

Programación de horarios de centros de llamadas utilizando máquinas de aprendizaje

Schedule Planning Software for Call Center Using Machine Learning

Fecha de recepción: 2024-02-23 • Fecha de aceptación: 2024-03-25 • Fecha de publicación: 2024-04-19

Luis Aguirre

Instituto Superior Tecnológico San Antonio TESA, Quito, Ecuador
ORCID 0009-0002-2898

Resumen

Las máquinas de aprendizaje facilitan el análisis de datos para la obtención de varios objetivos tanto de calidad de servicio, como eficiencia de recursos a nivel empresarial. En este proyecto evaluamos diversos algoritmos de máquinas de aprendizaje que facilitan la predicción de llamadas y posterior programación de horarios de un call center con el fin de mejorar su nivel de servicio. Basado en el histórico de llamadas y mediante la fórmula de Erlang-C se estiman los agentes necesarios por hora para un nivel de servicio deseado. Se desarrolla un Backend en Python y se integra a la aplicación web usando PHP. Se comprobó la eficacia de los algoritmos de máquinas de aprendizaje para estas predicciones, así como su utilidad para la programación de horarios. El nivel de servicio del call-center que se obtuvo fue superior a los niveles esperados gracias a una estimación de recursos humanos adecuada.

Palabras clave:

Máquinas de aprendizaje, centro de llamadas, planificación de horarios.

Abstract:

Machine learning algorithms ease data analysis to obtain several objectives both of service quality and resource efficiency at the enterprise level. In this project we evaluate various algorithms of machine learning that facilitate the prediction of calls and subsequent

scheduling of a call center in order to improve its level of service. Based on the call history and using the Erlang-C formula, the necessary agents per hour are estimated for a desired level of service. A Backend is developed in Python and it is integrated into the web application using PHP. The effectiveness of machine learning algorithms for these predictions, as well as their usefulness for scheduling, was tested. The level of call-center service obtained was higher than expected levels thanks to an adequate estimate of human resources.

Keywords:

Machine learning, call-center, schedule planning.

Introducción

En el presente trabajo se hace un análisis de las técnicas y metodologías de inteligencia artificial, para predecir un factor de operación de un call center de atención, como lo es el arribo de llamadas, y con esta métrica estimar los recursos necesarios para alcanzar el nivel de servicio objetivo de la empresa, es decir, dimensionar de la manera óptima el personal necesario para un call center.

Los centros de llamadas y procesamiento de información o call center, se han convertido en una herramienta de vital importancia para las empresas, ya que son una de las áreas responsables de la atención al cliente y por ende una de las áreas de las que depende la percepción del cliente acerca de la empresa, lo cual implica que del call center depende la satisfacción del cliente. Adicionalmente a nivel global, y a pesar a de la pandemia del COVID-19, la industria de los call center se estima es un mercado de 339.4 mil millones de dólares en el año 2020, y se prevé que alcanzará los 496 mil millones de dólares para el año 2027 (Report Linker, 2021).

En este trabajo se toman datos históricos del arribo de llamadas de un call center, para ejecutar una posterior predicción usando máquinas de aprendizaje. Se comparan varios

algoritmos de regresiones y evaluando diversas métricas se escoge el óptimo de acuerdo los objetivos del centro de llamadas. El objetivo es alcanzar un nivel de servicio de 80% de llamadas contestadas en menos de 20 segundos. Usando la fórmula de Erlang-C para calcular cuánto recurso humano será necesario para atender dicho call center y alcanzar un nivel de servicio deseado se muestran los resultados en un software desarrollado para este fin.

Materiales y métodos

Existen varios softwares que permiten programación de horarios, aunque no todos permiten predecir como tal los recursos para dichos horarios, sino que son softwares de tipo administrativo para gestión de horarios.

“Deputy” (Deputechologies Pty Ltd., 2021) es un software especializado en el manejo de turnos y horarios para muchos tipos de negocios, entre los que se indica call center y promocionan como una de sus características el poder crear horarios automáticos usando inteligencia artificial y poder comparar las predicciones de demanda con la data real para ajustar horarios. Es una solución que parece bastante completa y que promociona ser usada por varias empresas como Uber, Amazon, entre otros.

“TixTime” es un software de gestión de horarios que también tiene software de horarios dedico a call center (TixTime, Inc, 2021) y que entre sus características publicita poder combinar el pronóstico del call center con el manejo de horarios, más no se ve a detalle cómo se integra con dicho pronóstico y no se indica que el

pronóstico lo realice como tal el software. No muestra o promociona más que solamente dos clientes, Healthtek Inc., y The American Genius.

