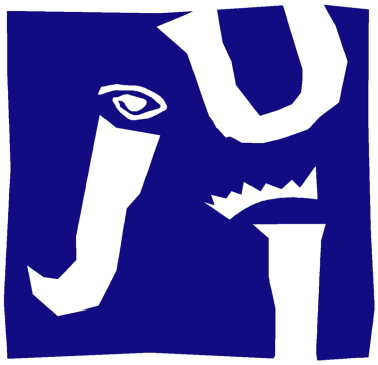

Detección de Ofertas de Trabajo Fraudulentas

Trabajo Final de Asignatura MT1033

Vicente López Oliva

Castelló de la Plana (Spain)

Tabla de Contenido

Chapter 1

Contextualización

En este apartado vamos a contextualizar el problema que vamos a resolver durante el resto del trabajo. Describiremos tanto el problema detectado como los datos con los que vamos a trabajar, así como de dónde hemos obtenido los datos. Comentaremos también el estado actual de las soluciones a este problema.

Remarcamos que todo el código utilizado durante el trabajo estará disponible en un repositorio público de Github [1].

1.1 Problema

Vivimos en una era en la que la globalización y la informatización de todos los procesos burocráticos comienzan a ser una realidad. Cada vez son más las empresas que optan por resolver sus trámites y ofrecer sus productos de forma telemática. En particular, muchas empresas están recurriendo a soluciones online para contratar nuevo personal. Páginas web como LinkedIn o Indeed son cada vez más usados por los responsables de recursos humanos para captar nuevo talento para la empresa. Esto conlleva de forma directa que cada vez es más común buscar trabajo en internet, de forma que la persona puede acceder a más ofertas de trabajo y encontrar la que más se adecúe a su perfil en mucho menos tiempo, beneficiando así tanto a la empresa empleadora como al propio empleado.

No obstante, todo tiene una contra parte. Estas ofertas online puede ser utilizadas por gente con malas intenciones. En particular, pueden crear ofertas falsas sobre una empresa concreta con el fin de acabar dañando su imagen,

[1] https://github.com/romOlivo/FraudDetection.git

técnica que podría ser utilizada por potenciales competidores. Este problema se está haciendo común en nuestros días y ha crecido hasta convertirse en uno de los problemas recogidos por la ORF (1) (Online Recruitment Frauds).

Podemos pensar que este problema puede ser resuelto disponiendo operarios que comprueben las ofertas de trabajo y filtren así aquellas de carácter sospecho, pero entonces nos encontraremos con que la publicación de ofertas fraudulenta puede ser automatizada, suponiendo así unos costes de control que no podrían asumir los portales web de publicación de trabajos.

En este trabajo, vamos a proponer una solución para detectar ofertas fraudulentas de forma automática que permitirá identificar las ofertas fraudulentas una vez intenten ser publicadas, previniendo así que personas maliciosas puedan afectar la imagen tanto de la empresa empleadora como de los portales web en los que se hizo pública.

1.2 Soluciones

Sabemos que este problema es uno importante, es por ello que se han elaborado soluciones que comentaremos y analizaremos en esta parte. En especial, hablaremos de dos de los métodos más utilizados para resolver el problema. El análisis de los métodos utilizados se basará en un estudio realizado en 2020 por Shawni y Samir (2).

El primer método que vamos a considerar es el algoritmo ingenuo de Bayes (3). En las pruebas realizadas en el estudio mencionado, ofrece el menor de los rendimientos en comparación con el resto de los algoritmos.

Los clasificadores basados en el algoritmo del vecino más cercano (4) son populares en nuestros días con aplicaciones en muchos y muy variados campos, ofreciendo en general buenos resultados -aunque no siempre los mejores-. En nuestro caso particular, estos algoritmos ofrecen un buen resultado.

En el estudio propuesto se puede observar cómo los algoritmos probados ofrecieron en general una alta precisión. No obstante, debemos tener en cuenta que el problema al que nos enfrentamos es un problema de clases desbalanceadas (5), lo que quiere decir que la precisión sobre un conjunto aleatorio siempre será elevada.

1.3 Datos

Los datos que utilizaremos para explorar soluciones al problema los encontraremos en la página web Kraggle [2], proporcionados por Laboratory of Information & Communication Systems Security de la universidad de Aegean.

Esta base de datos la componen 18.000 ofertas de trabajo diferentes de las cuales sobre 800 son fraudulentas -lo que confirma que efectivamente estamos ante un problema de clases desbalanceadas-. Contaremos con una columna que nos indicará si una oferta es fraudulenta. Entre los datos que tendremos disponibles sobre las diferentes ofertas de trabajo, dispondremos de su título, rango salarial, departamento, país entre otros atributos de utilidad. Analizaremos en profundidad los datos disponibles en los siguientes capítulos.

En una aproximaciones inicial, vemos que todas las variables son categóricas, no contamos con ninguna variable cuantitativa. Sabemos que el rango salarial podremos transformarlo en una variable cuantitativa sin mayores problemas, pero se deberán tratar las demás variables para poder transformarlas en variables cuantitativas sin difuminar, cambiar o perder la información que tienen o, en caso de no ser posible, reducir al máximo las consecuencias.

[2] https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction

Chapter 2

Análisis Exploratorio

En este capítulo realizaremos un análisis exploratorio de los datos. Indagaremos en la distribución de los datos, cantidad de datos y correlaciones. Se realizará un PCA (Principal Component Analysis) de los datos para reducir la dimensionalidad de los datos.

2.1 Descripción de los Datos

La dimensionalidad exacta de nuestros datos es de 17880 en los que contamos con 18 variables diferentes, que son las siguientes:

1. **job_id**: Identificador de la oferta de trabajo. Sirve para identificar de forma unívoca a cada oferta de trabajo.
2. **title**: Título bajo el que se publicó la oferta
3. **location**: Localización de la oferta. Se refiere al lugar en el que se oferta la vacante. Se incluye país y ciudad.
4. **department**: Departamento de la empresa que oferta el puesto.
5. **salary_range**: Rango salarial ofertado. Habitualmente númerico que incluye el mínimo y máximo salario que la empresa está dispuesta a aceptar en relación al perfil del ofertante.
6. **company_profile**: Perfil de la compañía. Opcional. Descripción introducida en el perfil de la empresa creado en el portal web.

