnicas de Minería de Datos [4] y nuevas variables lingüísticas.

Diseño y aplicación de una batería multidimensional de indicadores de rendimiento para evaluar la prestación competitiva en el fútbol de alto nivel

OBJETIVO:
El objetivo del presente estudio fue el de diseñar y aplicar una batería multidimensional compuesta por cinco indicadores de rendimiento: Índice de Iniciativa de Juego (IIJ), Índice de Carga Física (ICF), Índice de Volumen de Juego Ofensivo (IVJO), Índice de Precisión en el Juego Ofensivo (IPCJO) e Índice de Progresión en el Juego Ofensivo (IPGJO), orientada a la evaluación de la prestación competitiva de equipos de fútbol de alto nivel y a la diferenciación del perfil de rendimiento obtenido por los equipos ganadores y perdedores durante los partidos.

MUESTRA:
Una vez comprobada la hipótesis de distribución normal de los cinco indicadores de rendimiento señalados mediante la prueba de Kolmogorov-Smirnov [5], se procedió a la comparación de las medias obtenidas por ganadores y perdedores en cada uno de ellos, por medio de la prueba t para muestras independientes. Finalmente, se realizó un análisis de regresión logística tomando como variable dependiente el resultado del partido (ganado o perdido) y como covariables los indicadores de rendimiento que habían mostrado diferencias estadísticamente significativas entre ganadores y perdedores.

[4] Huapaya (2012).
[5] Valez Vázquez (2011).

Indicador de rendimiento	Z de Kolmogorov-Smirnov	p
IIJ	,803	,539
ICF	,566	,906
IVJO	,804	,537
IPCJO	,731	,659
IPGJO	,837	,486

RESULTADO:
Se utilizó la prueba de Kolmogorov-Smirnov para contrastar la hipótesis de distribución normal de los cinco indicadores de rendimiento utilizados en la investigación. En la Tabla se ofrecen los valores Z de Kolmogorov-Smirnov para cada uno de los indicadores de rendimiento y las probabilidades asociadas respectivas. Se observa que ninguno de los valores calculados alcanza significatividad estadística, por lo que se concluye que las distribuciones empíricas se ajustan en todos los casos a la distribución normal.

Aplicación de áboles de decisión en modelos de riesgo crediticio

OBJETIVO
En este artículo se presentan algunos riesgos a los que se enfrenta una institución financiera, su clasificación y definición, centrándose específicamente en el riesgo crediticio, para el que se presenta el marco legal: los enunciados básicos del Acuerdo de Basilea II y la reglamentación del sistema de administración de riesgo crediticio de la Superintendencia Bancaria en Colombia. se presenta la utilización de los aboles de decisión como herramienta para el cálculo de probabilidades de incumplimiento en crédito, mostrando sus ventajas y desventajas.
MUESTRA
La necesidad de medir el riesgo y promover que las instituciones financieras hagan una correcta evaluación de ellos ha sido un esfuerzo de todos los bancos a nivel mundial. El comité de supervisión bancaria de Basilea, ha sido pre Establecido por los bancos centrales del grupo de los 10, a finales de 1974 cuyos miembros son: Bélgica, Canadá, Francia, Alemania, Italia, Japón, Luxemburgo, Holanda, España, Aplicación de arboles de decisión en modelos de riesgo crediticio 143 cursor de la reglamentación de la medición integral de riesgos y el adecuado provisionamiento de capitales, para sobrellevar los posibles riesgos incurridos y evitar la quiebra de las instituciones financieras.

RESULTADO:
Los árboles de decisión [6] se presentan como una herramienta efectiva para la predicción de probabilidades de incumplimiento, no solo a nivel de capacidad de discriminación (potencia), estabilidad a través del tiempo, sino como una herramienta de fácil entendimiento que permite potencializar sus usos y servir además de la predicción, para la planeación de estrategias comerciales de venta de servicios, estrategias de cobranza entre muchas otras.

Investigaciones Nacionales

Datos Poblacionales reales de cáncer de México

OBJETIVO:
En este trabajo se propone una mejora al algoritmo heurístico de agrupamiento (clustering) K-means [7] y se muestran los resultados de su aplicación a bases de datos poblacionales reales de cáncer de México.

MUESTRA:
Los datos de mortalidad fueron obtenidos del "Núcleo de acopio y análisis de información de salud" (NAAIS) del Instituto Nacional de Salud Pública (INSP), en particular se seleccionaron los datos de muerte por cáncer de pulmón y estomago del año 2000.

[6] Hernández (2004).
[7] Pérez (2007).

RESULTADO:
Se utilizó la prueba de Kolmogorov-Smirnov para contrastar la hipótesis de distribución normal de los cinco indicadores de rendimiento utilizados en la investigación. En la Tabla se ofrecen los valores Z de Kolmogorov-Smirnov para cada uno de los indicadores de rendimiento y las probabilidades asociadas respectivas. Se observa que ninguno de los valores calculados alcanza significatividad estadística, por lo que se concluye que las distribuciones empíricas se ajustan en todos los casos a la distribución normal.

4.1. Marco teórico

4.1.1. Machine Learning

Machine Learning[8] es un subcampo de las ciencias computacionales y una rama de la inteligencia artificial que tiene por objetivo permitir a las computadoras un aprendizaje constante sobre las situaciones a las que se enfrentan para poder resolver problemas y optimizar soluciones en menor tiempo que con cálculo tradicional. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos.
En muchas ocasiones el campo de actuación del Machine Learning se complementa con el de la estadística computacional, puesto que ambas disciplinas trabajan con el análisis de datos. Sin embargo, también se centra en el estudio de la complejidad computacional de los problemas, puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos programados en un algoritmo.
El Machine Learning tiene una amplia gama de aplicaciones, incluyendo motores de búsqueda, diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, juegos y robótica.
Algunos sistemas de aprendizaje automático intentan eliminar toda necesidad de intuición o conocimiento experto de los procesos de análisis de datos, mientras otros tratan de establecer un marco de colaboración entre el experto y la computadora. De todas formas, la intuición humana no puede ser reemplazada en su totalidad, por la razón de que la transformación de las variables, la interpretación de los resultados y la creación de algoritmos necesita de un experto con razonamiento para poderse generar.
El aprendizaje automático tiene como resultado un modelo para resolver una tarea dada. Entre los modelos se distinguen

[8] González (2006)

- Los modelos geométricos: construidos en el espacio de instancias y que pueden tener una, dos o múltiples dimensiones.

- Los modelos probabilísticos: que intentan determinar la distribución de probabilidades descriptora de la función que enlaza a los valores de las características con valores determinados.

- Los modelos lógicos: que transforman y expresan las probabilidades en reglas organizadas en forma de árboles de decisión.

El Machine Learning se divide en dos áreas principales: aprendizaje supervisado y aprendizaje no supervisado. Aunque a-priori podemos suponer que el supervisado se refiere a la técnica de predecir resultados con una intervención humana constante esto no es así, está clasificación se refiere más a las intenciones que se tienen con el tratamiento de los datos.
Uno de los usos más extendidos del aprendizaje supervisado consiste en hacer predicciones a futuro basadas en comportamientos o características que se han visto en los datos ya almacenados (el histórico de datos). El aprendizaje supervisado permite buscar patrones en datos históricos relacionando todos campos con un campo especial, llamado campo objetivo. El aprendizaje supervisado se caracteriza por contar con información que especifica qué conjuntos de datos son satisfactorios para el objetivo del aprendizaje.
Por otro lado, el aprendizaje no supervisado usa datos históricos que no están etiquetados. El fin es explorarlos para encontrar alguna estructura o forma de organizarlos. En el aprendizaje no supervisado el programa no cuenta con datos que definan qué información es satisfactoria o no. El objetivo principal de estos programas suele ser encontrar patrones que permitan separar y clasificar los datos en diferentes grupos, en función de sus atributos.

4.1.2. Aprendizaje no supervisado

La historia de la clasificación comienza con la sistemática de Carl Von Linné en 1971, permitía clasificar animales y plantas según su género y especie. La clasificación moderna denominada "taxonomía numérica" se inicia en 1957 con la necesidad de proponer criterios objetivos de clasificación, Sokal, Sneathy Michener revelan un buen sentido de la teoría formalizada, la obra de base histórica, es la de Sokal y Sneath (1963), los primeros manuales publicados fueron los de Lerman(1970), Anderberg(1973), Benzécri (1973), Hartigan (1975), Lerman (1981) y Gordon (1981) principios fundamentales. Algunos autores (Benzecri, Jardine, Sibson, Johnson) relacionaron las clasificaciones jerárquicas con espacios ultra métricos aunque la pro-

piedad ultramétrica ya era conocida en otros campos de la matemática.
El Análisis de Conglomerados es una técnica de Análisis Exploratorio[9] de Datos para resolver problemas de clasificación. El objetivo consiste en ordenar elementos (personas, cosas, animales, plantas, variables) en grupos (conglomerados o clústers) de forma que el grado de asociación/similitud entre miembros del mismo cluster sea más fuerte que el grado de asociación/similitud entre miembros de diferentes Conglomerados.
En el análisis cluster, a diferencia del análisis discriminante[10] (donde los grupos están establecidos a priori y la función discriminante permite reasignar los elementos a los grupos), los conglomerados son desconocidos y el proceso consiste en su formación de modo óptimo, aglutinando unidades homogéneas.
Está claro que los grupos formados vendrán determinados por las múltiples variables usadas en el estudio, pero el interés está en caracterizar y resumir entre las múltiples variables, algo inherente a cada grupo. Tras el resultado del agrupamiento surge la necesidad de encontrar respuestas a esas agrupaciones.
Una vez establecida empíricamente la clasificación, para que ésta sea útil, puede ser analizada detenidamente con objeto de descubrir las claves o propiedades que han producido tal agrupamiento. La aparición de esta estructura puede llevar al investigador a aprehender aspectos o propiedades de los individuos que de otro modo habrían pasado inadvertidos. Lo que puede conducir, a su vez, a plantear nuevas hipótesis de trabajo y nuevas investigaciones desde diferentes perspectivas. Generalmente para el análisis de los grupos ya formados no se utilizan técnicas sino conocimiento y experiencia en el campo de dichos datos.
En general, estas técnicas son altamente recomendadas para cualquier tipo de negocio que desee tener un análisis robusto y bien justificado de todos los datos que genera. Son utilizados por empresas líderes en tecnología y pequeños pioneros que buscan ponerse a la vanguardia en el análisis de mercado y consumidores.
La importancia del análisis de datos puede influir altamente en los niveles de efectividad de una campaña publicitaria, de características de un nuevo producto, para agrupar tipos de consumidores y poder redireccionar los esfuerzos o tendencias del mercado, para determinar un nuevo mercado meta, entre muchas otras cosas.
Para realizar este análisis podemos encontrarnos dos tipos fundamentales de métodos para realizar clasificaciones: Jerárquicos y No Jerárquicos. En los métodos jerárquicos la clasificación resultante tiene un número creciente de clases anidadas mientras que en el segundo no. Los métodos pueden dividirse en aglomerativos y divisivos. En los primeros se parte de tantas clases como objetos tengamos que clasificar y en pasos sucesivos vamos obteniendo clases de objetos similares, mientras que en los segundos se parte de una única clase formada por todos los objetos que se va dividiendo en clases sucesivamente.
A continuación se mencionan y definen 2 técnicas que fueron clave para realizar el análisis y que por su potencia matemática lograron arrojar resultados claros y pre-

[9] K.Fukunaga (2013)
[10] Devijver (1982)

cisos para poder asignar categorías a los datos.
a

K-means

K-means[11] es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano.
El término "k-means"fue utilizado por primera vez por James MacQueen en 1967, aunque la idea se remonta a Hugo Steinhaus en 1957. El algoritmo estándar fue propuesto por primera vez por Stuart Lloyd en 1957 como una técnica para modulación por impulsos codificados, aunque no se publicó fuera de los laboratorios Bell hasta 1982. En 1965, E. W. Forgy publicó esencialmente el mismo método, por lo que a veces también se le nombra como Lloyd-Forgy. Una versión más eficiente fue propuesta y publicada en Fortran por Hartigan y Wong en 1975/1979.
El algoritmo más común utiliza una técnica de refinamiento iterativo, donde dado un conjunto inicial de k centroides, el algoritmo continúa alternando entre dos pasos:

- Paso de asignación: Asigna cada observación al grupo con la media más cercana, se asigna a un grupo (centroide) y solo a uno, aunque la observación pudiese converger en más de uno.

