Rapport Projet Data/IA

Projet Maladies Cardiaques

Réaliser Par : Candice Giami, Joe Haifa, Tom Huget

FACULTÉ
GESTION, ÉCONOMIE,
SCIENCES
Université Catholique
de Lille 1875

Abstract :

Jour après jour, le nombre de cas de maladies cardiaques augmente à un rythme rapide, ce qui rend leur prédiction essentielle et préoccupante. Ce diagnostic est une tâche complexe qui doit être réalisée avec précision et efficacité. Ce rapport de recherche se concentre principalement sur l'identification des patients les plus susceptibles de développer une maladie cardiaque en fonction de divers attributs médicaux.

Nous avons conçu un système de prédiction des maladies cardiaques permettant de déterminer si un patient est susceptible d'être diagnostiqué avec une maladie cardiaque en utilisant son historique médical. Pour cela, nous avons appliqué différents algorithmes d'apprentissage automatique, tels que la Régression Logistique, KNN et Random Forest, afin de classifier les patients atteints de maladies cardiaques.

La robustesse du modèle proposé s'est révélée satisfaisante, permettant de détecter des signes de maladies cardiaques grâce aux algorithmes. Ainsi, ce modèle permet de réduire significativement l'incertitude dans l'identification des maladies cardiaques, en améliorant la fiabilité du diagnostic.

Le système de prédiction des maladies cardiaques proposé optimise les soins médicaux tout en réduisant les coûts. Ce projet apporte une connaissance précieuse pour aider à prédire les patients à risque et est implémenté au format pynb.

1. Introduction

« *L'apprentissage automatique est un moyen de manipuler et d'extraire des informations implicites, auparavant inconnues/connues et potentiellement utiles à partir des données* ».

L'apprentissage automatique est un domaine vaste et diversifié dont la portée et les applications augmentent chaque jour. Il intègre divers classificateurs issus de l'apprentissage supervisé, non supervisé et ensembliste, qui sont utilisés pour effectuer des prédictions et évaluer la précision des ensembles de données. Nous pouvons exploiter ces connaissances dans notre projet Maladies Cardiaques, qui pourrait bénéficier à de nombreuses personnes.

Les maladies cardiovasculaires sont très courantes de nos jours et regroupent un ensemble de pathologies pouvant affecter le cœur. L'Organisation Mondiale de la Santé (OMS) estime que 17,9 millions de décès dans le monde sont dus aux maladies cardiovasculaires (CVD), ce qui en fait la première cause de mortalité chez les adultes. Notre projet vise à prédire les individus susceptibles d'être diagnostiqués avec une maladie cardiaque en se basant sur leur historique médical. Il permet d'identifier les personnes présentant des symptômes tels que des douleurs thoraciques ou une hypertension artérielle et facilite le diagnostic avec moins d'examens médicaux et des traitements plus efficaces, optimisant ainsi leur prise en charge.

Ce projet repose principalement sur trois techniques d'exploration de données :

1. **Régression logistique**,
2. **KNN (K-Nearest Neighbors)**,
3. **Random Forest Classifier**.

Les données utilisées, contiennent des antécédents médicaux et des caractéristiques cliniques des patients. En analysant ces 12 attributs médicaux, nous pouvons prédire si un individu est susceptible d'être atteint d'une maladie cardiaque.

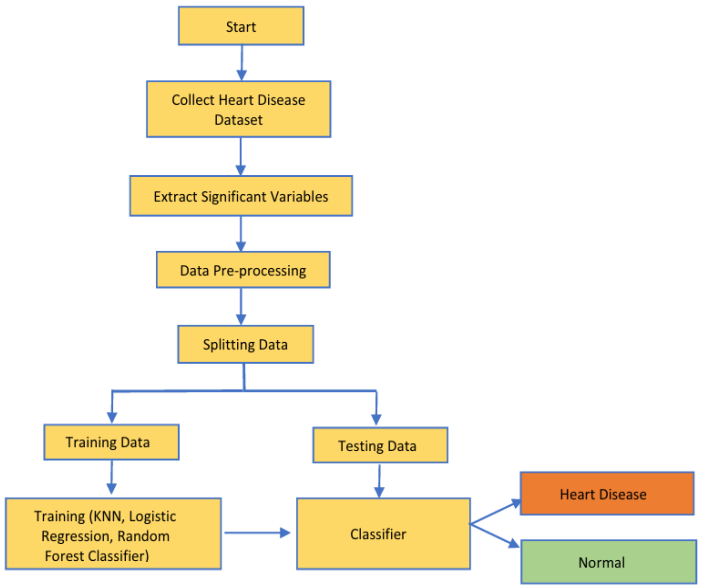
Enfin, notre modèle permet de classer les patients selon leur niveau de risque de développer une maladie cardiaque, tout en restant une solution **rentable et efficace**.