7. **description**: Descripción del puesto de trabajo. Opcional. La empresa puede detallar las funciones a realizar o que se espera que se realicen.
8. **requirements**: Requerimientos exigidos y detallados necesarios para poder aplicar a la oferta de trabajo.
9. **benefits**: Beneficios por trabajar para dicha empresa. Opcional.
10. **telecommuting**: Valor Booleano (solo puede tomar dos valores, verdadero o falso). Se refiere a si la empresa ofertante se enmarca dentro de las telecomunicaciones
11. **has_company_logo**: Valor Booleano. Denota si la empresa que publicó la oferta está registrada con un logo, es decir, ha añadido un logo al perfil creado en el portal web en el que publicó la oferta.
12. **has_questions**: Valor Booleano. Denota si el interesado en aceptar la oferta puede realizar preguntas acerca de ella a la empresa dentro del portal web.
13. **employment_type**: Tipo de contrato que se ofrece para el puesto.
14. **required_experience**: Denota la experiencia mínima necesaria para poder optar al trabajo ofertado.
15. **required_education**: Indica la educación mínima requerida para poder aplicar a la oferta.
16. **industry**: Industria en la que se enmarca la empresa que realiza la oferta.
17. **function.**: Función a desempeñar en el puesto. Se refiere al tipo de trabajo a desempeñar (no confundir con el cargo).
18. **fraudulent**: Valor Booleano. Denota si una oferta es o no fraudulenta. En particular es el valor a predecir.

Se encuentran exactamente 866 ofertas fraudulentas frente a las 16994 ofertas no fraudulentas que componen el dataset. Esto confirma que nos encontramos efectivamente ante un problema de clases desbalanceadas que deberemos solventar más tarde cuando conformemos los datasets (ver apartado 3.1).

Disponemos de muchas variables categóricas que va a ser necesario estudiar de forma individualizada. Primero atendemos a las variables ”title”, ”company_profile”,

”description” y ”benefits”. Estas variables están conformadas por texto plano. Para poder transformar dicho texto en información que nuestro modelo pueda utilizar será necesario aplicar técnicas de NLP (6) (Natural Language Processing). Ellas escapan de lo esperado en este trabajo, es por ello que vamos a descartar dichas variables. No obstante, queda abierta a su aprovechamiento mediante alguna transformación que elimine la complejidad del problema.

Vamos a explorar ahora la variable ”required_experience”. Analizamos primero sus valores únicos y nos damos cuenta de que existen valores nulos, que decidimos agrupar en la categoría de no aplicable. En el histograma mostrado en la Figura 2.1 podemos ver como la categoría de no aplicable es la mayoritaria, pero que no obstante los valores están distribuidos entre las diferentes categorías.

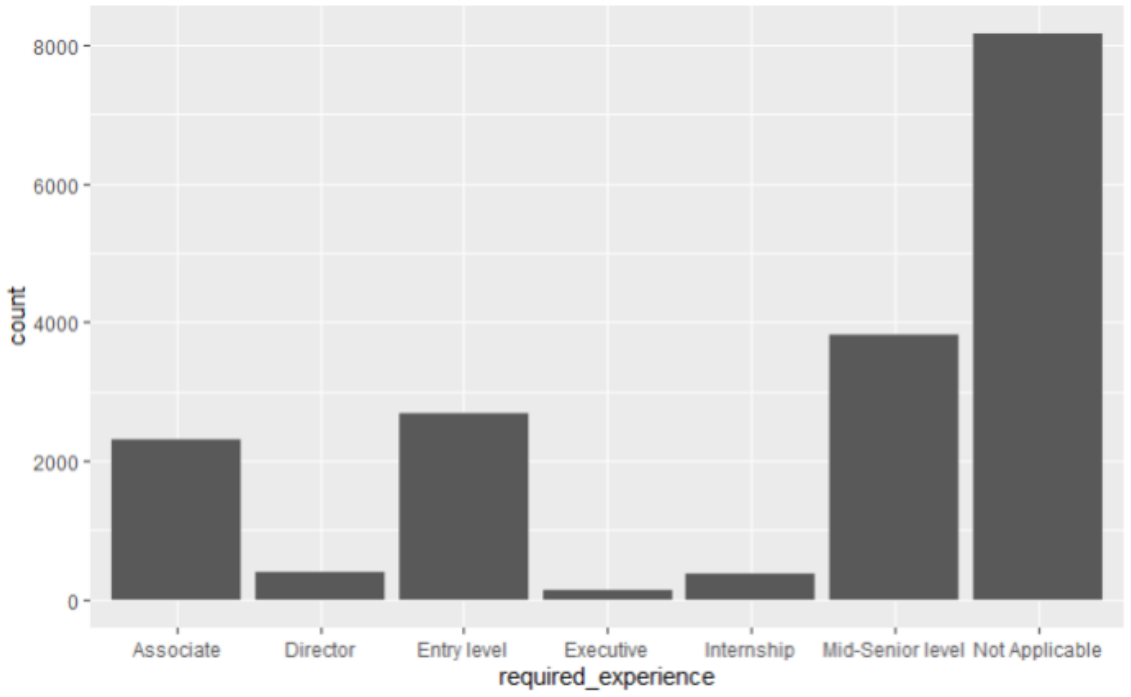

Figure 2.1: Histograma de la experiencia requerida.

Si atendemos a la variable la variable ”required_education” tenemos que existen categorías que signifcan lo msimo. Al unirlas, resulta en el histograma mostrado en la Figura 2.2. Igual que en el caso anterior, vemos que la categoría de no especificado es mayoritaria, pero existen otras clases que también son representativas. Así mismo, observamos también que existen muchas clases con muy

poca representación.

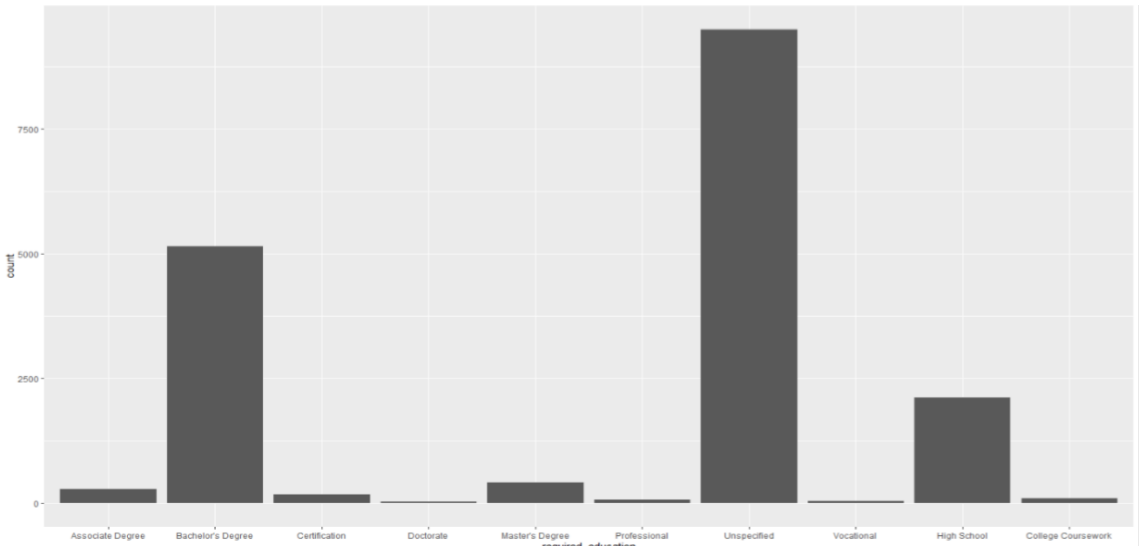

Figure 2.2: Histograma de la formación requerida.