- Paso de actualización: Calcular los nuevos centroides como el centroide de las observaciones en el grupo.

El algoritmo se considera que ha convergido cuando las asignaciones ya no cambian. Como se trata de un algoritmo heurístico, no hay ninguna garantía de que convergen al óptimo global, y el resultado puede depender de los grupos iniciales. Como el algoritmo suele ser muy rápido, es común para ejecutar varias veces con diferentes condiciones de partida. Sin embargo, en el peor de los casos, k-means puede ser muy lento para converger.

Gaussiano-Mixto

El Gaussiano-Mixto[12] un método probabilístico para obtener una clasificación difusa de las observaciones. Se calcula la probabilidad de pertenecer a cada grupo y generalmente se logra una clasificación asignando cada observación al grupo más probable. Estas probabilidades también se pueden usar para interpretar clasificaciones sospechosas.

[11] Wong (1979), Silverman (2002)
[12] P. Aguilera (2015)

El modelado de mezclas es muy flexible, el objetivo de los modelos de mezcla es estructurar el conjunto de datos en varios clusters que sean finamente diseñados por densidades de probabilidad. Este método permite que los clusters se generen de forma muy precisa, sin necesidad de un tratamiento robusto de datos ya que asigna curvas de probabilidad que se ajustan a la naturaleza de los datos.

4.1.3. Aprendizaje supervisado

Como pudimos notar el análisis de datos permite que las empresas se pongan en ventaja ante los consumidores pues las señales ocultas en los datos proveen la suficiente información para poder predecir comportamientos o inducir pensamientos. Después de todo ¿cómo crees que Amazon consigue ofrecer a pie de página esos productos que pueden interesar al usuario? ¿De qué modo Netflix planea el abanico de recomendaciones al espectador? ¿Cómo pueden las instituciones crediticias autorizan créditos a personas "no confiables"?
La clave está en la incorporación del learning machine en su estrategia. El aprendizaje automático[13] no es simplemente una forma más popular de aproximarse a la analítica predictiva, sino que podría considerarse como un universo separado, cuyo potencial tiene que ver con:

1. El uso de la lógica para modelar la planificación, el razonamiento y la resolución de cuestiones que no son problemas técnicos o relacionados con la tecnología, sino que tienen que ver con las personas y sus necesidades.

2. La utilización de big data para hacer predicciones o sugerencias calculadas basadas en abrumadoras cantidades de datos, ya sean estructurados y no estructurados, registros históricos y datos recientes, incluso recogidos en tiempo real.

3. El foco en la resolución de problemas, a los que busca dar respuesta mediante diferentes algoritmos.

No todos los algoritmos de learning machine tienen la misma naturaleza. Algunas de ellos funcionan mejor para ciertos objetivos de negocio, mientras que otros no podrían contribuir generando valor en esos mismos campos. Lo mismo sucede cuando se trata de evaluar las características del conjunto de datos a analizar.
El fin del modelaje supervisado es hacer aprender al programa sobre comportamientos que arrojan los datos, para así poder predecir comportamientos o tendencias

[13] S. Kotsiantis (2007)

que a simple vista no son tan sencillos de determinar. Mientras más tratamiento de datos se haga y más poder de cómputo requiera el algoritmo más exactas suelen ser las predicciones, de ahí la importancia del manejo de los datos y la selección del modelo óptimo para el problema, ya que no todos los problemas tienen el mismo tratamiento.
A continuación se mencionan algunos algoritmos empleados para el aprendizaje supervisado, que en particular fueron utilizados para la resolución de nuestro problema.

Árboles de decisión

Los árboles de decisión[14], son herramienta que logra servir como apoyo a una toma de decisiones informada, al exponer las distintas opciones y sus posibles consecuencias, incluidos los resultados de eventos fortuitos, los costos de recursos y la utilidad. A la hora de trabajar con este algoritmo es necesario tener en cuenta que hay que conocer el número mínimo de preguntas simples (es decir, las que puedan responderse con un sí o un no) que es preciso lanzar para evaluar la probabilidad de tomar una decisión correcta. La ventaja de los árboles de decisión es que permiten abordar el problema de una manera estructurada y sistemática para llegar a una conclusión lógica.

Regresión logística

Este algoritmo[15] se encarga de medir la relación entre la variable dependiente categórica y una o más variables independientes. Así, aplicando una función logística se pueden estimar las probabilidades de ocurrencia de un suceso.

Maquinas de Vector de Soporte

Una máquina de vectores de soporte[16] construye un hiperplano óptimo en forma de superficie de decisión, de modo que el margen de separación entre las dos clases en los datos se amplía al máximo. Los vectores de soporte hacen referencia a un pequeño subconjunto de las observaciones de entrenamiento que se utilizan como soporte para la ubicación óptima de la superficie de decisión. Las máquinas de vectores de soporte pertenecen a una clase de algoritmos de Machine Learning denominados métodos kernel[17] y también se conocen como máquinas kernel.

[14] Rokach (2008)
[15] Agresti (2002), Hosmer (2000)
[16] Schölkopf (2002)
[17] Bennett (2000)

Redes Neuronales

Una Red Neuronal Artificial[18] (RNA) es un modelo matemático inspirado en el comportamiento biológico de las neuronas y en cómo se organizan formando la estructura del cerebro. Respecto a su funcionamiento, el cerebro puede ser visto como un sistema inteligente que lleva a cabo tareas de manera distinta a como lo hacen las computadoras actuales. Si bien estas últimas son muy rápidas en el procesamiento de la información, existen tareas muy complejas, como el reconocimiento y clasificación de patrones, que demandan demasiado tiempo y esfuerzo aún en las computadoras más potentes de la actualidad, pero que el cerebro humano es más apto para resolverlas, muchas veces sin aparente esfuerzo (por ejemplo, el reconocimiento de un rostro familiar entre una multitud de otros rostros).
Para este algoritmo se necesita un conjunto de entradas, los pesos sinápticos correspondientes a cada entrada, una función de agregación, una función de activación y una salida. Las entradas son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida es la respuesta a tal estímulo. La neurona puede adaptarse al medio circundante y aprender de él modificando el valor de sus pesos sinápticos, y por ello son conocidos como los parámetros libres del modelo, ya que pueden ser modificados y adaptados para realizar una tarea determinada.

Random Forest

Es un método[19] que se basa en combinar una cantidad grande de árboles de decisión, independientes entre sí, probados sobre conjuntos de datos aleatorios con igual distribución. La fase de entrenamiento consiste en crear muchos árboles de decisión, construidos a partir de datos de entrada ligeramente distintos. Una vez creados se hace lo siguiente:
Se selecciona aleatoriamente con reemplazamiento un porcentaje de datos del total. En cada nodo, al seleccionar la partición óptima, tenemos en cuenta sólo una porción de los atributos, los cuales son elegidos al azar.
Cada árbol se evalúa de forma independiente y la predicción del bosque será la media de todos los árboles. La porción de árboles que toman una misma respuesta se interpreta como la probabilidad de la misma.

Gradient Boosting Trees

Esta técnica construye el modelo de una manera escalonada en etapas, al igual que otros modelos estilo "Boosting"[20], y los generaliza al permitir la optimización de una

[18] Bishop (1995)
[19] Dietterich (2000)
[20] Mason (1999)

función de pérdida diferenciable arbitraria. La idea de la técnica Gradient Boosting, se originó por la observación de Leo Breiman, donde el Boosting puede ser interpretado como un algoritmo de optimización en una función de coste adecuada.
Los Gradient Boosting de regresión explícita, fueren desarrollados posteriormente, simultáneamente con el Gradient Boosting más general y funcional. Los dos últimos trabajos introdujeron la visión abstracta de impulsar algoritmos como algoritmos de Functional Gradient Descent (Descenso Gradual Funcional). Es decir, los algoritmos que optimizan un coste funcional sobre el espacio funcional por la elección de una función iterativa (Hipótesis Débil) que apunta en la dirección del gradiente negativo. Este punto de vista funcional del "Boosting" lo ha llevado al desarrollo, a impulsar algoritmos en muchas áreas de aprendizaje automático y estadísticas más allá de la regresión y la clasificación. En conclusión, el método Gradient Boosting representa un algoritmo de aprendizaje automático aplicable en ámbitos muy generales, y con el que se puede obtener un gran rendimiento.

XGBoost

XGBoost[21] es una implementación del ya conocido algoritmo Gradient Boosting. Este modelo es a veces descrito como una caja negra (blackbox), refiriéndose a que trabaja bien pero no es trivial entender como lo hace. Ciertamente, el modelo está construido por cientos hasta incluso miles de árboles de decisión. Por ello es muy difícil que un ser humano pueda ser capaz de tener una visión general del modelo. Mientras XGBoost es conocido por sus capacidades de velocidad y precisión en la predicción, también es conocido por venir con varias funciones que ayudarán a que sea entendido el funcionamiento.
Para construir un árbol, el conjunto de dato es dividido recursivamente varias veces. AL final del proceso se habrán obtenido grupos observatorios. Cada división operativa es llamada Split (división). Cada grupo en cada nivel de división se les llama ramas, y el nivel más profundo hoja.
En el modelo final, estas hojas se supone que deben ser tan puros como sea posible para cada árbol, es decir que cada hoja debe ser de una clase, aunque esto no sea logrado, es el objetivo que se debe tratar de conseguir, en un mínimo de divisiones. No todos los Split son igual de importantes. Básicamente la primera división de un árbol tendrá un mayor impacto en la purea que, por ejemplo, la más profunda. Intuitivamente, se entiende que la primera división hace la mayor parte del trabajo y que las siguientes divisiones se centran en partes más pequeñas del conjunto de datos que han sido clasificadas erróneamente en el primer árbol.
De la misma manera, en Boosting, se intenta optimizar la correcta clasificación en cada ronda, por lo tanto el primer árbol va a hacer la mayor parte del trabajo y lo siguientes árboles se centrarán en los trabajos restantes, las partes que no han "aprendido" correctamente en los árboles anteriores. La mejora que se ha llevado a cabo en cada división puede ser medida, esta es llamada ganancia. Y por último

[21] Chen (2016))

para entender el funcionamiento de XGBoost hay que tener claro que cada división es hecha en una característica a un valor en concreto.

Si bien al oír hablar de inteligencia artificial o machine learning puede parecer que es un área reservada para un reducido grupo de negocios de perfil ultra-innovador y vocación disruptiva, en la actualidad, los usos del aprendizaje automático ya están tan extendidos y las herramientas que permiten incorporarlos a la estrategia de negocio tan avanzadas, que cada vez son más quienes pueden aprovecharse de su potencial para ofrecer a sus clientes una experiencia más personalizada, de muchas formas diferentes.

5. El conjunto de datos y sus variables

El conjunto de datos consta de 10 variables por registro, con al rededor de 1048 registros, tomados durante 6 días de 7.00 a 15.00 . De las 10 variables la mitad son continuas mientras que las restantes son discretas. Cada registro describe una compra realizada en el establecimiento por grupo de persona.

A continuación daremos una breve descripción de cada variable o columna del conjunto de datos, para así tener una mayor compresión de este.

Variables continuas:
• Dinero Gastado: Cantidad total gastada por compra
• Total de Personas: El número de personas del grupo.
• Total de hombres y Total de mujeres: La cantidad de hombres y mujeres por grupo respectivamente
• Hora: La hora que se tomo el pedido

Variables discretas:
• Edad: La edad aproximada de las personas del grupo.
• Local/Llegar: Si el grupo consumió los productos en el establecimiento o no.
• Bebida: El tipo de bebida que consumieron, si no consumieron bebidas, se representa con un 0.
• Comida: El tipo de comida que consumieron, si no consumieron comida, se representa con un 0.
• Día: El día que se realizo la transacción.

6. Análisis descriptivo de las variables

6.1. Análisis Univariante

En esta sección trataremos de ver la frecuencia de las diferentes variables que tenemos, así como de algunas relaciones entre ellos. Primero nos enfocaremos en el análisis univariado. Desarrollaremos un análisis diferente por variable. En el caso de que la variable sea continua, se realizará histogrma que muestre el comportamiento de la variabel. En el caso de discretas se realizara un simple conteo.