“Shiftboard” es un software de horarios con una solución para call center (Shiftboard, Inc, 2021) donde si ofrece tener una opción de pronosticar la demanda de personal basado en patrones del volumen de llamadas. Aunque no se indica exactamente si es mediante una integración con los softwares de call center o mediante otra estrategia como subir archivos, pero es uno de los que aparentemente si tiene lo que se busca hacer en el presente trabajo

“Genesys” es una plataforma de call center que ofrece entre sus características hacer pronósticos de llamadas y planificar los agentes necesarios en base a dicho pronóstico (Genesys, 2021), y al ser una solución de contact center completa se puede asumir que esta si presenta una característica integrada por completo al software de llamadas y que puede hacer lo que se plantea con el software en el presente trabajo.

“Verint Monet” (Verint, 2021) es un software de gestión de la fuerza laboral de contact centers, y que ofrece entre sus funcionalidades el pronóstico de llamadas y la gestión de horarios, en este caso habla de que tiene integración con varios softwares de ACD (automatic call distribution) es decir software de call center, ya que no es un software de call center como en el caso de Genesys, y sería una opción que entonces presenta una solución como la que se plantea en el presente trabajo.

Todos estos softwares, son de tipo comercial, y los costos son muy variables y en diferentes modelos de negocio, ya que varios son servicios en la nube y se cobra por usuario o por grupo de usuarios, pero por ejemplo Genesys al ser una solución completa de call center, ofrece implementación en sitio y los costos son muy elevados debido a que se ofrece toda la solución.

El software del presente trabajo aporta el hecho de que el mecanismo de predicción de arribo de llamadas, basado en un algoritmo de regresión, se escoge comparando varios algoritmos con los datos del call center y buscando el que menor raíz del error cuadrático promedio generé, y no se usan mecanismos tradicionales como ARIMA o Holt-Winters que son los que usan normalmente los sistemas mencionados.

No se creará un modelo basado en un algoritmo, que trate de funcionar para todos los call center, sino que se dará el mecanismo para crear el modelo de el algoritmo que mejor se ajuste a cada call center, y que se puede generar bajo demanda por el usuario de este software.

Esto se considera lo más práctico ya que cada call center, incluso siendo del mismo tipo o mismo giro de negocio, tendrá un flujo de llamadas propio y que se debe modelar con sus propios datos.

Para el desarrollo de software en cuestión, se sigue una metodología secuencial de pasos, donde cada paso está analizado en base a unos requerimientos funcionales, y así se puede completar el prototipo de software deseado.

- Se creará un script en Python para realizar las tareas de machine learning para la predicción del arribo de llamadas, esto incluye, preparación de los datos, basándose en un formato de archivo de datos predefinido; comparación de los diferentes algoritmos de regresiones; creación del modelo basado en el mejor resultado obtenido; predicción de arribo de llamadas en un rango de fechas a futuro deseado; aplicación de la fórmula de Erlang-C para obtener los recursos humanos necesarios.
- Luego se creará una interfaz web, que será la interfaz de usuario, que permita acceder a las funcionalidades requeridas como son, subir la data; ingresar parámetros de tiempos promedio de atención, nivel de servicio, rango de fechas para horarios requerido; tener botones de acción para las funcionalidades requeridas.
- Se integrará la interfaz de usuario, con el script de funcionalidades para tener la aplicación en su totalidad.
- Se realizará pruebas de evaluación con al menos dos data set
- Finalmente se dará acceso a la aplicación a un administrador de call center para validación y evaluación de la misma.

Resultados y discusión

El dimensionar y programar horarios de un call center implica tener un estimado de cuantas llamadas se recibirán, algo que en un principio se puede estimar sin una regla general más que intuición o una aproximación, pero mientras más data se vaya obteniendo se puede hacer una mejor estimación o, mejor aún, una predicción del arribo de llamadas, que junto con la duración promedio de llamadas, permitirá tener, probabilísticamente, una programación de horario para un nivel de servicio objetivo.

Por ello, todo este proceso se realizará dentro de un software que aplique algoritmos de inteligencia artificial para hacer la predicción del arribo de llamadas y permita hacer los

horarios óptimos, de forma que se tenga una mejora en el nivel de servicio que beneficie a la empresa y al usuario final.

Entonces se contribuye con un mecanismo, mediante un software, que permite comparar varios algoritmos de regresiones para cada conjunto de datos de cualquier call center, y así elegir el mejor algoritmo para programar el horario de acuerdo con un nivel de servicio deseado, usando las fórmulas de Erlang-C.