Centrando nuestra atención en la variable ”emplyment_type”, la situación es muy parecida a la descrita en los casos anteriores. Como muestra la Figura 2.3, tenemos pocas categorías y todas ellas tienen valores representativos, siendo en esta ocasión mayoritario el grupo de contratos a tiempo total.

Nos vamos a fijar ahora en el resto de variables que nos quedan. Atendiendo a ellas, vemos que la diversidad de categorías es muy extensa, en la mayoría de los casos demasiado para poder extraer información de ellas con tan poco volumen de datos. De ellas vamos a seleccionar dos en concreto con las que trabajaremos después, que son ”location”, ya que podremos transformarla fácilmente y con ”function.” que posee pocas categorías que pueden ser tratadas (posee exactamente 38 categorías). El resto de variables posee más de 300 valores diferentes y es por ello que deciden descartarse.

Las variables Booleanas las vamos a seleccionar todas debido a que podemos transformarlas fácilmente en valores cuantitativos sin perder información, ya que podemos considerar los valores falsos como 0 y los valores verdaderos como 1. No se mostrarán gráficos de estas variables por carecer de interés.

Mención aparte recibe la variable ”salary_range”, que no puede ser analizarse actualmente por ser una variable categórica con más de 800 valores diferentes, pero que puede ser transformada con facilidad.

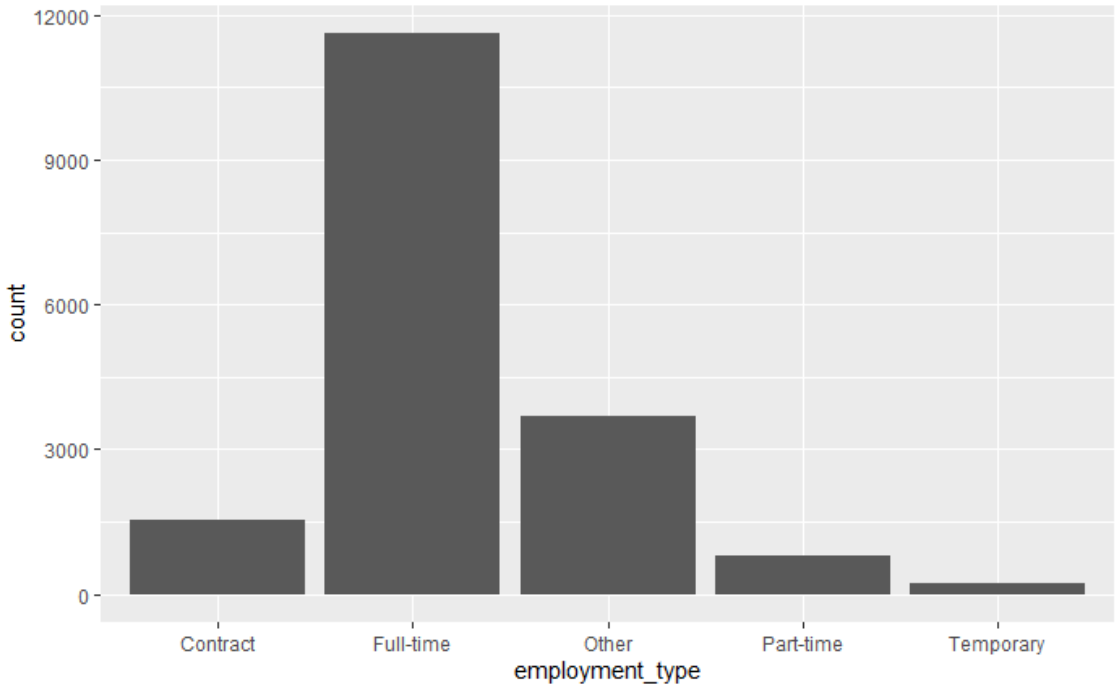

Figure 2.3: Histograma del tipo de contrato.

2.2 Transformaciones

En primer lugar vamos a tratar las variables descartadas, que son ”title”, ”company_profile”, ”description” y ”benefits”. En el apartado anterior descartamos su uso por escapar a la complejidad esperada, pero vamos a aprovechar cierta información con una sencilla transformación. Vamos a crear unas nuevas variables llamadas ”has_title”, ”has_company_profile”, ”has_description” y ”has_benefits” que denotaran si en la oferta fueron introducidos dichos campos. Al hacerlo, nos encontramos conque todos los títulos y descripciones fueron introducidos de forma válida, por lo que eliminamos las dos variables correspondientes.

La siguiente variable que vamos a transformar es ”salary_range”. De ella, vamos a separar los valores por el guión (el símbolo -), tomando el primer valor como el salario mínimo y el segundo valor como salario máximo. En caso de no existir el valor tomado como separador o encontrarnos un valor no numérico en la posición donde se debería especificar el valor, introduciremos un nulo. Después de elaborar estas dos variables, deduciremos una nueva variable: salario medio. Con ellas 3 decidimos crear un gráfico para poder observar su dispersión, que se muestra en

la Figura 2.5. Vemos que existen, en los tres casos, valores muy lejanos al grupo más común. Esto lo podemos explicar por el valor de la moneda, ya que cada país cuenta con una moneda distinta y no es lo mismo pensar en dólares que en yenes.

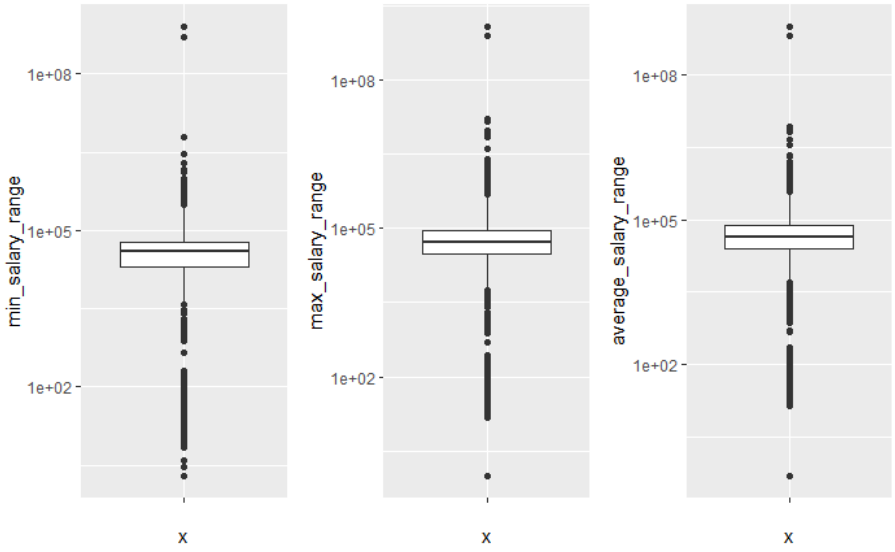

Figure 2.4: Boxplot de los rangos salariales en escala logarítmica.