De manera general lo datos se visualizan de la siguiente manera:

	Dinero Gastado	Total Personas	Total Hombres	Total Mujeres
count	1040.000000	1040.000000	1040.000000	1040.000000
mean	91.359806	1.429498	0.750405	0.683955
std	76.343403	0.903801	0.776650	0.763803
min	0.000000	1.000000	0.000000	0.000000
10 %	29.000000	1.000000	0.000000	0.000000
50 %	64.000000	1.000000	1.000000	1.000000
95 %	226.400000	3.000000	2.000000	2.000000
max	511.000000	10.000000	7.000000	6.000000

6.1.1. Variables discretas:

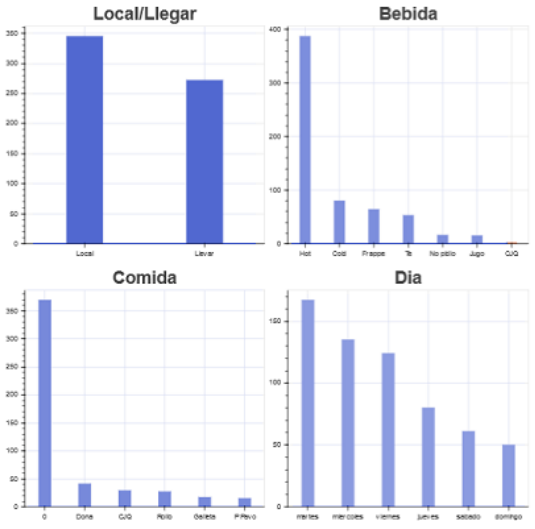


Figura 1: Frecuencia en los datos de las Variables Discretas

Se observa que en la frecuencia en los datos, hay preferencia por consumir los prodcutos en el establecimiento , así como que hay mayor preferencia por consumir bebidas a alimentos, esto es de esperarse por el rol que desempeña el negocio.

6.1.2. Variables continuas:

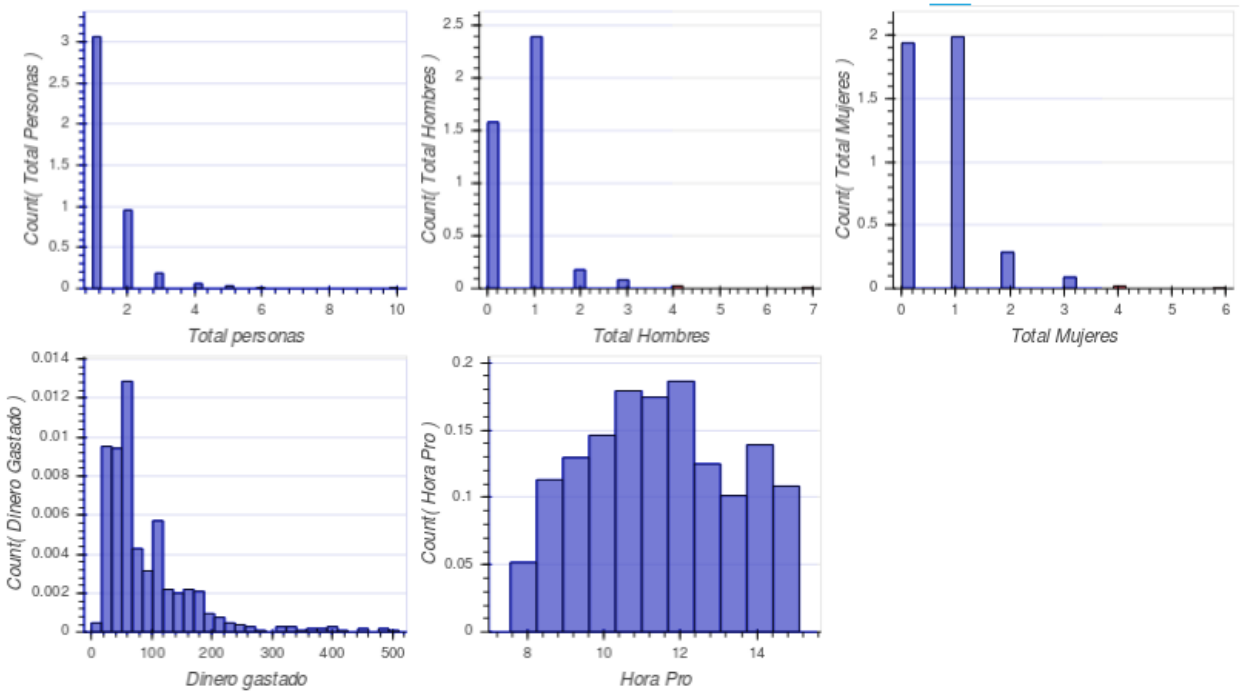


Figura 2: Histograma de todos los datos continuos

Se nota que la cantidad de personas que acuden por grupo son de una persona o dos personas. Así tambié se destaca que la media de gasto ronda por el intervalo del 0 al 100, al igual se observa que los horario popular es el de medio día.

6.2. Análisis Multivariado

En esta sección analizaremos la relación que existe entre nuestras variables tanto discretas como continuas, en este ultimo caso la categoría se compone de las siguientes entidades: Dinero gastado, Total personas, Total Hombres, Total Mujeres y la Hora. Se propone encontrar la correlación entre los campos que no sean discretos, para los datos categóricos nos enfocaremos en resaltar las variables que sean de importancia para la visión de nuestro negocio y así obtener información útil para el funcionamiento de el establecimiento (starbucks).

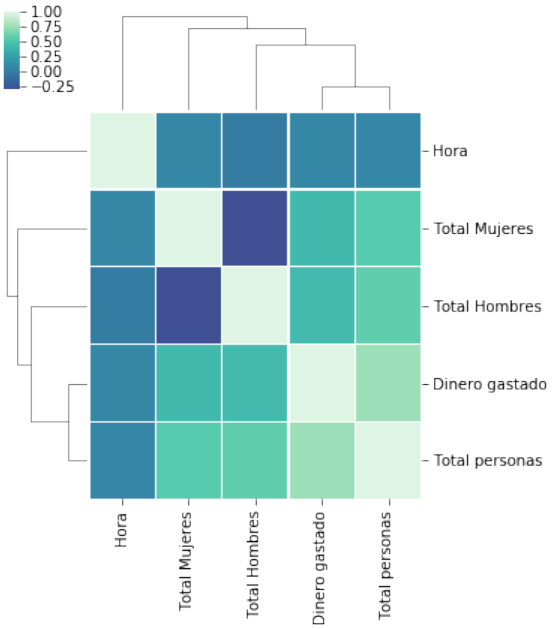
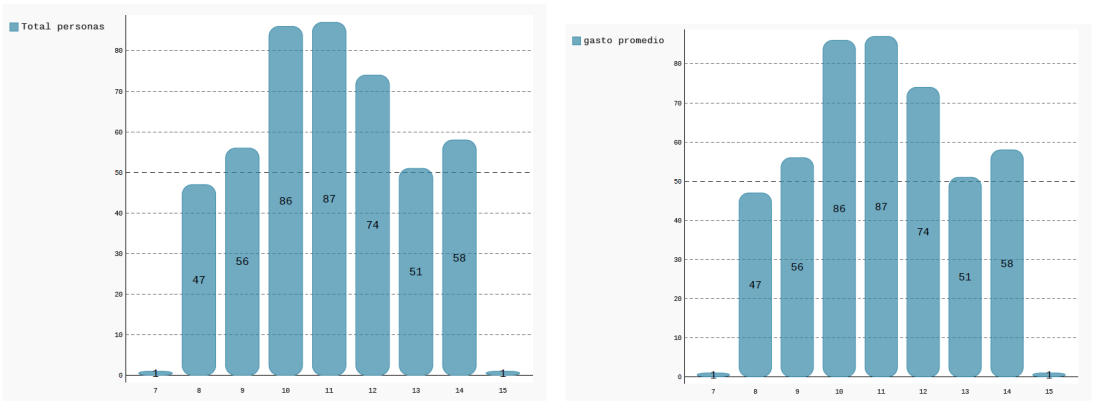


Figura 3: Correlacion.

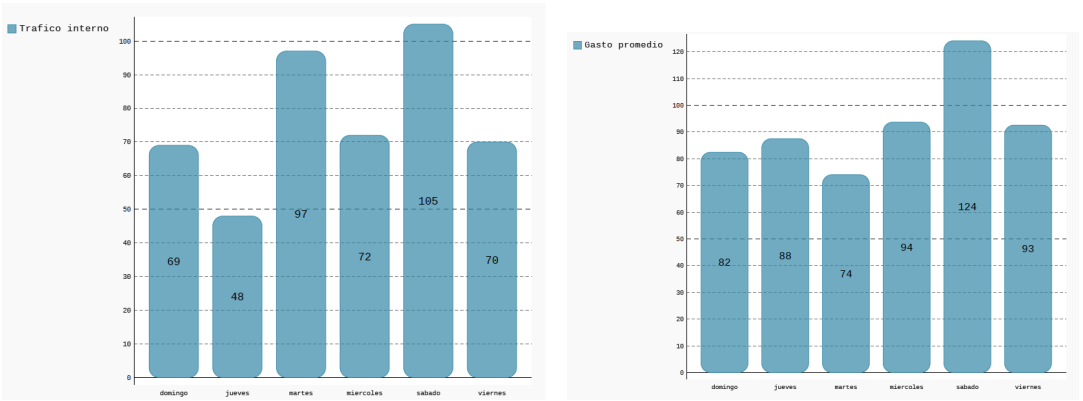
En la gráfica anterior podemos observar como la correlación es mas fuerte entre el dinero gastado y el numero de personas, dicha correlación resulta muy obvia aun sin esta gráfica, lo que no resulta ser del todo evidente es la poca interacción entre las horas de llegada y las demás variables, por ello resulta necesario ver el flujo de dinero promedio y el trafico interno que ocurre en cada hora, ya que estos indices reflejan el objetivo que impulso este trabajo. Derivada de nuestro punto anterior surge la necesidad de visualizar mas a detalle el comportamiento de ciertas variables dada una hora.



(a) (b)

Figura 4: Horas.

En los gráficos anteriores podemos observar que tanto los ingresos como la cantidad de personas que visitan el establecimiento alcanza un auge entre las 10 am y las 12 pm, esto refuerza nuestra conclusion acerca de la relacion entre enl numero de personas y el gasto que Se realiza, si analizamos los mismos factores pero a una escala diaria obtendremos lo siguiente:

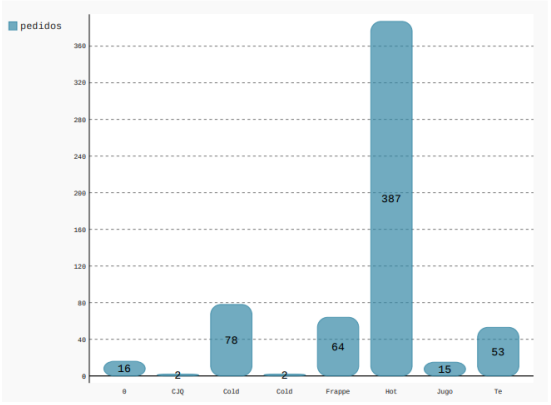


(a) (b)

Figura 5: Días.

analizando estos resultados podemos ver que el gasto promedio no se vio directamente afectado por el numero de personas puesto que en la figura (a) el dia sabado destaca, sin embargo en la figura (b) no representa un puntero. Por ultimo consideramos prudente analizar que producto es el que mayor volumen de venta posee, pues así nos podremos dar una idea de lo que más le gusta y o necesita nuestra clientela, con base en ello intentaremos predecir qué productos se venderán mas en un futuro cercano.

Dada la grafica que se muestra a continuacion podemos observar que hay una clara preferencia por las bebidas calientes volumen, esto resulta estar conectado con un



(a)

Figura 6: Productos.

tema que trataremos más adelante (clusters) con el cual trataremos de averiguar el porqué de estas tendencias.

6.3. Valores Extremos

En las secciones 7.1 observamos la frecuencia de los datos donde destacamos los valores medios y los más frecuentes. Así como hicimos con estos es necesario también destacar los valores extremos o atípicos, los llamados .ºutliers", que sin duda son valores poco comunes y no dejan hacer un tratamiento correcto a los datos.

Cabe destacar que cada variable será tratada de diferente manera.

***Total de personas*, *Total de Mujeres* y *Total de Hombres*:**

El tratamiento de estas variables debe ser conjunto, ya que estás varibles están fuertemente relacionadas como se vio en la sección 7.2.

	Total personas	Total Hombres	Total Mujeres
count	1040	1040	1040
mean	1.429498	0.750405	0.683955
std	0.903801	0.776650	0.763803
min	1	0	0
10 %	1	0	0
50 %	1	1	1
99 %	5	3	3
max	10	7	6

Observamos que en el caso que no es conveniente remover todos los valores por debajo del percentil 10 % debido a que perderiamos el 50 % de la información. Ahora bien, en el caso del limite superior se recomienda remover a partir del percentil 99 % para así conservar variedad en los datos.