Para la evaluación del software del presente trabajo, se va a realizar primero una evaluación de los algoritmos de regresión utilizados, usando dos conjuntos de datos muy distintos entre sí, determinando que algoritmo o tipo de algoritmo resulta mejor en ambos casos.

Posteriormente se realiza la evaluación del software como tal, por parte de tres expertos con amplia experiencia, uno experto en gestión y administración de callcenter, el segundo como desarrollador de software de callcenter y el tercero como comercializador de software de callcenter, todos con al menos 10 años de experiencia en sus respectivos campos.

Evaluación de algoritmos de regresión

Para evaluar algoritmos de regresión existen varias métricas de error, que, como regla general, se selecciona una y se aplica a todos los modelos de la misma forma y se elige el que menor valor de error resulte.

PyCaret (Ali, 2020), por defecto en caso de regresiones va a crear un modelo de cada algoritmo usando hiperparámetros por defecto y los compara usando las siguientes métricas de evaluación de regresiones:

- El error absoluto medio (MAE o mean absolute error) (Kampakis, 2020) simplemente calcula el promedio del valor absoluto del error, es decir, se obtiene el valor absoluto de cada error y se calcula el promedio para los N valores de test.



- El error cuadrático medio (MSE o mean squared error) calcula el valor promedio del error obtenido al cuadrado, es decir, calcula el error, se eleva al cuadrado y se obtiene el promedio de los N valores de test (scikit-learn developers, 2021).
- La raíz del error cuadrático medio (RMSE o root mean squared error) (Kampakis, 2020) es la raíz cuadrada del MSE, es decir, se calcula el promedio de los errores al cuadrado para los N valores de test y finalmente se obtiene la raíz cuadrada de este promedio. Esta métrica junto con el MSE, se caracteriza penalizar errores grandes.
- El logaritmo de la raíz del error cuadrático medio (RMSLE o Root Mean Squared Log Error) se calcula como la raíz cuadrada del promedio de la sumatoria de los cuadrados de la resta entre el logaritmo del valor real más uno, menos el logaritmo del valor predicho más uno (scikit-learn developers, 2021).
- El error porcentual absoluto medio (MAPE o Mean Absolute Percentage Error) es el promedio de las sumatorias de los errores absolutos divididos para el valor real, y que se puede convertir en porcentaje (scikit-learn developers, 2021).
- R² o coeficiente de determinación, se calcula restando uno menos, la sumatoria de los errores al cuadrado dividido para la sumatoria de las diferencias entre el valor real menos el promedio de los valores reales al cuadrado. Esta métrica determina la capacidad de predecir valores de un modelo, siendo 1 el valor ideal que indica que predice todo con éxito, es decir a diferencia de las métricas anteriores, mientras mayor sea/, indica que el modelo es mejor (scikit-learn developers, 2021).

De todos estos se va a usar el RMSE como métrica para determinar el mejor modelo que usará el software para predecir el arribo de llamadas y programar un horario.

Para la evaluación se va a utilizar dos conjuntos de datos, de dos call center totalmente distintos. El conjunto uno es las llamadas de un call center de atención de una empresa de medicina prepagada, que atiende 24/7, es decir está disponible todo el tiempo, todos los días del año. Este conjunto son el histórico de llamadas desde el 1 de enero del 2016 hasta el 28 de febrero del 2019.

El segundo conjunto de datos es de un call center, relativamente nuevo, que brinda atención a los clientes de una operadora de telefonía móvil, pero no es directamente la operadora de telefonía, sino que es un call center de terceros que brinda el servicio a la operadora. Este call center solo atiende de 8 a 19 horas de lunes a viernes y el sábado de 9 a 15 horas, por lo que datos hay solo en esas franjas horarias, y se cuenta con datos desde el 8 de abril de 2021, cuando empezó a funcionar este call center, hasta el 31 de mayo de 2021.

Como se nota, los dos conjuntos de datos son completamente diferentes, tanto en tamaño, donde el conjunto uno tiene 671925 filas de datos mientras el conjunto dos tiene 11089 filas, así como en el tipo de call center el uno atiende peticiones de médicos a domicilio, ambulancias, reembolsos, consultas y quejas que son de una aseguradora médica, mientras que el segundo call center atiende consultas y quejas de una operadora de telefonía móvil, y sobre todo en las franjas horarias que manejan siendo un call center 24/7 el uno y el dos solo cierto horario de lunes a sábado.