2. Méthodologie

La méthodologie est un processus qui comprend plusieurs étapes permettant de transformer des données brutes en modèles exploitables pour les utilisateurs. La **méthodologie proposée** (Figure 1) suit plusieurs étapes :

1. **Collecte des données** : La première étape consiste à rassembler les données pertinentes.
2. **Extraction des valeurs significatives** : Dans cette phase, nous identifions et sélectionnons les attributs médicaux pertinents.
3. **Prétraitement des données** : Cette étape implique le traitement des données, notamment la gestion des valeurs manquantes, le nettoyage des données et la normalisation en fonction des algorithmes utilisés.
4. **Classification des données** : Après le prétraitement, les données sont classifiées à l'aide des algorithmes **KNN, Régression Logistique et Random Forest**.
5. **Évaluation du modèle** : Le modèle proposé est ensuite testé et évalué en fonction de sa précision et de ses performances à l'aide de divers indicateurs de performance.

Dans ce modèle, un **Système de Prédiction des Maladies Cardiaques** a été développé en utilisant plusieurs classificateurs. Ce modèle exploite **12 paramètres médicaux**, tels que les douleurs thoraciques, le taux de sucre à jeun, la pression artérielle, le cholestérol, l'âge, le sexe, etc., pour effectuer ses prédictions.



3. Choix et décision

Dans cette section, nous présentons les principales décisions prises tout au long de notre projet, notamment lors du nettoyage des données et du choix des algorithmes. Ces choix ont été faits dans le but d'optimiser la qualité des données en entrée et de garantir de meilleures performances des modèles de machine learning.

Nettoyage et préparation des données

Pour commencer, nous avons renommé les colonnes du jeu de données afin de les rendre plus lisibles et explicites. Par exemple, « Age » est devenu « Âge », « Sex » est devenu « Sexe », et ainsi de suite. Cette étape, bien que purement esthétique, facilite grandement la compréhension du jeu de données, notamment lors de l'analyse ou de la présentation des résultats.

Nous avons ensuite analysé la présence de valeurs aberrantes à l'aide de boxplots. Dans le contexte médical, certaines valeurs extrêmes peuvent refléter une situation clinique réelle (par exemple une tension élevée chez un patient souffrant d'une maladie cardiaque). Toutefois, certaines valeurs étaient clairement invalides, comme une tension artérielle au repos égale à 0, ce qui n'est pas compatible avec la vie. Ces enregistrements ont donc été supprimés.

Concernant la variable « Cholestérol », nous avons identifié 171 valeurs égales à 0. Étant donné le nombre significatif de ces cas, nous avons opté pour une imputation par la médiane plutôt que la suppression, afin de conserver un maximum d'informations sans introduire de biais importants.

Par ailleurs, la variable « Sexe » a été transformée en une variable binaire : 1 pour les hommes et 0 pour les femmes. Cette transformation est couramment utilisée en machine learning, car elle permet aux algorithmes de traiter plus efficacement les variables catégorielles, en particulier lorsqu'il s'agit d'une variable binaire.

Pour les variables à plusieurs modalités telles que « Type_de_douleur_thoracique », « ECG_au_repos » et « Pente_ST », nous avons opté pour un encodage *one-hot* (encodage fictif), c'est-à-dire la création d'une colonne binaire pour chaque modalité. Cette technique évite à l'algorithme d'interpréter une relation d'ordre entre des catégories qui n'en ont pas, et améliore ainsi la performance des modèles.

4. Choix des algorithmes de classification

Pour ce projet de prédiction de la maladie cardiaque, nous avons sélectionné trois algorithmes de classification supervisée : **la régression logistique**, **le K-Nearest Neighbors (KNN)** et **le Random Forest**. Chacun de ces algorithmes présente des caractéristiques uniques qui en font des choix pertinents dans le cadre d'une tâche de classification binaire avec des données médicales structurées. Ci-dessous, nous détaillons les raisons spécifiques qui nous ont poussés à les choisir.