Dinero Gastado:

count	mean	std	min	1 %	10 %	50 %	95 %	99 %	max
1040	91.35	76.34	0	19.96	29	64	226.40	404.36	511

En este caso se recomienda eliminar todos los valores mayores del percentil 95 % y menores del percentil 1 %.

Hora:

count	mean	std	min	1 %	10 %	50 %	95 %	99 %	max
1040	11:28	01:55:14	07:33	08:07	08:51:48	11:22	14:38	14:58	15:07

Para la Hora se procede removiendo del percentil 1 % al 99 %.

Una vez aplicando este flitro los datos quedan con una dimensión de (932,9).

	Dinero gastado	Total personas	Total Hombres	Total Mujeres	Hora
count	932	932	932	932	932
mean	77.605	1.296	0.679	0.623	11.472
std	45.150	0.540	0.599	0.642	1.868
min	25	1	0	0	8.133
25 %	40	1	0	0	9.966
50 %	64	1	1	1	11.366
75 %	107	2	1	1	12.933
max	205	4	4	3	14.966

6.4. Valores Categoricos

Con el fin de cumplir con el objetivo a nuestro conjunto de datos les aplicaremos diversas transformaciones, las cuales garantizaran un mejor rendimiento en nuestros modelos.

Bebida:

Su importancia deriva por el rol de la empresa, y esta será la variable objetivo de nuestro primer modelo.

Bebida	Conteo	Frecuencia
Hot	387	0.627229
Cold	80	0.129660
Frappe	64	0.103728
Te	53	0.085900
Sin bebida	16	0.025932
Jugo	15	0.024311
CJQ	2	0.003241

Eliminamos los valores atípicos.

Bebida	Conteo	Frecuencia
Hot	387	0.728814
Cold	80	0.150659
Frappe	64	0.120527

Dado esto el modelo resultará en predecir cuales en que clase queda, de acuerdo a sus habitos como cliente. Por tanto es un problema de multiclases.

Día:

Como hemos visto en el apartado anterior el modelo resultante se reduce a la tarea de predecir a que clase pertenece cada vector. Ahora bien dado que los valores categoricos seran escensiales para esta predicción, es necesario transformarlos en algo númerico, no obstante, la transformación woe no es posible en este caso puesto que las clases resultantes son más de 2, por tanto se le dará un tratamiento más sencillo.

Aplicando la función **get dummies** (disponible en pandas) resulta en lo siguiente:

domingo	jueves	martes	miercoles	sabado	viernes
1	0	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	0	1	0
0	0	0	0	0	1

La función genera 6 nuevas variables, donde cada variable nos indica si la observación fue hecha en ese día o no.

En el siguiente capítulo se generará una nueva variable la cual será será de suma importancia para el entrenamiento de nuestros modelos. Esta variable se le dará el tratamiento anterior.

7. Modelación no supervisada

Una vez analizadas las variables, es ahora de interés crear clusters o grupos de los nuevos clientes de Starbucks La Cúspide de acuerdo a sus similitudes, para esto utilizaremos las variables continuas “Dinero gastado”, ”Total personas”, ”Total Hombres”, ”Total Mujeres” y “Hora promedio” y veremos que relaciones pueden encontrarse entre los grupos.
Para la mejor visualización de los clusters utilizaremos las técnicas de análisis de componentes principales y escalamiento multidimensional.
Antes de aplicar alguna técnica de clustering procedemos a transformar las variables, utilizando el método de min max scaler que transforma todas las variables hasta dejarlas en el mismo rango en este caso entre 0 y 1, aplicado en el escalamiento multidimensional, MDS, que busca preservar las distancias y el método de standard scaler cuyo resultado arroja las variables con media cero y varianza unitaria que fue aplicado para el análisis de componentes principales PCA.
El procedimiento que llevaremos a cabo para hacer los grupos es el de clustering de optimización como el K-MEANS.
A continuación se muestra la gráfica de inercia bajo este método de clustering

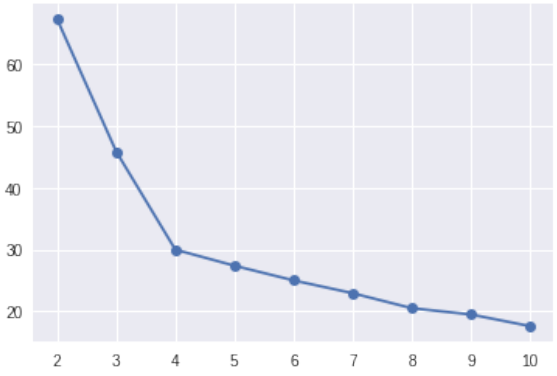


Figura 7: Gráfica de Inercia

Se sugiere la descomposición de los clientes en cuatro grupos principales. Las relaciones entre estos grupos quedan de la siguiente manera:

Cluster	Dinero Gastado	Total Personas	Total Hombres	Total Mujeres	Hora Promedio
0	-0.225609	-0.198764	-0.129385	0.086539	0.994521
1	-0.410006	-0.475598	-0.243878	-0.321162	-0.109907
2	2.295215	2.335332	1.511135	1.220453	0.408551
3	0.438543	0.631738	0.200826	0.536955	0.036689

Por lo que vemos que el primer grupo cl 0 gasta poco, no son muchas las personas que asisten al negocio, y se reúnen por la tarde.
El grupo 1 de igual forma gasta poco y son pocas las personas que llegan, la cantidad de hombres y mujeres es similar y llegan a la sucursal por las mañanas.
El cl2 es el grupo de consumidores que más dinero invierte en el local, llegan en grupos numerosos, las cantidades de hombres y mujeres que asisten tienen proporciones similares, y llegan por las tardes.
En el grupo 3 se gasta una cantidad considerable de dinero , representa una buena cantidad de la afluencia de la sucursal, y acuden temprano.
Así, de acuerdo a las características encontradas a través de las variables propuestas nombramos a los grupos de clientes de Starbucks “La Cúspide” como sigue:

cl0= Estudiante
cl1= Trabajador
cl2=Grupos de amigos
cl3= Parejas

Cluster	Frecuencia
Estudiantes	35.33 %
Trabajadores	34.85 %
Grupos de Amigos	5.67 %
Parejas	24.15 %

Además, haciendo el conteo de el número de personas que acuden a la sucursal,notamos que la mayoría de sus clientes pertenecen a el cluster de los grupos de amigos, que como se mencionó anteriormente acuden en cantidades numerosas, aspecto que es importante tomar en cuenta en cuestiones de recursos de la tienda.

7.1. Análisis de Componentes Principales

Ahora mediante componentes principales encontramos una transformación ortogonal que aplicada a nuestras 5 variables estandarizadas como se estableció previamente , nos devuelva dos variables nuevas de tal forma que reflejen el total de la varianza de nuestras variables originales.
Los componentes principales, que han sintetizado nuestras variables continuas en dos nuevas variables, constituyen aproximadamente el 88 % de la varianza. Disminuyendo dimensiones y permitiéndonos observar los grupos descritos anteriormente.

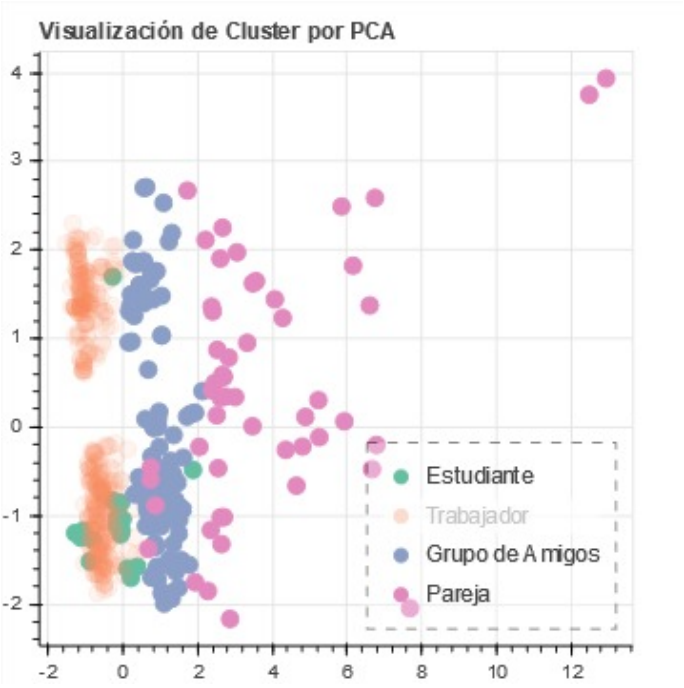


Figura 8: Visualización de Cluster por PCA

7.2. Escalamiento Multidimensional

Siguiendo el análisis de escalamiento multidimensional, sabemos que en esta técnica, entre más próximos sean dos objetos, éstos serán percibidos de forma semejante, de forma contraria entre más lejanos se encuentren poco tendrán que ver.
Así, los grupos con características similares son el de las parejas y grupos de amigos, así como también el de el estudiante y las parejas, por lo que las estrategias de negocio podrían ser similares para estos grupos de clientes.
Mientras que el grupo de los trabajadores tiene características que difieren a los de los otros 3 clusters por lo que probablemente deba ser incentivado con tácticas distintas.

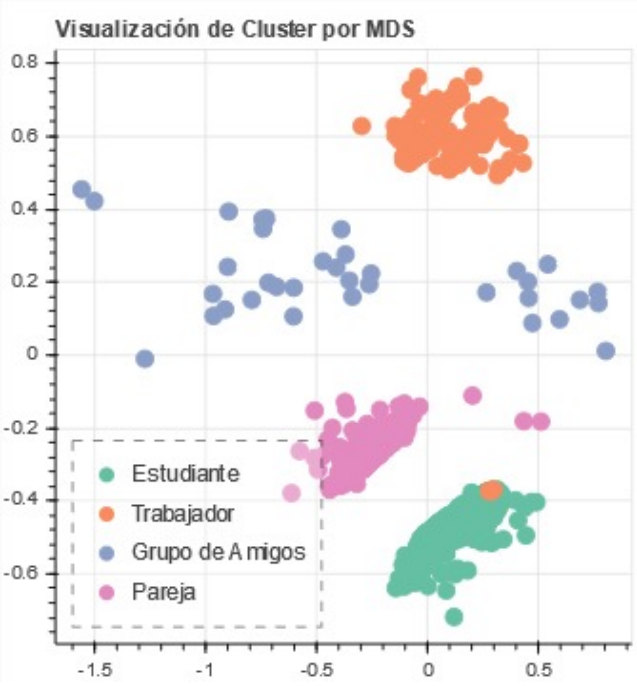


Figura 9: Visualización de Cluster por MDS

7.3. Análisis Estadístico de los Grupos

Ahora, después de haber definido bien los tipos de clientes, nos interesa conocer las características de estos grupos.
La primera gráfica nos representa el total de personas que pertenece a cada uno de los conglomerados, como puede notarse la mayoría de los clientes y debido a las característcas mencionadas al principio del trabajo sobre la localización del establecimiento, los clientes están liderados por trabajadores y por la poca recreación, lo grupos de amigos son pocos entre los que frecuentan Starbucks "La Cúspide"

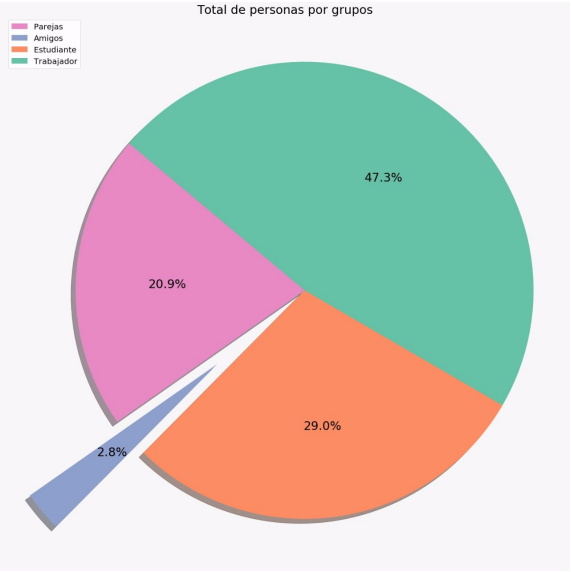


Figura 10: Porcentaje total de clientes por Grupo

Como puede observarse en la Figura 11, de entre los productos que son ofrecidos en el local, las bebidas calientes son las preferidas por el grupo de trabajadores,y como puede resultar intuitivo pensar, son las menos aclamadas por el grupo de los estudiantes. Si nos fijamos ahora en las bebidas frías, de nueva cuenta es el cluster de los trabajadores el que gusta este producto, seguido por las parejas, este mismo comportamiento es observado en el tercer producto ofrecido . Enfocándonos en los bocadillos, aquellos clientes que no compran bebida de nuevo están liderados por el cluster de los trabajadores,mientras que el grupo de amigos no tiene ninguna preferencia por este producto y el cluster de los estudiantes vuelve a tener una presencia casi nula.
De aquí se puede concluir que el grupo de trabajadores gusta de todos los productos, y el grupo de estudiantes tiene una baja simpatía hacia los productos, lo cual podría indicar que el consumo por parte de este grupo es poco.