En la **Figura 1** y **Figura 2**, se puede observar el flujo de llamadas por hora, de los últimos treinta días, de cada conjunto de datos:

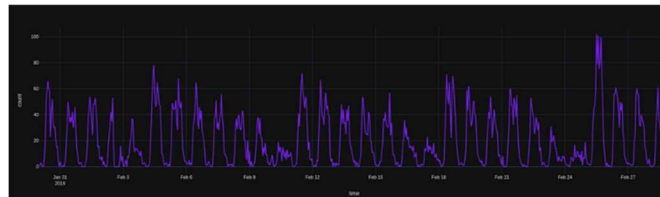


Fig 1. Flujo de llamadas de call center 1. (Fuente propia)

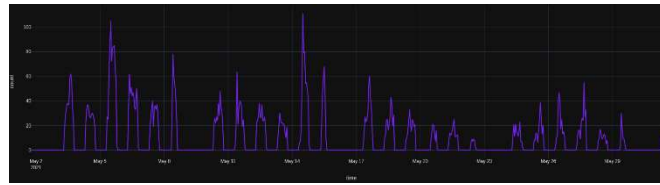


Fig 2. Flujo de llamadas de call center 2. (Fuente propia)

En las figuras se puede notar la diferencia entre ambos datos sobre todo en que el conjunto de datos del call center 1 no tiene espacios tan grandes con 0 llamadas, ya que como se mencionó el primer call center atiende 24/7 mientras el segundo solo en cierto horario.

En las **Figuras 3 y 4** se muestra los resultados de comparar todos los modelos de regresión mencionados en el presente trabajo, ordenado por el RMSE de menor a mayor, para cada conjunto de datos de call center.

Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	
lightgbm	Light Gradient Boosting Machine	8.799100e+00	2.519403e+02	1.515990e+01	6.422000e-01	0.5586	6.002000e-01
gbr	Gradient Boosting Regressor	9.667300e+00	2.489460e+02	1.525700e+01	6.215000e-01	0.6764	8.656000e-01
catboost	CatBoost Regressor	9.076500e+00	2.506519e+02	1.526790e+01	6.308000e-01	0.6158	6.951000e-01
rf	Random Forest Regressor	8.980100e+00	2.678074e+02	1.564130e+01	6.207000e-01	0.5626	6.055000e-01
et	Extra Trees Regressor	9.206700e+00	2.808147e+02	1.605320e+01	5.781000e-01	0.5895	6.334000e-01
xgboost	Extreme Gradient Boosting	9.688600e+00	2.852718e+02	1.636760e+01	5.807000e-01	0.6354	7.189000e-01
dt	Decision Tree Regressor	1.048790e+01	3.683046e+02	1.858430e+01	4.711000e-01	0.6868	7.243000e-01
ard	Automatic Relevance Determination	1.469420e+01	3.657371e+02	1.879840e+01	4.744000e-01	1.1960	2.530300e+00
br	Bayesian Ridge	1.481460e+01	3.702463e+02	1.891780e+01	4.687000e-01	1.2026	2.559200e+00
ridge	Ridge Regression	1.481820e+01	3.703406e+02	1.892050e+01	4.685000e-01	1.2028	2.559900e+00
mlp	MLP Regressor	1.310840e+01	3.657898e+02	1.895630e+01	3.889000e-01	0.9210	1.557500e+00
knn	K Neighbors Regressor	1.234580e+01	4.201956e+02	2.002720e+01	2.125000e-01	0.6728	8.943000e-01
huber	Huber Regressor	1.798960e+01	5.807544e+02	2.330120e+01	2.965000e-01	1.3481	3.325500e+00
lasso	Lasso Regression	1.836640e+01	5.814291e+02	2.334210e+01	2.929000e-01	1.4072	3.487400e+00
ada	AdaBoost Regressor	1.902420e+01	5.861951e+02	2.389140e+01	-4.920000e-02	1.2741	2.901500e+00
omp	Orthogonal Matching Pursuit	1.908840e+01	6.105331e+02	2.433960e+01	1.720000e-01	1.4562	3.515200e+00
ransac	Random Sample Consensus	1.713420e+01	6.247280e+02	2.461430e+01	8.310000e-02	1.1630	2.393900e+00
en	Elastic Net	2.121680e+01	7.491241e+02	2.648740e+01	9.420000e-02	1.5471	4.268200e+00
llar	Lasso Least Angle Regression	2.481470e+01	9.175560e+02	2.934720e+01	-1.139000e-01	1.7097	5.084800e+00
svm	Support Vector Regression	2.157520e+01	1.077274e+03	3.115620e+01	-1.921000e-01	1.4327	2.138500e+00
kr	Kemel Ridge	2.748320e+01	1.211006e+03	3.356440e+01	-6.926000e-01	1.7330	5.223700e+00
par	Passive Aggressive Regressor	2.991290e+01	1.618128e+03	3.779610e+01	-1.019600e+00	1.8444	5.210300e+00
tr	TheilSen Regressor	5.174120e+01	1.409739e+04	8.783650e+01	-2.210740e+01	1.8876	1.287900e+01
lr	Linear Regression	1.463086e+04	2.369494e+09	1.547910e+04	-2.190026e+06	2.2493	2.870637e+03
lar	Least Angle Regression	1.374532e+09	2.038188e+19	1.506701e+09	-1.281867e+16	9.3333	2.551294e+08