Comenzamos con las transformaciones relativas a las variables categóricas propiamente dichas. Vamos a estudiar las variables ”employment_type”, ”required _ experience” y ”required_education”. Estas tres variables son especiales ya que podemos definir un orden sobre ellas. Ordenamos los factores y transformamos las variables en variables numéricas y elaboramos un boxplot de ellas, mostrado en la Figura para ver de nuevo su distribución. Este gráfico nos dice a golpe de vista que ninguna de las categorías de ninguna variable es poco representativa o acapara la mayoría de los valores, por lo que consideramos que es una transformación interesante y que podemos utilizar.

La siguiente variable categórica que vamos a transformar es la variable ”function.”. En esta variable se detalla la función a desempeñar dentro de la empresa. Listando las categorías nos encontramos con 38 funciones diferentes, que hemos agrupado finalmente en 8 grandes grupos, a saber, grupo desconocido (no se dispone información sobre la función a desempeñar), marketing, finanzas, empleos internos, ciencias, letras, artistas y directivo. Transformamos cada grupo en una variable Booleana que indica si cada oferta pertenece a cada grupo o no.

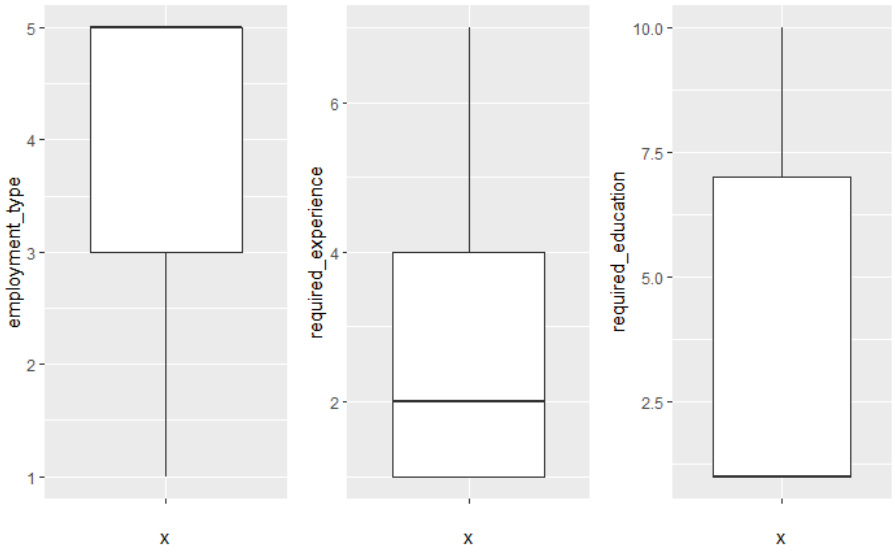

Figure 2.5: Boxplot de las variables ordenadas.

Se necesita descartar uno de los grupos (ya que si una oferta no pertenece a ninguno de los grupos anteriores, entonces pertenece al grupo que falta) y se decide descartar el grupo desconocido.

Por último, vamos a transformar la variable ”location”. Inicialmente se disponen de 367 categorías diferentes. Es por ello que decidimos extraer de la variable solo es país, olvidando la ciudad desde la que se realiza la oferta. Al hacerlo, reducimos la cantidad a 91 países. En base a criterios de moneda (valor monetario de la moneda legal del país) y a su situación económica, agrupamos los 91 países en 10 categorías diferentes que finalmente representamos cada una con una variable que indica si cada oferta pertenece a dicho grupo o no. Finalmente descartamos uno de los grupos, que es el grupo de los países desconocidos.

Después de todas las transformaciones, disponemos de 28 variables numéricas con las que vamos a poder aplicar nuestros algoritmos. Incluimos un resumen estadístico de las variables resultantes en la Figura 2.6. En ella podemos ver que se han generado nuevos valores nulos al realizar las transformaciones. Como al eliminarlos tendríamos un conjunto de datos de un tamaño aceptable, decidimos eliminar todas aquellas ofertas de la que no se dispone de algún dato.

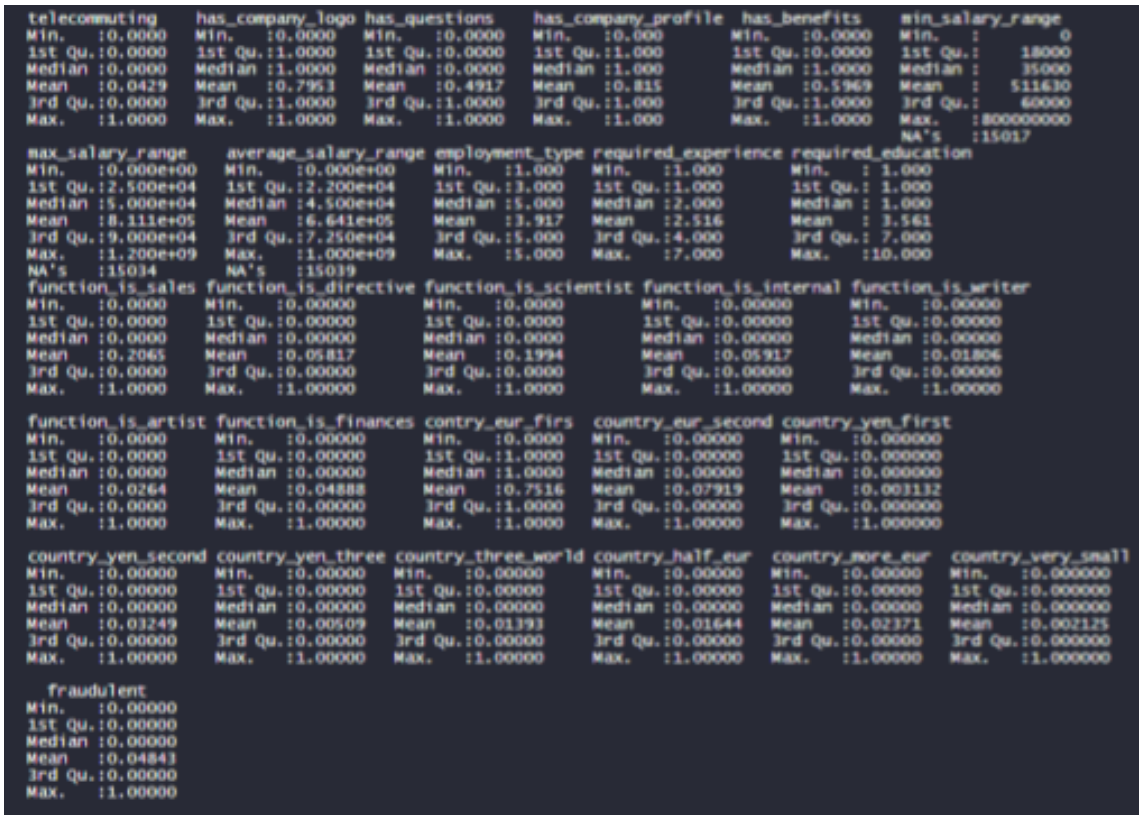

Figure 2.6: Resumen estadístico de las variables.