1. Régression logistique

La régression logistique est souvent utilisée comme point de départ dans les problèmes de classification binaire, en raison de sa simplicité, de son efficacité et de son interprétabilité. Dans notre projet, son choix se justifie par plusieurs raisons :

- **Modèle linéaire explicatif** : La régression logistique permet non seulement de prédire une probabilité d'appartenance à une classe (ici, malade ou non malade), mais aussi d'interpréter l'effet de chaque variable indépendante sur le résultat. Cela est particulièrement important dans un contexte médical, où la compréhension des facteurs de risque est aussi cruciale que la précision du modèle.
- **Rapide à entraîner** : Contrairement à des modèles plus complexes, la régression logistique s'entraîne rapidement, même sur des ensembles de données de taille modérée. Cela nous a permis d'effectuer des ajustements rapides et de tester diverses configurations sans surcharger les ressources.
- **Moins sensible au surapprentissage** : En particulier lorsque les données ont été correctement nettoyées et que les variables pertinentes ont été sélectionnées, la régression logistique tend à généraliser correctement sans surapprentissage excessif.
- **Bonne base de comparaison** : Elle nous a également servi de modèle de référence (« baseline ») pour comparer les performances d'autres algorithmes plus complexes. Si un modèle ne fait pas mieux que la régression logistique, cela signifie généralement qu'il n'apporte pas une réelle valeur ajoutée.

2. K-Nearest Neighbors (KNN)

L'algorithme des k-plus proches voisins est un modèle non paramétrique, simple mais puissant, particulièrement adapté aux problèmes de classification dans un espace de caractéristiques bien défini.

- **Principe intuitif et compréhensible** : KNN repose sur une idée intuitive : un individu est classé en fonction de la majorité des catégories de ses voisins les plus proches. Dans un domaine aussi sensible que la médecine, cette logique est particulièrement claire et peut même s'apparenter à une prise de décision clinique, basée sur la similarité avec d'autres cas connus.
- **Aucune hypothèse sur la distribution des données** : Contrairement à la régression logistique, qui suppose une relation linéaire entre les variables indépendantes et le logarithme de la probabilité, KNN ne fait aucune hypothèse particulière. Cela lui permet de s'adapter à des structures complexes dans les données, ce qui est souvent le cas en médecine.
- **Bon comportement dans des datasets équilibrés** : Notre jeu de données comportant un nombre relativement équilibré de cas positifs et négatifs (maladie vs. non maladie), KNN peut fonctionner efficacement sans être biaisé vers une classe majoritaire.
- **Robustesse à la redondance** : KNN peut gérer relativement bien des données où certaines variables peuvent être partiellement redondantes, surtout après encodage one-hot. Cela a été pertinent dans notre cas où plusieurs variables catégorielles ont été encodées en plusieurs colonnes.

3. Random Forest

Le modèle Random Forest est un algorithme d'ensemble basé sur des arbres de décision, qui combine puissance prédictive et robustesse. Son choix a été motivé par plusieurs avantages importants dans le contexte de notre projet :

- **Capacité à modéliser des relations non linéaires complexes** : Les arbres de décision sont capables de capturer des interactions subtiles entre les variables, et Random Forest, en combinant plusieurs arbres, permet d'augmenter la précision tout en réduisant le risque de surapprentissage.
- **Tolérance au bruit et aux valeurs aberrantes** : Grâce à sa structure en forêts (plusieurs arbres qui votent), l'algorithme est naturellement robuste face aux données bruitées et aux valeurs extrêmes, ce qui est un avantage considérable dans un domaine aussi sensible que les données cliniques.
- **Estimation de l'importance des variables** : L'un des atouts majeurs du Random Forest est sa capacité à fournir une estimation de l'importance de chaque variable dans la prédiction. Cela nous a permis non seulement de mieux comprendre le rôle des différents facteurs de risque, mais aussi d'identifier les variables peu pertinentes à supprimer, améliorant ainsi la performance globale du modèle.
- **Moins de réglages hyperparamétriques critiques** : Bien que Random Forest dispose de plusieurs paramètres, il offre souvent de bons résultats par défaut, ce qui le rend plus facile à utiliser pour un premier modèle robuste.
- **Bonne performance même sur des datasets de taille moyenne** : Contrairement à certains modèles qui nécessitent de très grandes quantités de données pour bien fonctionner, Random Forest s'adapte bien aux tailles de datasets comme le nôtre.