Fig 3. Comparación de modelos para call center 1. (Fuente propia)

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
catboost	CatBoost Regressor	6.8624	1.789972e+02	12.3352	3.305000e-01	0.9515	0.6484
lightgbm	Light Gradient Boosting Machine	7.2338	1.822836e+02	12.5929	3.155000e-01	1.0328	0.6409
gbr	Gradient Boosting Regressor	7.4503	1.857540e+02	12.7481	2.946000e-01	1.0914	0.6650
rf	Random Forest Regressor	6.8185	1.941796e+02	13.0071	2.256000e-01	0.8224	0.6765
et	Extra Trees Regressor	6.6905	1.979407e+02	13.1980	2.188000e-01	0.7854	0.6684
xgboost	Extreme Gradient Boosting	7.1298	2.052922e+02	13.4552	2.061000e-01	0.9349	0.6691
ridge	Ridge Regression	10.4997	2.168133e+02	13.9772	1.614000e-01	1.6819	0.7869
ard	Automatic Relevance Determination	10.5261	2.165804e+02	14.0176	1.343000e-01	1.6814	0.7988
br	Bayesian Ridge	10.5751	2.195162e+02	14.0959	1.261000e-01	1.6855	0.8051
lr	Linear Regression	10.5036	2.248256e+02	14.1453	1.582000e-01	1.6743	0.7825
huber	Huber Regressor	8.9578	2.366360e+02	14.2577	1.576000e-01	1.3781	0.7053
tr	TheilSen Regressor	10.7852	2.312849e+02	14.3592	1.398000e-01	1.7154	0.7945
ada	AdaBoost Regressor	9.9155	2.402518e+02	14.5847	8.290000e-02	1.4571	0.8100
knn	K Neighbors Regressor	8.3518	2.634700e+02	15.1148	1.102000e-01	1.1451	0.7713
mlp	MLP Regressor	10.8757	2.813717e+02	15.6677	1.520000e-02	1.6538	0.8109
dt	Decision Tree Regressor	8.1699	2.854151e+02	15.8211	-8.620000e-02	0.9494	0.8387
omp	Orthogonal Matching Pursuit	11.7354	3.150927e+02	16.7208	-1.836000e-01	1.7280	0.9067
llar	Lasso Least Angle Regression	12.0805	3.262930e+02	16.8286	-9.550000e-02	1.7515	0.8237
en	Elastic Net	12.1251	3.284561e+02	16.9511	-1.490000e-01	1.7685	0.9018
lasso	Lasso Regression	12.1481	3.318183e+02	17.0186	-1.433000e-01	1.7761	0.8998
kr	Kernel Ridge	12.3592	3.323161e+02	17.1499	-2.126000e-01	1.8003	0.9392
svm	Support Vector Regression	10.3386	4.169917e+02	19.1264	-4.221000e-01	1.8858	0.9920
par	Passive Aggressive Regressor	13.4756	4.798499e+02	20.5438	-7.580000e-01	2.0236	1.1538
lar	Least Angle Regression	3287.1036	1.964120e+08	4626.1172	-8.145839e+06	2.7494	455.8652

Fig 4. Comparación de modelos para call center 2. (Fuente propia)

Como se observa en el caso de cada conjunto de datos el resultado es diferente tanto en los valores de las métricas de evaluación, así como en el mejor modelo que se obtiene como resultado de la comparación, siendo para el primer call center el algoritmo “Light Gradient Boosting Machine” mientras que para el segundo call center el algoritmo con menor error es “Catboost”.

Aunque en ambos casos se ve que el mejor algoritmo es un método de ensamble, o combinación, de modelos basado en arboles de decisión, no se podría asegurar con total certeza que para todo call center o todo problema de análisis de series temporales este tipo de algoritmos sean los mejores.