2.3 Correlaciones y PCA

En esta sección vamos a estudiar las relaciones entre las variables para elegir un subconjunto de métricas para realizar el PCA y así reducir nuestra dimensionalidad a una más reducida. En primer lugar vamos a estudiar las correlaciones entre nuestras variables. Para ello nos valdremos del mapa de calor mostrado en la Figura 2.7. En él podemos observar (en una escala donde el amarillo representa la máxima correlación y el azul oscuro representa la no correlación) cómo las correlaciones más fuertes se presentan en las 3 variables deducidas de los salarios: máximo, mínimo y promedio. Es por ello que decidimos quedarnos únicamente con el salario promedio. En cuanto al resto de variables, se observa también una correlación elevada entre si la compañía tiene logo y si la compañía tiene descripción, pero dicha correlación no supera el 0.8, por lo que decidimos no descartar ninguna de las variables.

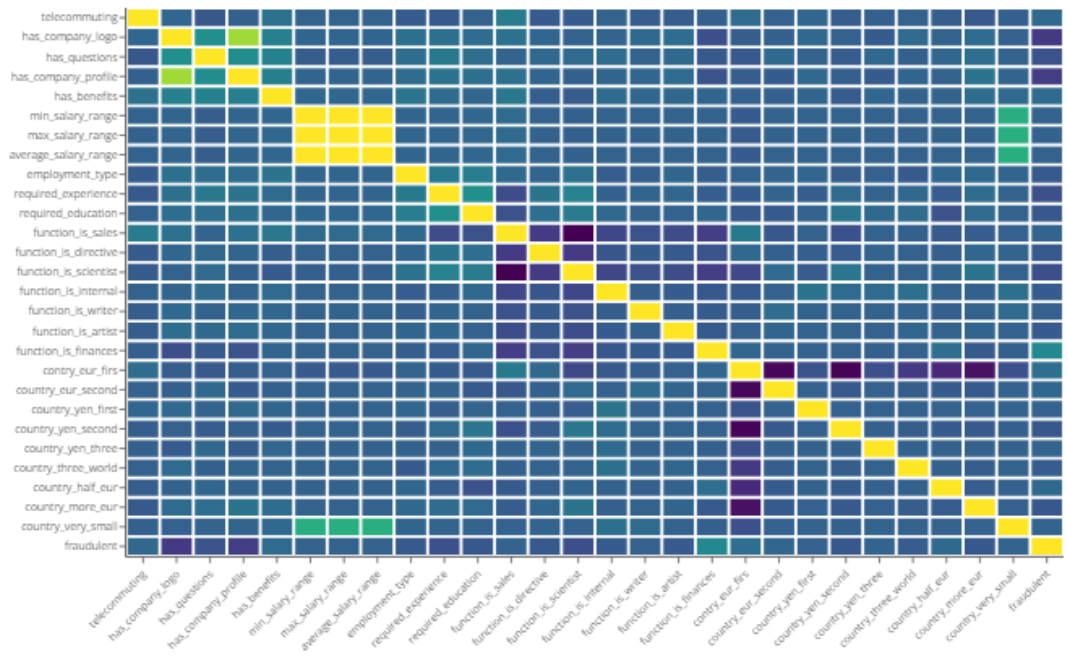

Figure 2.7: Mapa de calor de las correlaciones entre variables.

Realizamos el PCA sobre el nuevo subconjunto de métricas. La primera prueba la realizamos utilizando los datos puros, sin ningún tipo de transformación, pero cuando graficamos las dos primeras componentes (Figura 2.8), nos damos cuenta de que el análisis no ha sido efectivo, debido a que la mayoría de los datos parecen alinearse sobre una única línea, indicando que las correlaciones no han sido eliminadas.

Realizamos una segundo análisis utilizando esta vez la matriz de correlación, cuyos resultados se pueden ver en la Figura 2.9. En la gráfica se muestran en azul las ofertas fraudulentas y en negro aquellas ofertas no fraudulentas. En esta ocasión, podemos ver como la nube de puntos se distribuye por todo el espacio, significado de que esta vez las correlaciones sí han sido eliminadas y por tanto tenemos un mejor resultado.

Ahora debemos decidir con cuántas componentes nos quedamos. Cada componente representa una pequeña cantidad de información, consecuencia de haber utilizado muchas variables categóricas, especialmente con valores binarios, es por ello que la decisión la basaremos especialmente con el método del codo.

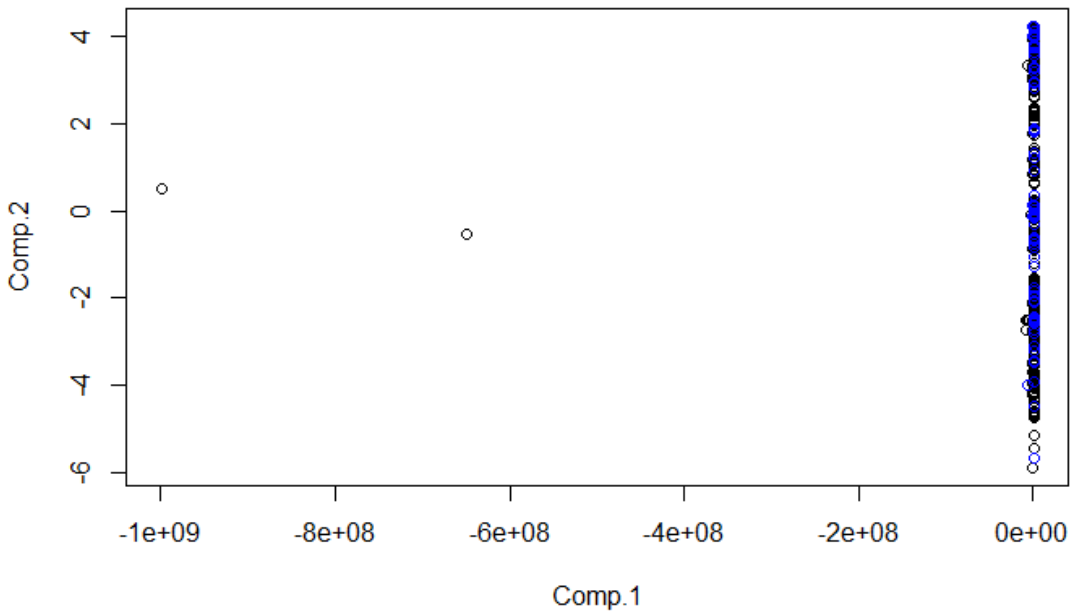

Figure 2.8: PCA sin transformaciones.