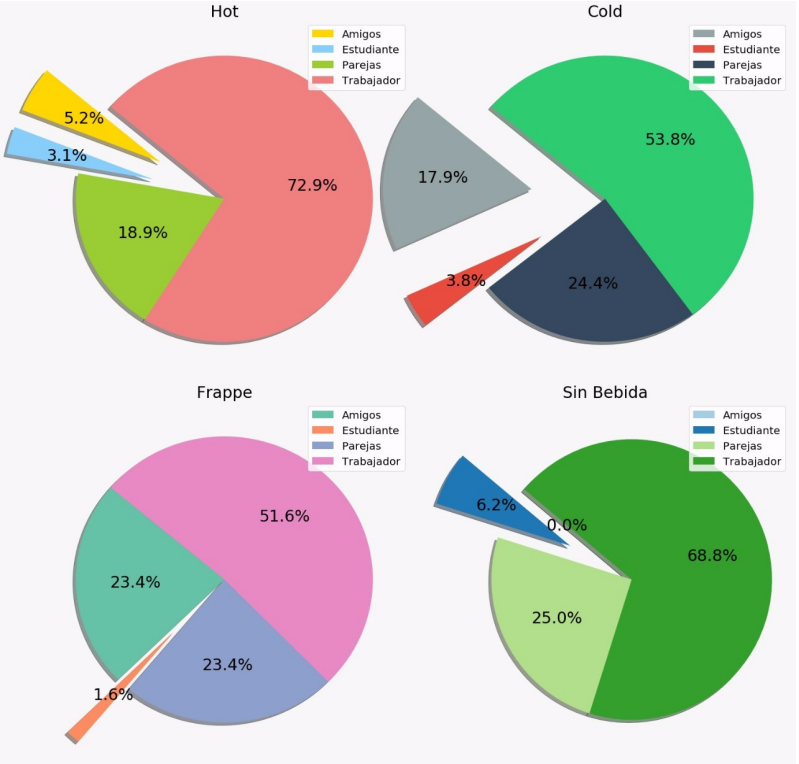


Figura 11: Preferencia de productos por cluster

Relaccionado a la figura anterior, analizamos ahora el gasto que en promedio realizan los grupos de clientes de la compañía,y sorpresivamente nos encontramos con que el cluster conformado por los trabajadores es el de consumo mínimo,seguido por el de los estudiantes, conociendo un poco de negocio y derivado de la cantidad de personas que llegan a el establecimiento, el grupo de amigos es el que más invierte en los artículos con una media de $266, sin embargo aunque la cantidad de gasto sea poca, el trabajador es un cliente frecuente, por lo que la entrada de capital por parte de este cluster es grande, mientras que como se observó en gráficas anteriores a pesar de que consumen mucho son pocos los pertenecientes al cluster de tipos de amigos.

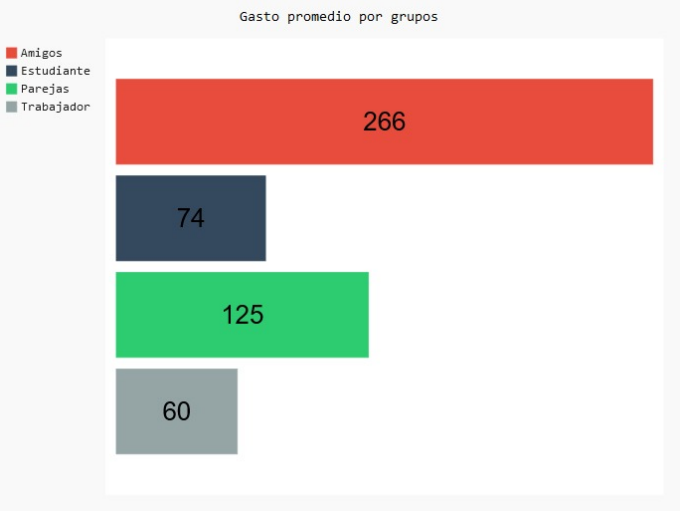


Figura 12: Gasto promedio por cluster

Se muestra a continuación la afluencia de clientes por grupo dependiendo la hora del día.

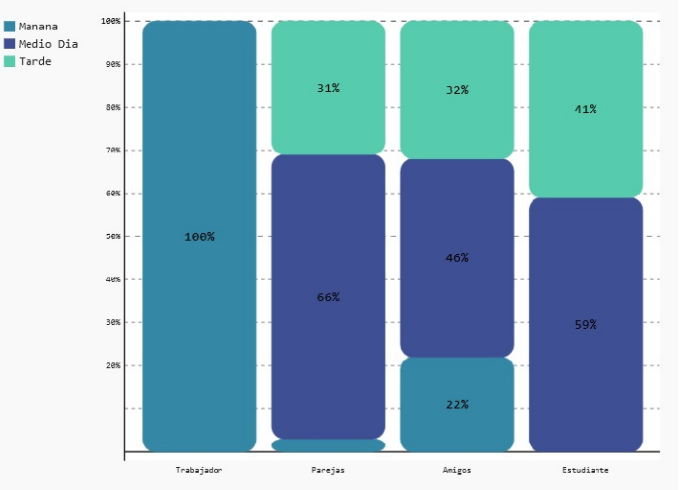


Figura 13: Porcentaje de llegada de los grupos por hora del día

La concentración de clientes de acuerdo al grupo se da de la siguiente manera, la totalidad de los trabajadores llega por las mañanas ;las parejas visitan el local mayormente al medio día de la misma manera que lo hacen los grupos de amigos, y los estudiantes se concentran en su mayoría al mediodía y la cantidad de personas que acude al local va disminuyendo conforme pasan las horas,de nuevo haciendo uso de la inteligencia del negocio, probablemente la razón sea el uso de las instalaciones para la elaboración de tareas escolares con la ayuda del wifi libre.

Llevado a cabo esta interpretación de los grupos y las variables que intervienen en su desempeño, notamos que los grupos clave son el de trabajadores y estudiantes, sin embargo sería lo óptimo rediseñar, evaluar y poner en práctica diversas políticas que incentiven el consumo entre todos los tipos de clientes, además de buscar la buena calidad en el servicio, atendiendo las horas en las que los clientes pueden ser más numerosos como es el caso del medio día o teniendo los recursos necesarios (ni excedentes ni faltantes) para dar a basto a los productos que son de la preferencia de cada grupo, tomando en cuenta de igual forma la hora del día en que se presente esta necesidad.

8. Modelación Supervisada

En la siguiente sección dos modelos predictivos que fueron desarollados con el dataset. El primer modelo buscará predecir la orden del cliente, usando un **Voting classifier** de varios algoritmos. Después el segundo modelo permitira predecir cuantos clientes vendrán en los siguientes 5 grupos de personas con los datos recabados hasta el momento para esto usaremos el algoritmo **XGBOOST**.

8.1. Modelo Predictivo (Comida)

Ya llegando a este punto es preciso destacar las variables que pueden ser necesarias para poder predecir el comportamiento, habiendo agrupado las observaciones:

- **Días**
- **Hora**
- **Total de Personas**
- **Local/Llegar**
- **cl**
- **Dinero gastado**

Este problema se reduce a predecir a que clase pertenece la observación, pero como se busca predecir antes de que se presente el cliente algunas de estas solo se pueden estimar.

Retomando todas las transformaciones, nuestra tabla resulta de la siguiente manera:

Total personas	Total Hombres	Total Mujeres	Hora	domingo	jueves	martes	miercoles	sabado	viernes	Amigos	Estudiante	Parejas	Trabajador
1	0	1	8.1	0	0	1	0	0	0	0	0	0	1
1	0	1	8.116666666666667	0	0	1	0	0	0	0	0	0	1
1	0	1	8.15	0	0	1	0	0	0	0	0	0	1
1	1	0	8.166666666666666	0	0	1	0	0	0	0	0	0	1
1	1	0	8.483333333333333	0	0	1	0	0	0	0	0	0	1

Seguido de esto ocupamos PCA para reducirlo a variables las cuales sean más fáciles de manipular al modelo.

Se probaron diversos modelos:

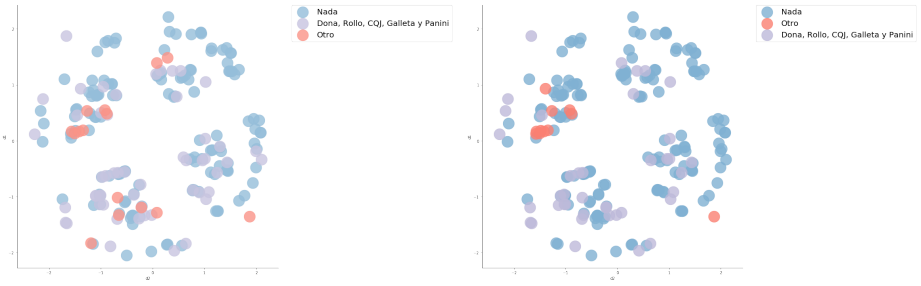
- **Red neuronal:** acc 72 %
- **XGBoost:** acc 72 %
- **Regresión Logística:** acc 72 %

- **SVC:** acc 74 %
- **Bosque Aleatorio:** acc 73 %
- **Extra Tree:** acc 74 %

Cada uno de los anteriores modelos fue hiperparametrizado, dado que el score no salia de ese intervalo, se propuso usar un VotingClassifier, el cual permitiera usar los modelos anteriores.

Voting Classifier: 83 %

Usando Reducción de dimensiones de esta tabla resulta de la siguiente manera:



(a) (b)

Figura 14: El izquierdo la Real y El lado derecho es el Prediccion.

9. Estrategia de uso

Después del análisis de las características de los clientes del establecimiento y la realización de modelos que nos permiten identificar patrones de compra, decidimos utilizar esta herramienta para personalizar nuestras promociones, esto se logró al observar que cuando una persona posee ciertas características tiene una mayor probabilidad de ordenar un producto específico, por ello resulta conveniente ofrecerle la mercancia por la cual el consumidor tiene una afinidad.

La razon por la cual funciona nuestra propuesta es que se basa en incrementar el gasto de los grupos de estudiantes y trabajadores, los cuales como ya se ha mencionado anteriormente visitan frecuentemente las instalaciones de Starbucks La Cúspide pero su aportación individual a las ganancias es mínima.

10. Conclusiones

Un buen manejo de los recursos supone el exito de una empresa, conocer al cliente es esencial, por ello nuestra investigacion aporta informacion necesaria para la toma de desiciones en un corporativo, mas aun nuestro producto brinda una oportunidad de crecimiento para cualquier empresa. Este trabajo resultó ser una idea muy rentable, ya que es una herramienta que ayuda a mejorar la productividad mediante diferentes tecnicas las cuales ayudan a la institucion a conocer a su clientela así como su comportamiento, y con base en ello poder hacer saber qué se necesita para ofrecerles lo que ellos quieren.