Con PyCaret el proceso de entrenamiento y evaluación de las diferentes métricas se lo realiza usando validación cruzada en k iteraciones o k-fold cross validation, solamente usando los datos de entrenamiento, que en este caso son el 80% de los datos, por eso es importante analizar cómo se comporta el mejor modelo elegido en el conjunto de datos de pruebas o test o hold-out, que al ser datos dejados de lado en el entrenamiento, nos permite tener una idea

de que tan bueno resulta el modelo con datos nuevos. Estos resultados se pueden observar en las **Figuras 5 y 6**.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Light Gradient Boosting Machine	5.5328	106.274	10.3089	0.7658	0.5109	0.443

Fig 5. Resultados en conjunto de datos de prueba para call center 1. (Fuente propia)

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	CatBoost Regressor	5.0481	77.5178	8.8044	0.1275	0.9227	1.0118

Fig 6. Resultados en conjunto de datos de prueba para call center 2. (Fuente propia)

En general ambos resultados son bastante buenos, mejores que los resultados obtenidos en comparación general, esto para ambos call center, lo que indica que no hay un sobreajuste u overfitting. Finalmente se hace también una optimización de hiperparámetros, para obtener el mejor resultado final posible.

Los resultados de la evaluación final, posterior a la optimización de hiperparámetros, se los puede observar en las **Figuras 7 y 8**:

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Light Gradient Boosting Machine	4.591	55.9303	7.4787	0.8767	0.5242	0.5182

Fig 7. Resultados de test posterior a optimización de hiperparámetros para call center 1. (Fuente propia)

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	CatBoost Regressor	2.3597	19.3814	4.4024	0.7819	0.5236	0.4521

Fig 8. Resultados de test posterior a optimización de hiperparámetros para call center 2. (Fuente propia)

Como se observa claramente, la optimización de hiperparámetros sí logra reducir el error, si bien para el segundo call center la mejor es más notable, ya que reduce el error a la mitad aproximadamente, en el primer call center también existe una mejora bastante buena.

Entonces se puede concluir que, a nivel de algoritmo, los resultados de error son bastante buenos, y para confirmar los resultados, se creó en una gráfica de los datos reales (count) contra los datos predichos (Label). Estos resultados se encuentran en la **Figura 9 y 10**.

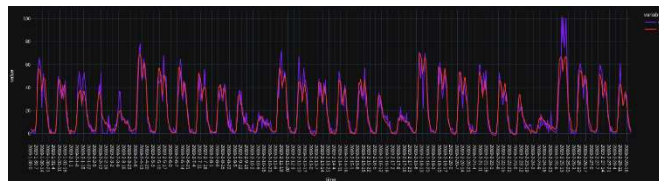


Figura 9. Valores reales y valores predichos de call center 1. (Fuente propia)

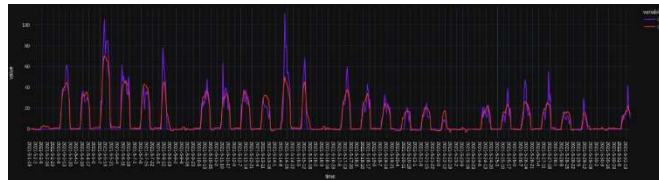


Figura 10. Valores reales y valores predichos de call center 2. (Fuente propia)

En las **Figuras 9 y 10**, se puede apreciar que los valores predichos, que se observan con color rojo, son bastante buenos al compararlos con los valores reales en color morado, esto en general para ambos call center, aunque a simple vista son mejores en el primer call center ya que en el segundo call center no se predice bien algunos picos que se observan claramente en la gráfica, pero a pesar de eso y considerando que son mucho menos datos que se tienen, el resultado es aceptable.

Cabe mencionar que el proceso de comparación de todos los modelos posibles con PyCaret, toma un determinado tiempo que depende de la cantidad de datos y que por ejemplo en el caso del conjunto de datos del primer call center, que es el más extenso, el evaluar todos los modelos tomó aproximadamente dos horas, esto teniendo en cuenta que son datos de

prácticamente más de tres años, pero en el caso del conjunto de datos del segundo call center, que apenas son algo menos de dos meses de datos, evaluar todos los algoritmos tomo alrededor de dos minutos, una diferencia bastante considerable.

Por ello el parámetro “turbo” se lo considera importante dejarlo como una funcionalidad del software para poder decidir si se desea evaluar todos los algoritmos, considerando el tiempo que podría tardar dependiendo de la cantidad de datos, o solo los más rápidos, ya que si bien en los dos conjuntos de datos evaluados, los mejores algoritmos resultaron también de los más rápidos, no se puede asegurar que ese siempre sea el caso.