Atendiendo a la gráfica de la Figura 2.10, un buen corte es a partir de las 4 componentes, que explican un 29% de la varianza de los datos, aproximadamente un tercio de la información. Daremos por bueno dicho corte y nos quedaremos con las primeras cuatro componentes.

Con el análisis realizado, hemos conseguido reducir la dimensionalidad de nuestro problema de las 28 variables iniciales a únicamente 4 de ellas, reduciendo así la complejidad de nuestros modelos.

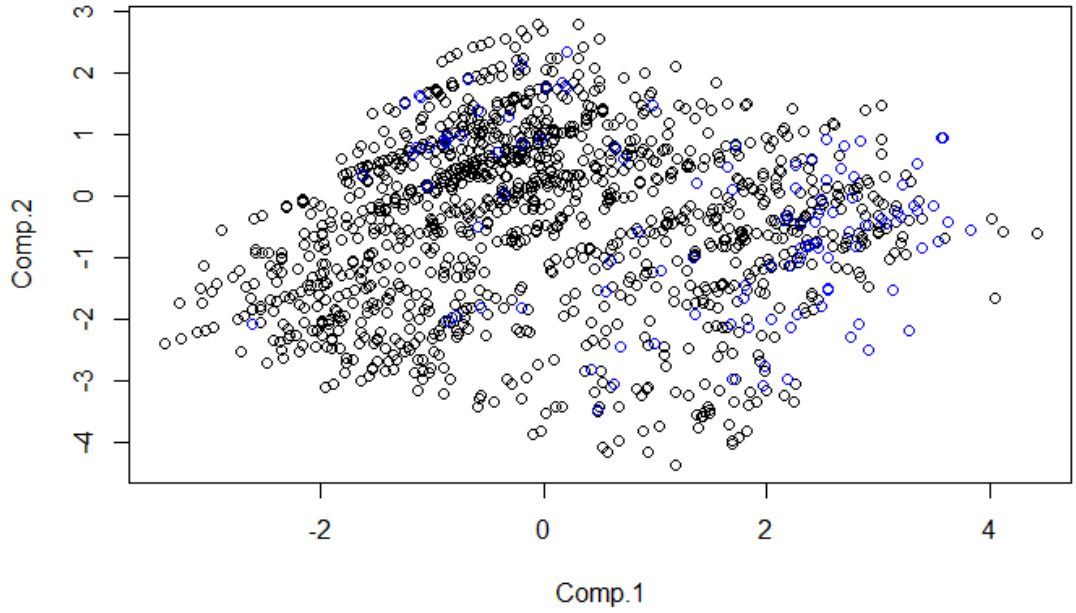

Figure 2.9: PCA con matriz de correlación.

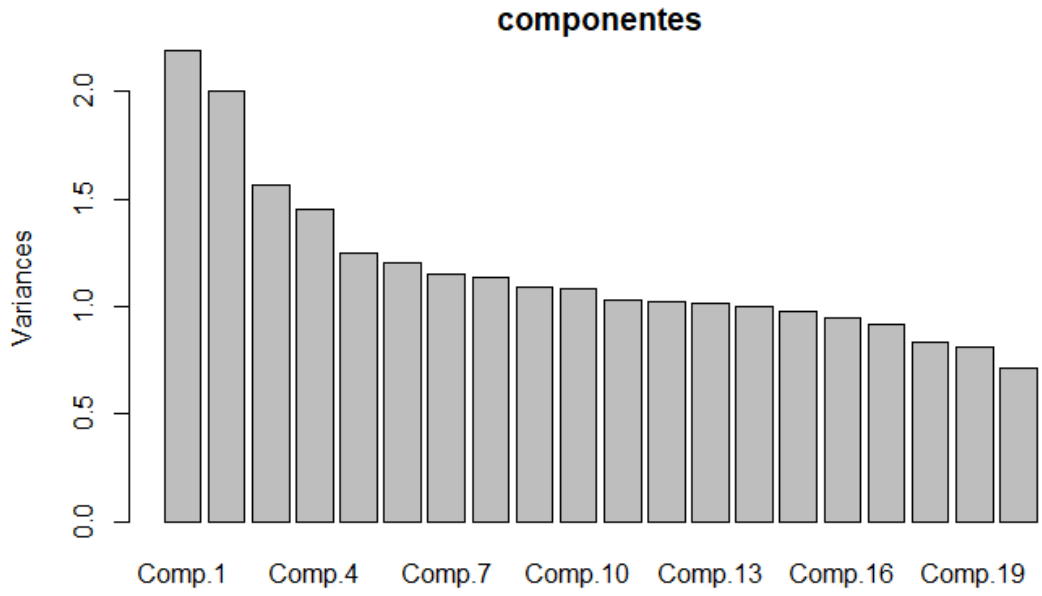

Figure 2.10: Varianza explicada por las componentes.

Chapter 3

Metodología

3.1 Datasets

En este apartado vamos a hablar de los diferentes datasets que vamos a preparar para trabajar con los distintos algoritmos. Será necesario aplicar algunos algoritmos para realizar los datasets que utilizaremos debido a que nos encontramos ante un problema de clases desbalanceadas. Aplicaremos dos técnicas diferentes para resolver el problema.

Oversampling: Mediante el uso de esta técnica, replicaremos los datos del grupo minoritario hasta equiparar en tamaño a la clase mayoritaria, es decir, dispondremos de 2619 individuos por cada clase.

Undersampling: Esta técnica defiende reducir el número de datos en la clase mayoritaria hasta igualar al número de representantes de la minoritaria. En nuestro caso, contaremos con clases con 222 elementos.

Además de aplicar estas técnicas, dividiremos los conjuntos en dos diferenciados, uno que utilizaremos para entrenar a los modelos y otro que usaremos para poner a prueba su capacidad. Usaremos el 20$\%$ de los datos disponibles para probar al modelo y el otro 80$\%$ para entrenarlo. Aplicando ambas técnicas, nos deja con grupos de tamaño 4190 para el conjunto de entrenamiento y 1048 para el de test utilizando la técnica de oversampling. Si atendemos a la técnica de undersampling, tendremos un conjunto de entrenamiento con 400 individuos y uno de pruebas con 88 elementos.