11. Referencias

[1] *Sebastian Raschka. (2016). Python Machine Learning. UK: PACKT.*

[2] *Tianqi Chen. (2016). XGBoost: A Scalable Tree Boosting System. 01-05-2018, de Cornell University Sitio web: https://arxiv.org/abs/1603.02754*

[3] *Navarro, J. B., i Massons, J. M. D., de la Osa, N., Ascaso, L. E. (1998). El análisis de curvas ROC en estudios epidemiológicos de psicopatología infantil: aplicación al cuestionario CBCL. Anuario de psicología/The UB Journal of psychology, 29(1), 3-16.*

[4] *Valenga, F., Britos, P. V., Perversi, I., Fernández, E., Merlino, H., García Martínez, R. (2007). Aplicación de Minería de Datos para la exploración y detección de patrones delictivos en Argentina. In XIII Congreso Argentino de Ciencias de la Computación.*

[5] *Huapaya, C. R., Lizarralde, F. Á. J., Arona, G., Massa, S. M. (2012). Minería de datos educacional en ambientes virtuales de aprendizaje. In XIV Workshop de Investigadores en Ciencias de la Computación.*

[6] *Valez Vázquez, Á., Areces Gayo, A., Blanco Pita, H., Arce Fernández, C. (2011). Diseño y aplicación de una batería multidimensional de indicadores de rendimiento para evaluar la prestación competitiva en el fútbol de alto nivel. RICYDE. Revista Internacional de Ciencias del Deporte, 7(23).*

[7] *Hernández, P. A. C. (2004). Aplicación de árboles de decisión en modelos de riesgo crediticio. Revista colombiana de estadística, 27(2), 139.*

[8] *Pérez, J., Henriques, M. F., Pazos, R., Cruz, L., Reyes, G., Salinas, J., Mexicano, A. (2007). Mejora al algoritmo de agrupamiento K-means mediante un nuevo criterio de convergencia y su aplicación a bases de datos poblacionales de cáncer. Liver-2do Taller Latino Iberoamericano de Investigacion de Operaciones "la IO aplicada a la solución de problemas regionales, 1-7.*

[9] *Andres Gonzalez, ¿Qué es Machine Learning?. En Big Data, Data prediction, Machine Learning.(2006) /Clever Data, http://cleverdata.io/que-es-machine-learning-big-data*

[10] *K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, segunda edición.*

[11] *Pattern Recognition: A Statistical Approach.Devijver, Kittler. Prentice Hall, Sin ISBN. (1982)*

[12] *Hartigan, J. A.; Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society, Series C (Applied Statistics) 28 (1): 100–108.*

[13] *Kanungo, T.; Mount, D. M.; [[Nathan Netanyahu—Netanyahu, N. S.]]; Piatko, C. D.; Silverman, R.; Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Analysis and Machine Intelligence 24: 881–892.*

[14] *A. Sarmiento, I. Fondón, M. Velasco, A. Qaisar P. Aguilera. Modelo de Mezcla de Gaussianas Generalizadas para Segmentación de Melanomas. Departamento de Teoría de la Señal y Comunicaciones, Universidad de Sevilla, Sevilla, España.*

[15] *S. Kotsiantis, Supervisado Aprendizaje Automático: Una Revisión de la Clasificación de las técnicas de Informática Diario 31 (2007) 249-268, http* : $//informatica.si/PDF/31-3/11_K otsiantis$

[16] *Lior Rokach and Oded Maimon (2008). Data mining with decision trees: theory and applications. World Scientific.*

[17] *Agresti, Alan. (2002). Categorical Data Analysis. New York: Wiley-Interscience.*

[18] *Hosmer, David W.; Stanley Lemeshow (2000). Applied Logistic Regression, 2nd ed. New York; Chichester, Wiley.*

[19] *Schölkopf, Bernhard; and Smola, Alexander J.; Learning with Kernels, MIT Press, Cambridge, MA, 2002.*

[20] *Bennett, Kristin P.; and Campbell, Colin; "Support Vector Machines: Hype or Hallelujah?", SIGKDD Explorations, 2, 2, 2000, 1–13.*

[21] *C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, USA, 1995.*

[22] *Dietterich, Thomas (2000). .An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization". Machine Learning: 139–157*

[23] *Mason, L.; Baxter, J.; Bartlett, P. L.; Frean, Marcus (May 1999). Boosting Algorithms as Gradient Descent in Function Space*

Model Total

May 14, 2018

In [1]:
```
from __future__ import division import pandas as pd
import numpy as np import matplotlib.pyplot as
plt pd.set_option("display.max_columns", 200)

%matplotlib inline
```

In [31]:
```
from sklearn.neural_network import MLPRegressor from sklearn.preprocessing import MinMaxScaler from
sklearn.ensemble import ExtraTreesRegressor, RandomForestRegressor,AdaBoostRegres from
sklearn.ensemble import RandomForestClassifier, GradientBoostingRegressor

from sklearn.model_selection import GridSearchCV, RandomizedSearchCV, train_test_split from
sklearn.metrics import r2_score, mean_absolute_error, accuracy_score, f1_score from xgboost import
XGBRegressor, XGBClassifier from sklearn.linear_model import LogisticRegression, logistic from
sklearn.svm import SVC, SVR from sklearn.naive_bayes import GaussianNB, BernoulliNB from
sklearn.manifold import MDS

from sklearn.preprocessing import LabelEncoder from sklearn.neighbors import
KNeighborsClassifier, KNeighborsRegressor import pickle
```

In [3]:
```
def rmv_extv(df, column,down=.1,up=.95):
    ext_95=df[(df[column]<=df[column].quantile(up))].reset_index(drop=True) return
    ext_95[(ext_95[column]>=ext_95[column].quantile(down))].reset_index(drop=Tru
```

In [4]:
```
df=pd.read_csv("Starbucks_Cluster_600_rempl.csv", index_col="Unnamed: 0")
#df=rmv_extv(df,"Total personas", down=.1).copy() df.shape
```

Out[4]: (617, 11)

In [5]:
```
df["Total personas"].value_counts()
```

Out[5]:
```
1     436
2     136
3      27
4       9
5       5
10      2
6       2
Name: Total personas, dtype: int64
```

```
In [6]: martes1=df.iloc[:77].reset_index(drop=True)

        miercoles1=df.iloc[77:129].reset_index(drop=True) jueves1=df.iloc[129:160].reset_index(drop=True)

        viernes1=df.iloc[160:209].reset_index(drop=True) sabado1=df.iloc[209:270].reset_index(drop=True)

        domingo1=df.iloc[270:320].reset_index(drop=True) martes2=df.iloc[320:410].reset_index(drop=True)

        miercoles2=df.iloc[410:493].reset_index(drop=True) jueves2=df.iloc[493:542].reset_index(drop=True)

        viernes2=df.iloc[542:].reset_index(drop=True)

In [7]: def creat(data,start,end, join=False): if join==False:
                    aux=pd.DataFrame([], columns=["X(%d)" %-x for x in np.arange(-start+1,0)]+["Anc for x in
                                                                    np.arange(1,end+1)])
                    a=data[["Total personas"]].T.copy()


                    for j in range(a.shape[1]-start-end+1):
                        va=a.iloc[:,j:start+end+j] va.columns=["X(%d)" %-x for x in np.arange(-

                start+1,0)]+["Ancla"]+["y(%s)" aux=pd.concat([aux,va]) return aux.reset_index(drop=True) else:

                    aux2=pd.DataFrame([], columns=["X(%d)" %-x for x in np.arange(-start+1,0)]+["Ancla"]+["y(%s
                                                            for x in np.arange(1,end+1)])
                    for d in data:
                        aux2=pd.concat([aux2,creat(d, start,end)]) return


                    aux2.reset_index(drop=True)

In [8]: def ninc(x): return sum([int(a<b) for a,b in zip(x,x[1:])])

In [9]: def delta(x): return np.mean([(b-a) for a,b in zip(x,x[1:])])

In [10]: df=creat([martes1,martes2]
                  , 10,5, join=True)

            for j in range(0,10,2):
                for k in ["mean", "sum", "max", "min", "std", ninc, ]:
                    if type(k)==str:
                        nombre=k
                    else:nombre=k.func_name


                    if j==0:
```

```
                    df["v_%s_%s" %(nombre, 10-j)]=df.loc[:,"X(%d)"%(9):"Ancla"].apply(k,axis= else:

                df["v_%s_%s" %(nombre, 10-j)]=df.loc[:,"X(%d)"%(10-j):"Ancla"].apply(k,a
```

In [11]:
```
Y=[x for x in df.columns if "y" in x] df[Y]=df[Y].replace(range(3,10), "3 o más")
```

In [12]:
```
df["target"]=df[Y].sum(axis=1) df=df[df["target"]<9].copy()

    lbl_1=LabelEncoder() lbl_1.fit(df["y(1)"]) for i in Y: df[i]=lbl_1.transform(df[i])
```

In [13]:
```
X=df[list(set(df.columns)-(set(Y)|{"target"}))] y=df["target"].copy()
```

In [14]:
```
y.value_counts()
```

Out[14]:
```
6.0     44
7.0     38
5.0     30
8.0     16
Name: target, dtype: int64
```

In [15]:
```
scx=MinMaxScaler()
scy=MinMaxScaler()
scx.fit(X) scy.fit(y)
Xm=pd.DataFrame(scx.transform(X), columns=X.columns) ym=pd.Series(scy.transform(y))
/home/comdisde/anaconda2/lib/python2.7/site-packages/sklearn/preprocessing/data.py:321: Deprecа
warnings.warn(DEPRECATION_MSG_1D, DeprecationWarning)
/home/comdisde/anaconda2/lib/python2.7/site-packages/sklearn/preprocessing/data.py:356: Deprecа
  warnings.warn(DEPRECATION_MSG_1D, DeprecationWarning)
```

In [33]:
```
df.shape
```

Out[33]:
```
(128, 46)
```

In [45]:
```
mdsx=MDS(n_components=46) mdsx.fit(Xm)

        Xmm=pd.DataFrame(mdsx.fit_transform(Xm), columns=["d%d" %x for x in range(46)]) In [47]:

Xt,Xv,yt,yv = train_test_split(Xmm,ym,train_size=0.7)
```

1 ML

In [49]:
```
model = XGBRegressor() model.fit(Xt,yt)

        print r2_score(y_pred=model.predict(Xt),y_true=yt) print
        r2_score(y_pred=model.predict(Xv),y_true=yv) print
```

```
mean_absolute_error(y_pred=model.predict(Xt),y_true=yt) print
mean_absolute_error(y_pred=model.predict(Xv),y_true=yv) """print
accuracy_score(y_pred=model.predict(Xt),y_true=yt) print
accuracy_score(y_pred=model.predict(Xv),y_true=yv) print
f1_score(average="weighted",y_pred=model.predict(Xt),y_true=yt) print
f1_score(average="weighted",y_pred=model.predict(Xv),y_true=yv)"""
```

```
0.989564615507
0.213177484863
0.0253118604756
0.207655483077
```

Out[49]: 'print accuracy_score(y_pred=model.predict(Xt),y_true=yt)\nprint accuracy_score(y_pred

1.1 Classifier

```
In [50]: param_grid_SVC=dict(C=np.arange(.5,2,.1),kernel=["rbf", "linear", "poly", "sigmoid"],
                                coef0=np.arange(.001,.011,.001), decision_function_shape=['ovo', '
                    probability=[True])
param_grid_KN=dict(n_neighbors=range(1,66), weights=['uniform', 'distance',], algorithm=['auto'],
                   leaf_size=[30], p=[2], metric=['minkowski'], metric_params=[None],)
param_grid_XGBC=dict(max_depth=range(15,30,3), n_estimators=range(2000,10000,1000),
                     learning_rate=np.arange(.001,.011,.001), booster=["gbtree", "gblinear
                     base_score=np.arange(.1,1.1,.1),
                         )
```