Evaluación de software desarrollado

Para la evaluación del software como tal, se comparte acceso al sistema para su validación y pruebas a tres expertos en administración, desarrollo y comercialización de call center. Eduardo Javier Martínez Toloza, experto administrador de callcenter con 20 años de experiencia en toda área operativa de un call center; Luis Felipe Machado Redrobán, Ingeniero con 10 años de experiencia en desarrollo de software de call center y telefonía IP quien cuenta con su propia solución de gestión telefónica y de call center llamada Contactvox; y Carlos Billy Albán Méndez un Ingeniero con 12 años de experiencia comercializando soluciones de call center y telefonía IP.

A los tres expertos se los encuesta con preguntas calificadas de 1 a 5, donde 5 es la mejor calificación. Estas preguntas se las puede observar en la Tabla 1.

Tabla 1. Preguntas de evaluación. (Fuente propia)

Número de pregunta	Descripción
1	El acceso al sistema es óptimo
2	El tiempo de respuesta del sistema es adecuado
3	El manejo del sistema es sencillo y amigable

4	Los resultados obtenidos del sistema son claros y ayudan a cumplir con el nivel de servicio deseado de un <i>call center</i>
5	El software tiene aplicación real para <i>call center</i> de atención

De esta encuesta se obtuvo resultados totalmente favorables al software, como se puede observar en la tabla 2, donde vemos los resultados de las encuestas.

Tabla 2. Resultados de encuesta de evaluación. (Fuente propia)

Número de Pregunta	Respuestas de:			Promedio
	Luis Machado	Carlos Alban	Eduardo Martínez	
1	5	5	5	5
2	5	5	5	5
3	5	5	5	5
4	5	5	5	5
5	5	5	5	5

Por lo tanto, todos coinciden en que, en general, se encuentran satisfechos con el resultado, y que consideran, al software presentado, como una gran ayuda para las labores de programación de horarios de un *call center* ya que simplifica el trabajo de realizar los cálculos de Erlang para estimar el nivel de servicio y recursos necesarios por cada hora, y da el gran valor agregado de una predicción de arribo de llamadas comparada entre varios algoritmos. También coinciden, en que, en su estado actual, es un prototipo, ya que se sugiere hacer mejoras de detalles visuales de la aplicación y potenciales mejoras de funcionalidades, como mostrar gráficas de los resultados, pero tiene mucho potencial de ser un producto o herramienta completa si se combina con un software de gestión de horarios y con un software

de distribución de llamadas de call center. Es decir, se lo percibe como una gran ayuda y muy funcional en su estado actual, pero a su vez se lo ve como una parte o una funcionalidad de un software más grande de gestión total de call center.

Análisis y discusión

Como se puede analizar en la evaluación de los algoritmos de regresión, para los conjuntos de datos usados, los algoritmos basados en árboles de decisión con mejoras de gradiente o de tipo ensamble, son los que en general dan mejores resultados, específicamente fueron “Light Gradient Boosting Machine”, “Catboost regressor” y “Gradient Boosting Regressor” los que menor error presentaban, y a pesar de no específicamente el mismo algoritmo resultó el mejor para ambos casos, si es un algoritmo del mismo tipo el que en los dos casos resulta mejor.

Aun así, no se puede asegurar que en todo call center van a ser los algoritmos de este tipo que den los mejores resultados en todos los casos, siendo así, que el software desarrollado permite justamente, identificar qué modelo resulta mejor, en cada caso individual, y con más conjuntos de datos para experimentar, se podría ir comprobando si en efecto algún algoritmo en específico funciona bien para predecir arribo de llamadas y dedicar ese algoritmo para el problema en general de todo call center.

La gráfica de los valores predichos y los valores reales en el tiempo es una guía fundamental para validar que el algoritmo seleccionado usando las métricas de error, en efecto funciona, porque solo con los valores de error no se tiene la certeza de que los valores en efecto se ajustan lo más posible a la realidad, porque se puede presentar casos que el error es bajo, pero al analizar la gráfica se observa que los valores no se ajustan a la realidad como se desea.

Con los resultados presentados, se valida que se puede predecir el arribo de llamadas, tratando el problema como una serie temporal, mediante el uso de algoritmos propios del aprendizaje automático, como lo son los árboles de decisión, y que esta predicción es la base para obtener el tráfico de llamadas y poder mediante teoría de colas y las fórmulas de Erlang-C, estimar los agentes necesarios por franja horaria, para cumplir con un nivel de servicio deseado.

Finalmente, el software tiene un funcionamiento idóneo, intuitivo, de fácil manejo, con tiempos de respuesta adecuados, y que prácticamente se puede usar en cualquier call center de forma real, con potencial de mejora para pasar de un prototipo a un producto o complemento de otros productos.