3.2 Algoritmos

Existen multitud de algoritmos que pueden utilizarse para tratar de resolver el problema. Nosotros vamos a centrarnos en el estudio de los siguientes algoritmos

1. **Clasificador Ingenuo de Bayes:** Como hemos avanzado en la introducción, es uno de los métodos a considerar cuando tratamos de resolver. Previsiblemente, y basándonos en estudios anteriores, dará los peores resultados y nos servirá para comparar la eficiencia de los demás métodos.

2. **K-Medoides:** En la introducción hemos resaltado la versatilidad y buenos resultados de los algoritmos basados en el vecino más cercano, es por ello que incluimos este algoritmos como representante de los algoritmos de dicha familia.

3. **Discriminador Lineal de Fisher:** Algoritmo propuesto en este trabajo para su estudio. Se trata de un clasificador lineal cuya idea intuitiva es trazar una línea que divida ambos grupos. La recta a trazar, si se consideran más de dos variables será una recta en un hiperplano.

Se eligieron estos algoritmos dado que son los que se han estudiado en la teoría, escogiendo algunos que ya han sido propuestos en estudios anteriores y que se presuponen que pueden funcionar bien y realizando una nueva propuesta para compararla con los de los estudios anteriores.

3.3 Presentación de Resultados

Vamos a proceder a realizar los diferentes experimentos con los algoritmos ya detallados. El primero que vamos a someter a estudio es el Clasificador Ingenuo de Bayes. Empezamos por probar su efectividad utilizando el conjunto sobre el que hemos utilizado la técnica de oversampling. Podemos ver la dispersión de las predicciones de los datos de entrenamiento en el gráfico de la Figura 3.1. Los resultados son los mostrados en la Figura 3.2, donde Falso Positivo indica el grupo de ofertas no fraudulentas que se han clasificado como ofertas fraudulentas, Falso Negativo el grupo de ofertas fraudulentas que han sido clasificadas como ofertas no fraudulentas, Positivo Acertado el grupo de ofertas fraudulentas clasificadas correctamente y Negativo Acertado denota el grupo de ofertas no fraudulentas clasificado correctamente. El algoritmo muestra una precisión del 62%.

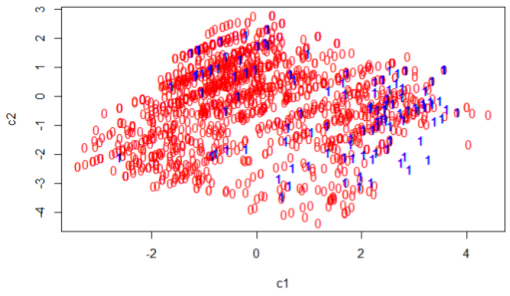

Figure 3.1: Clasificación Clasificador de Bayes con Oversampling.

Figure 3.2: Resultados Clasificador de Bayes con Oversampling.

Si utilizamos en cambio el conjunto de datos sobre que el que hemos aplicado los principios de undersampling, los resultados por el Clasificador de Bayes son los mostrados en la Figura 3.3, mostrando unos resultados muy parecidos a los anteriores (Figura 3.4) con una precisión del 60%.

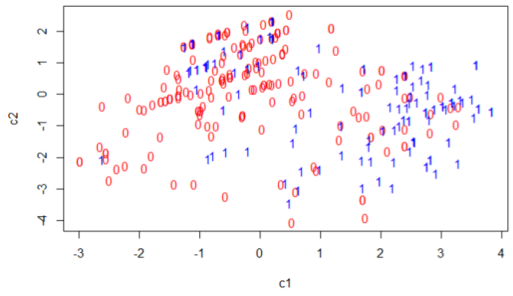

Figure 3.3: Clasificación Clasificador de Bayes con Undersampling.

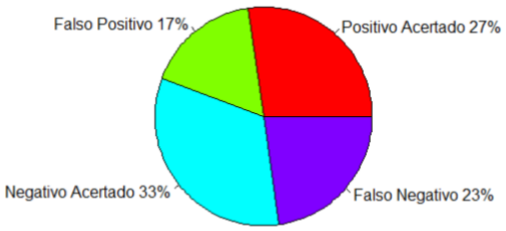

Figure 3.4: Resultados Clasificador de Bayes con Undersampling.

Pasemos ahora a ver los resultados del algoritmo de Vecino más cercano, en nuestro caso el k Medoides. Para nuestro ejemplo hemos utilizado 4 grupos (k = 4). Los resultados han sido de una precisión del 93% para los datos con oversampling (Figura 3.5) y de 75% para el dataset en los que se aplicó undersampling (Figura 3.6).

Figure 3.5: Clasificación Vecino más Cercano con Oversampling.

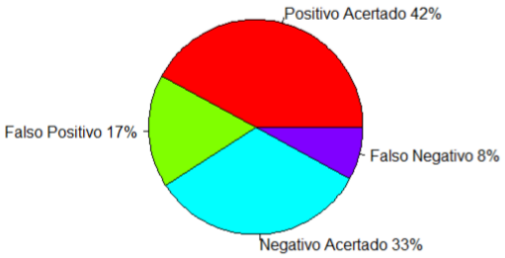

Figure 3.6: Resultados Vecino más Cercano con Undersampling.

Por último vamos a estudiar el discriminador Lineal de Fisher. Nuevamente probaremos su rendimiento tanto para el conjunto de datos sobre que el hemos aplicado oversampling como para el que hemos aplicado undersampling. Para el de oversampling, podemos ver su dispersión en las predicciones en la Figura 3.7, que alcanza un 68% de precisión, tal y como podemos ver en la Figura 3.8

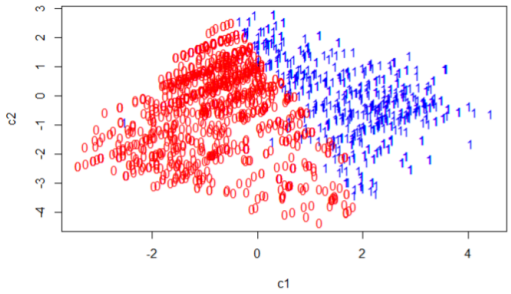

Figure 3.7: Clasificación Clasificador Lineal de Fisher con Oversampling.

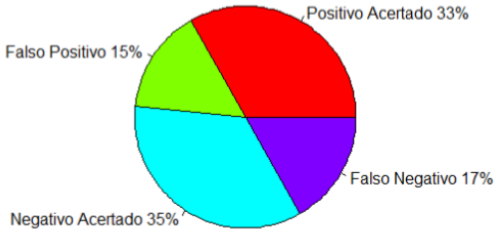

Figure 3.8: Resultados Clasificador Lineal de Fisher con Oversampling.