1.2 Regressor

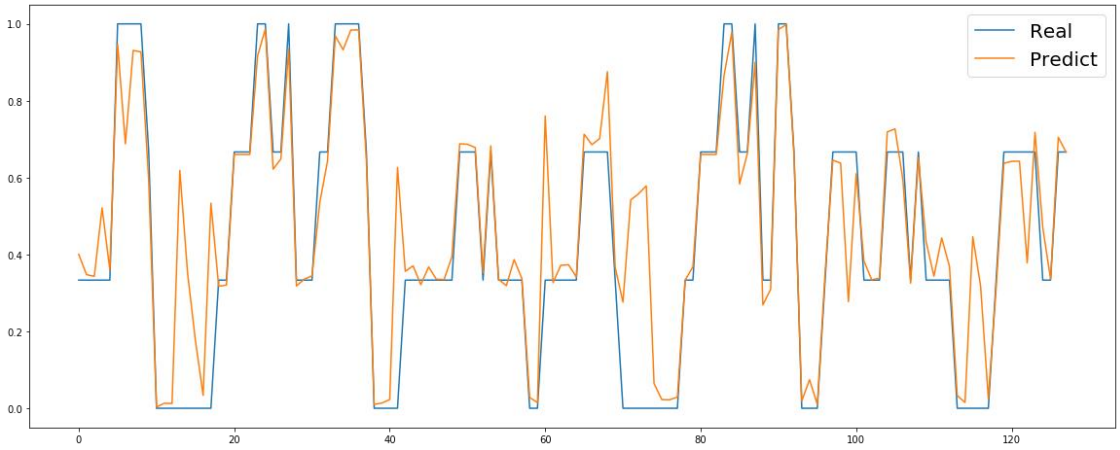
```
In [51]: param_grid_RF=dict(n_estimators=range(1000,10000,100), criterion=["mse", "mae"], bootstrap=[True],
                           min_samples_split=range(200,500,100),
                            min_samples_leaf=range(1000,5000,500),max_features=["auto", "sqrt" )
 param_grid_SVR=dict(C=np.arange(.5,2,.1),kernel=["rbf", "linear", "poly", "sigmoid"], coef0=np.arange(.001,.011,.001),
                                                                    decision_function_shape=['ovo', '
                    probability=[True])
param_grid_XGB=dict(n_estimators=range(60000,70000,1000), learning_rate=np.arange(.004,.005,.0001),
                    booster=["gbtree",], base_score=[.54]
                        , max_depth=range(60000,70000,1000), colsample_bytree=np.arange(.
                         objective=["reg:logistic", ],silent=[True],
                                   max_delta_step=range(10000,20000,1000), subsample=np.arange(.2,.3,

                          )

In [53]: grid= RandomizedSearchCV(model,scoring="r2", verbose=True
                                  , param_distributions=param_grid_XGB, cv=6, n_jobs=-1, n_iter=20
         grid.fit(Xt,yt)
```

```
Fitting 6 folds for each of 20 candidates, totalling 120 fits


[Parallel(n_jobs=-1)]: Done 42 tasks                                | elapsed:         53.1s
[Parallel(n_jobs=-1)]: Done 120 out of 120 | elapsed: 2.4min finished


Out[53]: RandomizedSearchCV(cv=6, error_score='raise', estimator=XGBRegressor(base_score=0.5, booster='gbtree',
                                colsample_bylevel=1
                    colsample_bytree=1,      gamma=0,      learning_rate=0.1,      max_delta_step=0,      max_depth=3,
                    min_child_weight=1,      missing=None,      n_estimators=100,      n_jobs=1,      nthread=None,
                    objective='reg:linear',      random_state=0,      reg_alpha=0,      reg_lambda=1,      scale_pos_weight=1,
                    seed=None,      silent=True,      subsample=1),      fit_params={},      iid=True,      n_iter=20,      n_jobs=-1,
                    param_distributions={'colsample_bytree': array([ 0.3 , 0.31, 0.32, 0.33,
                                                                     0.39, 0.4 ]), 'silent': [True], 'learning_rate': array([ 0.004 , 0.0041, 0
                     0.0047, 0.0048, 0.0049]), 'max_delta_step': [10000,...pth': [60000, 61000, 6 pre_dispatch='2*n_jobs',
                                random_state=None, refit=True, return_train_score=True, scoring='r2', verbose=True)

In [54]: print grid.best_params_
                    print grid.best_estimator_
                    print grid.score(X=Xt,y=yt) print
                    grid.score(X=Xv,y=yv)

{'colsample_bytree': 0.29999999999999999, 'silent': True, 'learning_rate': 0.004100000000000000
XGBRegressor(base_score=0.54, booster='gbtree', colsample_bylevel=1,
                    colsample_bytree=0.29999999999999999, gamma=0,
                    learning_rate=0.0041000000000000003, max_delta_step=11000, max_depth=66000,
                    min_child_weight=1, missing=None, n_estimators=68000, n_jobs=1, nthread=None,
                    objective='reg:logistic', random_state=0, reg_alpha=0, reg_lambda=1,
                    scale_pos_weight=1, seed=None, silent=True, subsample=0.20000000000000001)
0.990037194254
-0.200033472368


    model=grid.best_estimator_ print     accuracy_score(y_pred=model.predict(Xt),y_true=yt) print
accuracy_score(y_pred=model.predict(Xv),y_true=yv)  print
f1_score(average="weighted",y_pred=model.predict(Xt),y_true=yt)       print
f1_score(average="weighted",y_pred=model.predict(Xv),y_true=yv)

In [22]: model=grid.best_estimator_ print r2_score(y_pred=model.predict(Xt),y_true=yt)
                    print r2_score(y_pred=model.predict(Xv),y_true=yv) print
                    mean_absolute_error(y_pred=model.predict(Xt),y_true=yt) print
                    mean_absolute_error(y_pred=model.predict(Xv),y_true=yv)

0.801196116531
0.268894641636
0.109251373039
```

0.185239610931

In [56]:
```
fig, ax = plt.subplots(figsize=(20, 8))

    fig=ym.astype(float).reset_index(drop=True).plot(ax=ax,label="Real")
    pd.Series(model.predict(Xmm)).plot(ax=ax, label="Predict") plt.legend(fontsize=20)
```

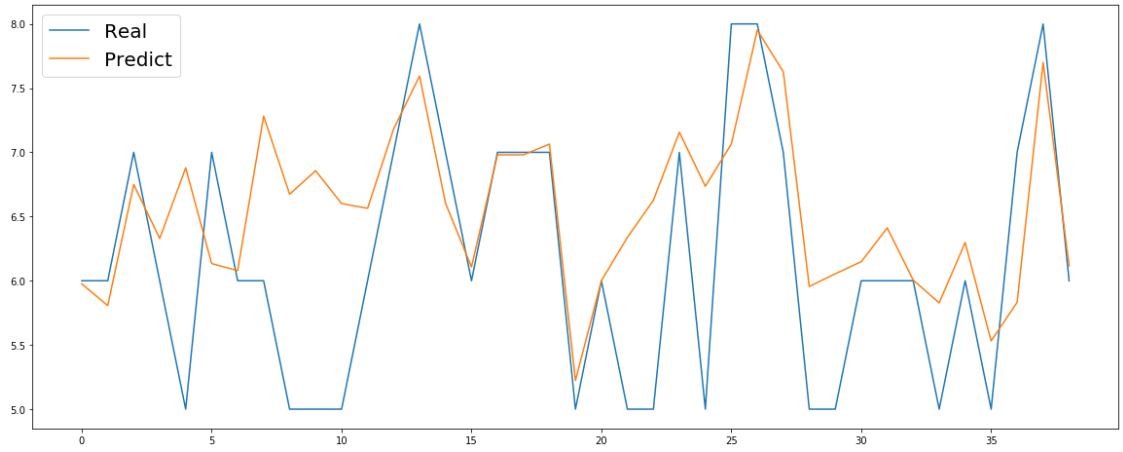
Out[56]: <matplotlib.legend.Legend at 0x7f3dab453610>



In [57]:
```
fig, ax = plt.subplots(figsize=(20, 8))

    fig=pd.Series(scy.inverse_transform(yv)).plot(ax=ax, label="Real")
    pd.Series(scy.inverse_transform(model.predict(Xv))).plot(ax=ax, label="Predict")


    plt.legend(fontsize=20)
```

/home/comdisde/anaconda2/lib/python2.7/site-packages/sklearn/preprocessing/data.py:374: Deprecawarnings.warn(DEPRECATION_MSG_1D, DeprecationWarning)
/home/comdisde/anaconda2/lib/python2.7/site-packages/sklearn/preprocessing/data.py:374: Deprecawarnings.warn(DEPRECATION_MSG_1D, DeprecationWarning)

Out[57]: <matplotlib.legend.Legend at 0x7f3dab52a450>



pickle.dump(model, open("Clientes 18%", "wb"))

Modelo comida

May 14, 2018

In [88]:
```
from __future__ import division import pandas as pd
import numpy as np import matplotlib.pyplot as
plt import seaborn as sns
pd.set_option("display.max_columns", 200)

%matplotlib inline
```

In [151]:
```
from sklearn.neural_network import MLPClassifier from sklearn.preprocessing import MinMaxScaler,
OneHotEncoder, LabelEncoder, Standard from sklearn.ensemble import AdaBoostClassifier,
RandomForestClassifier, VotingClassi from sklearn.model_selection import GridSearchCV,
RandomizedSearchCV, train_test_spli from sklearn.metrics import r2_score, mean_absolute_error,
accuracy_score, f1_score from sklearn.linear_model import LogisticRegression from
sklearn.decomposition import PCA from sklearn.svm import SVC from xgboost import XGBClassifier
from sklearn.manifold import MDS


from Tools import frecuencia, metricas import pickle
```

In [52]:
```
df=pd.read_csv("Starbucks_Cluster_600_rempl.csv", index_col="Unnamed: 0") df.Bebida.replace(["Cold ","0"],
["Cold", np.nan]
                                , inplace=True)
df.Bebida.replace(["Jugo","CJQ", "Te"], np.nan, inplace=True) df["Local/Llegar"].replace("Local
","Local", inplace=True) df.dropna(inplace=True)
```

In [54]:
```
frecuencia(df, "Bebida")
```

Out[54]:

	Bebida	Conteo	Frecuencia
0	Hot	387	0.728814
1	Cold	80	0.150659
2	Frappe	64	0.120527

In [177]:
```
lbl=LabelEncoder() lbl.fit(df.Bebida)
df.Bebida=lbl.transform(df.Bebida)
```

In [57]:
```
df.Hora=pd.to_timedelta(df.Hora.map(lambda x: x+":00")).dt.seconds/60**2
```

In [59]:
```
df.head()
```

Out[59]:

	Dinero gastado	Total personas	Total Hombres	Total Mujeres	Hora \

```
0                    25     1     0     1 8.100000
1                    29     1     0     1 8.116667
2                    48     1     0     1 8.150000
3                     0     1     1     0 8.166667
6                    25                 1                 1                 0 8.483333

      Edad Local/Llegar Bebida Comida      Dia      cl 0 20-30
     Local   2        0 martes Trabajador
1    40-50   Local    2 Rollo martes Trabajador
2    20-30   Llevar   2        0 martes Trabajador
3    40-50   Local    1        0 martes Trabajador
 6 40-50             Llevar            2              0 martes Trabajador
```

In [61]:
```
df=pd.concat([df,pd.get_dummies(df["Local/Llegar"]), pd.get_dummies(df.cl), pd.get_dummies(df.Dia)],
                                axis=1).copy()
    #df.Dia=df.Dia.map({"martes":2, "miercoles":3, "jueves":4,"viernes":5, "sabado":6, "do
```

In [62]:
```
var=["Total personas", "Total Hombres", "Total Mujeres", "Hora",
                                "domingo", "jueves" , "martes" , "miercoles", "sabado","viernes",
                    "Amigos", "Estudiante" ,"Parejas","Trabajador"]
```

In [81]:
```
X=df[var].copy() y=df["Bebida"].reset_index(drop=True)
```

In [78]:
```
std=StandardScaler() std.fit(X)
        Xs=pd.DataFrame(std.transform(X), columns=X.columns)
```

In [157]:
```
n=10 pca=PCA(n_components=n)
        pca.fit(Xs)

        Xp=pd.DataFrame(pca.transform(Xs), columns=["p%s" %s for s in range(1,n+1)])
        pca.explained_variance_ratio_.sum()
```

Out[157]: 0.9837905013426772

In [150]:
```
scx=MinMaxScaler()
        scy=MinMaxScaler() scx.fit(X)
        scy.fit(y.reshape(-1,1))
        Xm=pd.DataFrame(scx.transform(X), columns=X.columns) ym=pd.DataFrame(scy.transform(y.reshape(-
        1,1))) C:\Users\Comdisde\Anaconda2\lib\site-packages\ipykernel_launcher.py:4: FutureWarning: reshape i
        after removing the cwd from sys.path.
C:\Users\Comdisde\Anaconda2\lib\site-packages\ipykernel_launcher.py:6: FutureWarning: reshape i
```