Conclusiones

Se diseñó un software para programar horarios, que, basado en históricos, predice el arribo de llamadas por horas o por franja horaria, mediante el uso de todos los algoritmos de regresión planteados en el trabajo, permitiendo determinar también, los recursos humanos necesarios para un call center, por hora de acuerdo a un nivel de servicio deseado.

Se implementó el software diseñado en una aplicación web, usando HTML, PHP y Python, con el último, se realizó el trabajo de aprendizaje automático, a través del uso de la librería PyCaret se simplificó el código y se realizó todas las tareas de creación, evaluación y comparación de 19 o 25 modelos de algoritmos de regresión de forma correcta, implementado también fórmulas de Erlang-C con las librerías Math y Pandas, obteniendo resultados en HTML que puedan visualizarse de forma correcta en la aplicación web.

Se comparó las predicciones de arribo de llamadas obtenido, usando el RMSE como métrica de error guía, observando que los algoritmos con los resultados más óptimos fueron “Light Gradient Boosting Machine”, “Catboost regressor” y “Gradient Boosting Regressor”, los cuales presentaron errores muy similares entre sí, con diferencias en el orden de las décimas, concluyendo así que para los call center considerados en el presente trabajo, se puede considerar únicamente los tres algoritmos mencionados, confirmando que, los algoritmos basados en árboles de decisión con mejora de gradiente funcionan muy bien para problemas de series temporales cómo es el caso de este estudio.

Se evaluó el software con tres expertos, uno en manejo de callcenter, uno en desarrollo de software de callcenter y uno en comercialización de software de call center, los cuales

manifestaron, que el software cumple con el objetivo de usar inteligencia artificial para predecir el arribo de llamadas, creando un horario idóneo y la asignación de recursos que permitan cumplir con un nivel de servicio deseado.

Se alcanzó un nivel de servicio de 80% de llamadas contestadas en menos de 20 segundos en un call center, al hacer uso de horario generado por el software desarrollado, donde también se indica los recursos humanos que se necesitan por franja horaria, basado en la predicción de arribo de llamadas por hora y las fórmulas de Erlang-C.

El presente trabajo abre la posibilidad primero a tener versiones mejores a futuro, donde se dé más posibilidades al usuario de probar, optimizar y mejorar entre las varias opciones de algoritmos posibles, así como principalmente abrir la posibilidad a que el usuario elija el algoritmo y se compare entre algoritmos optimizados.

Otro trabajo a futuro es la posibilidad de integrar el software con otros softwares de call center, principalmente con los softwares de distribución de llamadas, que son los que tienen los datos de arribo de llamadas reales, esto como una sugerencia del usuario y algo que lógicamente ayudaría al software a ser más utilizado.

Si bien para hacer el presente software lo más práctico y general se usó solamente la fecha y hora para obtener las variable predictoras, hay que considerar que se puede usar variables adicionales para buscar aún mejores resultados, es decir que, a pesar del uso general de la herramienta actual, siempre un análisis más profundo de los datos de cada caso es lo mejor, pudiendo usarse esta herramienta desarrollada como un punto de partida para este análisis más a profundidad, o usar la herramienta como un complemento para un análisis que ya se tenga hecho en cada caso específico de cada call center.

Referencias

- [1] Ali, M. (Julio de 2020). PyCaret: An open source, low-code machine learning library in Python. Obtenido de <https://pycaret.org/about>
- [2] Deputechologies Pty Ltd. (2021). Obtenido de <https://www.deputy.com/industry/call-centre>



- [3] Genesys. (2021). CALL CENTRE FORECASTING. Obtenido de <https://www.genesys.com/en-gb/capabilities/forecasting-and-decisions>
- [4] Kampakis, S. (3 de Enero de 2020). thedatascientist. Obtenido de <https://thedata scientist.com/performance-measures-rmse-mae/>
- [5] Nice. (2021). Workforce Management. Obtenido de <https://www.nice.com/engage/workforce-management/>
- [6] scikit-learn developers. (2021). Metrics and scoring: quantifying the quality of predictions. Obtenido de https://scikit-learn.org/stable/modules/model_evaluation.html
- [7] Shiftboard, Inc. (2021). Obtenido de <https://www.shiftboard.com/industries/call-centers/>
- [8] TixTime, Inc. (2021). Obtenido de <https://www.tixtime.com/employee-scheduling-software/call-center-scheduling-software/>
- [9] Verint. (2021). Verint Monet Workforce Engagement. Obtenido de <https://www.verint.com/es/engagement-5/our-offerings/solutions/small-and-medium-sized-businesses/verint-monet/>

Copyright (2024) © Luis Aguirre

Este texto está protegido bajo una licencia internacional Creative Commons 4.0.



Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)