Atendiendo ahora al conjunto de datos de undersampling, nos encontramos con un gráfico de dispersión como el de la Figura 3.9. En este caso, según los resultados mostrados en la Figura 3.10, la precisión que alcanza el modelo es del 65%.

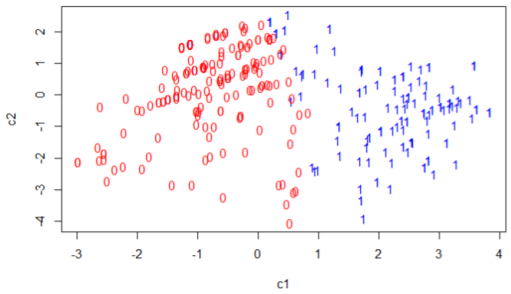

Figure 3.9: Clasificación Clasificador Lineal de Fisher con Undersampling.

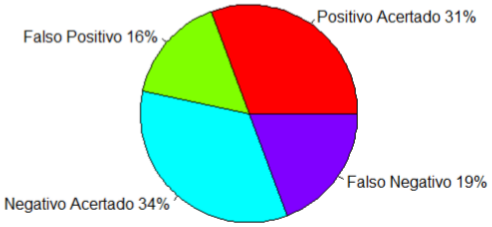

Figure 3.10: Resultados Clasificador Lineal de Fisher con Undersampling.

Chapter 4

Conclusiones

4.1 Discusión de Resultados

Hemos comprobado que en general todos los modelos superan el 60% de precisión, por lo que podemos asumir que todos ellos son usables. No obstante, el algoritmo de vecino más cercano a logrado obtener la precisión más elevada tanto en general como en particular, llegando a alcanzar el 93% de precisión, suficiente para que se pueda considerar como un modelo utilizable y útil.

Por otro lado, hemos visto que el discriminador Lineal de Fisher ofrece unos resultados ligeramente superiores al del Clasificador Ingenuo de Bayes, pero tampoco muy distante de sus valores, habiendo una diferencia menor del 10%, es por ello que podemos concluir que el algoritmo de Fisher ofrece mejores resultados que el de Bayes, pero no son lo suficientemente superiores como para considerarse uno de los algoritmos que mejor se adaptan al problema.

Se ha observado que en general el dataset donde se aplicaron técnicas de Oversampling han ofrecido mejores resultados que el de Undersampling. esto podemos explicarlo dado que no tenemos un volumen de datos excesivamente elevado, lo que hace que reducir la cantidad de datos a utilizar nos haga perder generalidad y por tanto información sobre nuestra población. Por ello, concluimos que en nuestro problema la técnica de Oversampling se adapta mejor que las de Undersampling, al hacer que nuestros datos crezcan (aunque sean duplicados) en lugar de reducirlos.

Por último, hemos sido capaces de alcanzar una alta precisión en la predicción de ofertas fraudulentas, lo que indica que las transformaciones de las variables

categóricas que hemos realizado han sido correctas (aunque puede que no las mejores o las más eficientes) que nos han permitido transformar en valores numéricos las variables categóricas para utilizarlas en nuestros modelos.

En resumen, hemos comprobado que el algoritmo del vecino más cercano en combinación con el dataset sobre el que se ha aplicado Oversampling ofrecen los mejores resultados y la precisión alcanzada es lo suficientemente buena como para poder defender las transformaciones sobre las variables categóricas realizadas.

4.2 Trabajo Futuro

En el estudio en el que nos hemos basado para seleccionar como referencia algunos algoritmos (2) se propusieron diferentes algoritmos con buenos resultados, como la familia de algoritmos de los árboles de decisiones. Dejamos para estudios posteriores el probar algoritmos de dicha familia para la elaboración de una solución más efectiva a nuestro problema.

Por otra parte, se han utilizado dos de los métodos más sencillos para resolver el problema de las clases desbalanceadas, como son el Oversampling y el Undersampling. Es conveniente investigar técnicas alternativas, como la elaboración de nuevos elementos del grupo minoritario o la fragmentación en más grupos de la clase mayoritaria.

Por último, algunas agrupaciones fueron realizadas en base a criterios poco rigurosos. La solución podría mejorar si se realizara un estudio sobre las variables y las agrupáramos siguiendo alguna metodología para que los grupos sean lo más representativos posibles de la información subyacente.

Chapter 5

Anexo

Hemos utilizado un grupo de ofertas de trabajo muy diferentes de varios países averiguando que los usuarios de los portales web en los que se publican las ofertas se equivocan rellenando los campos o los dejan en blanco a propósito. Si a esto le sumamos la poco homogeneidad de las respuestas (ya que el usuario puede introducir cualquier cosa no predeterminada para aspectos como el departamento, formación requerida o experiencia previa) hace que los datos sean más difíciles de trabajar e impiden que se puedan sacar todo el rendimiento que se debería.

No obstante y pese a ello, hemos conseguido elaborar un modelo capaz de detectar con un 93% de seguridad las ofertas fraudulentas que se suban a la web, lo que significa que disponemos de un modelo lo suficientemente bueno como para poder plantearnos utilizarlo en la práctica.

Es posible que se pudiera aumentar el rendimiento del modelo que ya hemos obtenido. Para ello, sería necesario contar con un especialista en finanzas que nos pueda ayudar a la clasificación y agrupación de los diferentes departamentos y funciones que se pueden desempeñar.

Cabe resaltar que el modelo ha sido diseñado de forma que no se necesita excesivo tiempo en poder determinar si una oferta es fraudulenta o no al momento de su publicación, por lo que esta solución es totalmente apta para implementarla en tiempo real, comprobando al momento de publicar la oferta si es o no fraudulenta.

Bibliography

[1] B. Alghamdi y F. Alharby, ”An Intelligent Model for Online Recruitment Fraud Detection”, J. Inf. Secur., 2019.

[2] Shawni Dutta y Samir Kumar Bandyopadhyay, ”Fake Job Recruitment Detection Using Machine Learning Approach”, International Journal of Engineering Trends and Technology(IJETT), 2020.

[3] Pouria Kaviani y Mrs. Sunita Dhotre, ”Short Survey on Naive Bayes Algorithm”, International Journal of Advance Engineering and Research Development, 2017.

[4] Nitin Bhatia y Vandana, ”Survey of Nearest Neighbor Techniques”, International Journal of Computer Science and Information Security, 2010.

[5] Xinjian Guo, Yilong Yin1, Cailing Dong, Gongping Yang y Guangtong Zhou. ”On the Class Imbalance Problem”. 2008.

[6] Yonatan Belinkov y James Glass, ”Analysis Methods in Neural Language Processing: A Survey”, MIT, 2019.