In [67]:
```
Xt, Xv, yt, yv= train_test_split(Xp,y, test_size=.3)
```

1 ML

```
In [44]: model=XGBClassifier() model.fit(Xt,
             yt) print model.score(X=Xt,y=yt)
             print model.score(X=Xv,y=yv)
```

```
0.894878706199461
0.78125
```

```
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:
```

```
In [170]: param_grid_XGB=dict(max_depth=range(15,30,3), n_estimators=range(1000,2000,100),
                                learning_rate=np.arange(.001,.011,.001), booster=["gbtree", "gblinear
                                base_score=np.arange(.1,1.1,.1)
                                    )

          param_grid_RF=dict(n_estimators=range(1000,10000,100), criterion=["gini", "entropy" bootstrap=[True],
                                min_samples_split=range(200,500,100),
                                 min_samples_leaf=range(1000,5000,500),max_features=["auto", "sqrt )

          param_grid_RED=dict(hidden_layer_sizes=[(x,y,z) for x in range(100,1000,100) for y in
                                                          range(100,1000,100) for z in range(
                                 activation=['identity', 'logistic', 'tanh', 'relu'], solver=["lbfgs", "sgd","adam"],
                                 learning_rate_init=np.arange(.0001 learning_rate=['constant', 'invscaling', 'adaptive'],
                                 max_iter=[20

          param_grid_AD=dict(n_estimators=range(1000,10000,100),)


          param_grid_svc=dict(C=np.arange(.5,2,.1),kernel=["rbf", "linear", "poly", "sigmoid"]
                                 coef0=np.arange(.001,.011,.001), decision_function_shape=['ovo',
                                  probability=[True])
          grid= GridSearchCV(model,scoring="accuracy", verbose=True
                                   , param_grid=param_grid_svc, cv=8, n_jobs=-1)




          grid.fit(Xt, yt)
```

```
Fitting 8 folds for each of 16200 candidates, totalling 129600 fits

[Parallel(n_jobs=-1)]: Done 472 tasks                | elapsed:         3.7s
[Parallel(n_jobs=-1)]: Done 2872 tasks               | elapsed:        20.8s
```

```
[Parallel(n_jobs=-1)]: Done 6872 tasks                  | elapsed:           49.3s
[Parallel(n_jobs=-1)]: Done 12472 tasks                 | elapsed: 1.7min
[Parallel(n_jobs=-1)]: Done 19672 tasks                 | elapsed: 2.7min
[Parallel(n_jobs=-1)]: Done 28472 tasks                 | elapsed: 3.8min
[Parallel(n_jobs=-1)]: Done 38872 tasks                 | elapsed: 5.1min
[Parallel(n_jobs=-1)]: Done 50872 tasks                 | elapsed: 6.7min
[Parallel(n_jobs=-1)]: Done 64472 tasks                 | elapsed: 8.5min
[Parallel(n_jobs=-1)]: Done 79672 tasks                 | elapsed: 10.6min
[Parallel(n_jobs=-1)]: Done 96472 tasks                 | elapsed: 12.9min
[Parallel(n_jobs=-1)]: Done 114872 tasks                | elapsed: 15.6min
[Parallel(n_jobs=-1)]: Done 129600 out of 129600 | elapsed: 17.7min finished


Out[170]: GridSearchCV(cv=8, error_score='raise', estimator=SVC(C=1.0, cache_size=200,
                  class_weight=None, coef0=0.0,
                decision_function_shape=None, degree=3, gamma='auto', kernel='rbf', max_iter=-1, probability=False,
                random_state=None, shrinking=True, tol=0.001, verbose=False), fit_params={}, iid=True, n_jobs=-1,
                param_grid={'kernel': ['rbf', 'linear', 'poly', 'sigmoid'], 'C': array([ 0.5,
                                                        1.6, 1.7, 1.8, 1.9]), 'degree': [1, 2, 3, 4, 5, 6, 7, 8, 9], 'probability
                  0.009, 0.01 ]), 'decision_function_shape': ['ovo', 'ovr', None]}, pre_dispatch='2*n_jobs', refit=True,
                return_train_score=True, scoring='accuracy', verbose=True)

In [175]: modelhy=grid.best_estimator_ print accuracy_score(modelhy.predict(Xt), yt) print
          accuracy_score(modelhy.predict(Xv), yv) print f1_score(average='weighted',
          y_pred=modelhy.predict(Xt), y_true=yt) print f1_score(average='weighted',
          y_pred=modelhy.predict(Xv), y_true=yv)

0.787061994609
0.725
0.737891133355
0.66292759324
In [97]: a=pickle.load(open("72% bebidas MLP get dummies", "rb"))

In [98]: b=pickle.load(open("72% bebidas Log get dummies", "rb")) In [99]:

c=pickle.load(open("73% bebidas RF get dummies", "rb"))

In [102]: d=pickle.load(open("73% bebidas vc get dummies", "rb"))

In [118]: e=pickle.load(open("72% bebidas EXT get dummies", "rb"))

In [185]: f=pickle.load(open("76% bebidas svc get dummies", "rb"))

In [68]: g=pickle.load(open("75% bebidas vc get dummies", "rb"))

In [187]: vc = VotingClassifier(estimators=[('RED', a), ("Log", b), ("RF", c),
                                            ("vc",d), ("svc", f), ("XGB",modelhy), ("vc2",g)],
```

```
                                voting='soft')

        vc.fit(Xt,yt)


        print accuracy_score(vc.predict(Xt), yt) print accuracy_score(vc.predict(Xv), yv)
        print f1_score(average='weighted', y_pred=vc.predict(Xt), y_true=yt) print
        f1_score(average='weighted', y_pred=vc.predict(Xv), y_true=yv)
```

```
0.757412398922
0.725
0.678351142467
0.631266229329
```

```
In [69]: print accuracy_score(g.predict(Xt), yt)
        print accuracy_score(g.predict(Xv), yv) print f1_score(average='weighted',
        y_pred=g.predict(Xt), y_true=yt) print f1_score(average='weighted',
        y_pred=g.predict(Xv), y_true=yv)
```

```
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:
```

```
0.6954177897574124
```

```
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:
```

```
0.775
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:
```

```
0.5968236239257007
0.6989316239316239
```

```
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:
```

```
In [193]: pickle.dump(pca,open("PCA get dummies", "wb"))

In [165]: mds=MDS(n_components=2)
        Xmm=pd.DataFrame(mds.fit_transform(Xm), columns=["d1","d2"])

In [180]: Xmm["Predict"]=lbl.inverse_transform(g.predict(Xp))
```

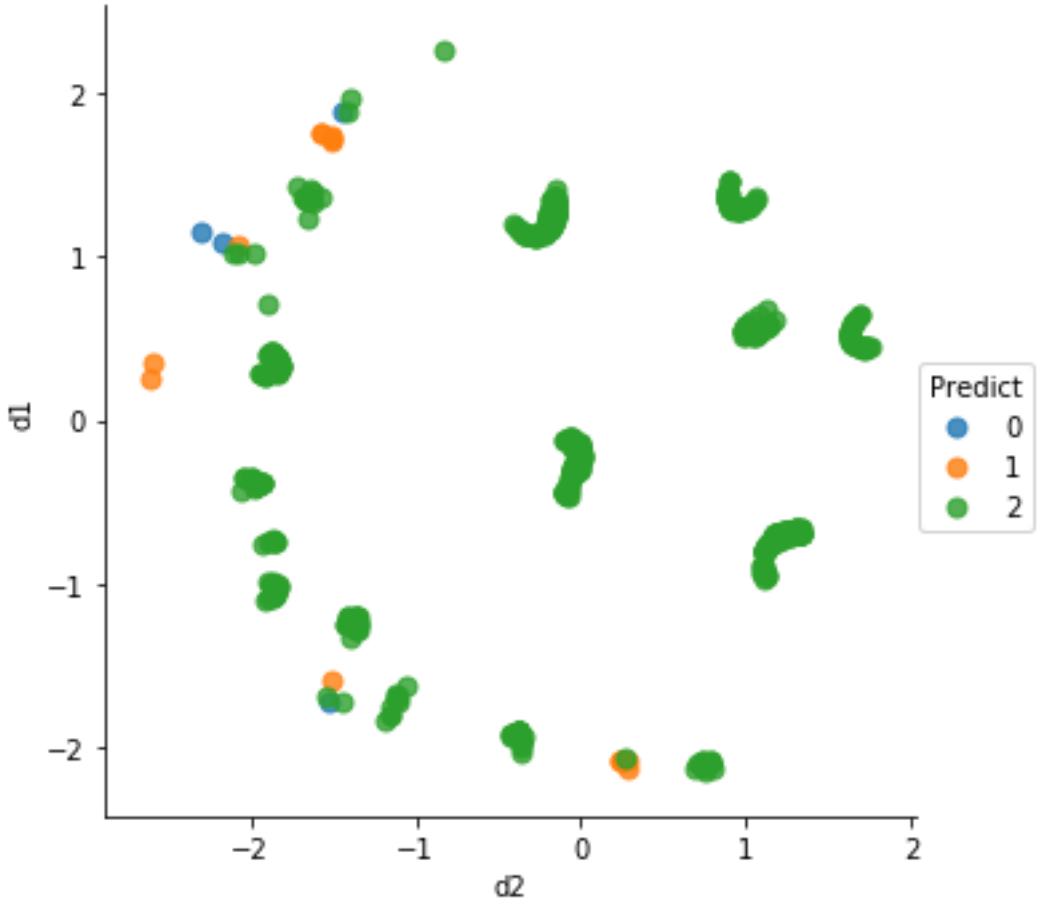
```
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:
```

C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:

In [182]:
```
sns.lmplot(fit_reg=False,data=pd.concat([Xmm, pd.Series(lbl.inverse_transform(y))]
                                                                    , axis=1), x="d2",y="d1",hue="Predict", size
                    scatter_kws={'s':50})
```

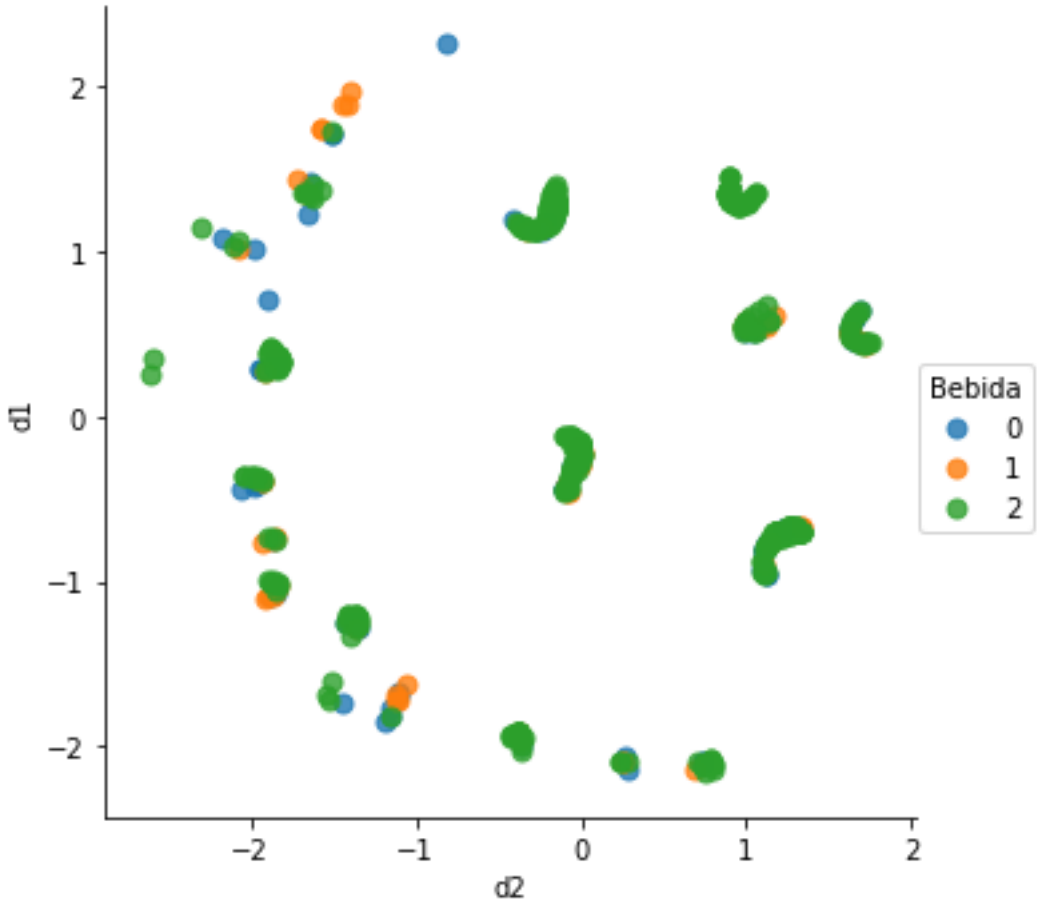
C:\Users\Comdisde\Anaconda2\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWa if diff:

Out[182]: <seaborn.axisgrid.FacetGrid at 0x138a4c50>



In [183]:
```
sns.lmplot(fit_reg=False,data=pd.concat([Xmm, y], axis=1), x="d2",y="d1",hue="Bebida scatter_kws={'s':50})
```

Out[183]: <seaborn.axisgrid.FacetGrid at 0x1ec9a7b8>


d1
d2
Bebida
0
1
2