En résumé, notre stratégie consistait à combiner **un modèle simple et interprétable (régression logistique)**, **un modèle basé sur la similarité locale (KNN)**, et **un modèle puissant basé sur l'agrégation d'arbres (Random Forest)**. Cette diversité nous a permis de comparer les performances de plusieurs approches complémentaires, afin d'identifier celle qui offre le meilleur compromis entre précision, robustesse, et interprétabilité.

5. Fonctionnement des algorithmes

Dans cette section, nous décrivons le mécanisme de fonctionnement des trois algorithmes que nous avons utilisés pour la classification des patients en fonction de la présence ou non d'une maladie cardiaque. Ces explications permettent de mieux comprendre comment chaque modèle traite les données pour effectuer des prédictions.

1. Régression Logistique

La régression logistique est un algorithme de classification binaire qui permet d'estimer la probabilité qu'une observation appartienne à une classe donnée. Bien qu'elle porte le nom de « régression », il s'agit bien d'un algorithme de classification.

- **Principe de base** :
  L'algorithme cherche à modéliser la relation entre les variables indépendantes (par exemple : l'âge, le cholestérol, etc.) et une variable cible binaire (présence ou absence de la maladie) en utilisant une **fonction logistique (ou sigmoïde)**. Cette fonction transforme une combinaison linéaire des variables en une valeur comprise entre 0 et 1, interprétée comme une **probabilité**.

- **Fonction sigmoïde** :

  $\sigma(z) = 1 / (1 + e^{(-z)})$,
  où **$z = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + ... + wn \cdot xn$**,

  Avec **w** les poids à apprendre, et **x** les variables d'entrée.**Seuil de décision** :
  Si la probabilité prédite est supérieure à 0.5 (ou un autre seuil), l'observation est classée dans la classe positive (malade), sinon dans la classe négative (non malade).

- **Apprentissage des poids** :
  Le modèle ajuste les coefficients (poids) associés à chaque variable grâce à un processus d'optimisation (descente de gradient) afin de minimiser une fonction de coût appelée **log-loss**, qui pénalise les erreurs de prédiction.
- **Interprétabilité** :
  Un grand avantage de la régression logistique est que les coefficients appris peuvent être interprétés comme l'influence de chaque variable sur la probabilité de maladie.

2. K-Nearest Neighbors (KNN)

KNN est un algorithme **non paramétrique** et **à base de distance**, qui ne construit pas de modèle explicite pendant la phase d'apprentissage. Il se base uniquement sur les données d'entraînement pour prendre une décision lors de la phase de prédiction.

- **Principe de base** :
  Lorsqu'un nouvel échantillon doit être classé, l'algorithme calcule la distance entre cet échantillon et **tous les exemples du jeu d'entraînement**. Il sélectionne ensuite les **K observations les plus proches** (appelées « voisins »).
- **Distance utilisée** :
  La distance la plus courante est la distance euclidienne, mais d'autres métriques (manhattan, minkowski, etc.) peuvent être utilisées selon le contexte.
- **Vote majoritaire** :
  Une fois les K voisins identifiés, l'algorithme regarde à quelle classe appartiennent ces voisins. La **classe majoritaire** parmi eux est alors attribuée à la nouvelle observation.
- **Paramètre K** :
  Le choix de la valeur de K est crucial :
  - Un K trop petit rend le modèle sensible au bruit.
  - Un K trop grand peut noyer l'effet local et rendre la prédiction moins précise.
- **Besoin de normalisation** :
  Comme KNN se base sur les distances, les variables doivent être **normalisées** (mises à

la même échelle), sinon les variables à grande échelle domineront la mesure de distance.

- **Pas d'apprentissage à proprement parler** :
  Contrairement aux autres modèles, KNN ne fait **aucune généralisation** pendant la phase d'entraînement. Il garde simplement les données en mémoire. C'est lors de la prédiction que le calcul est effectué.

3. Random Forest

Random Forest est un algorithme **d'ensemble** basé sur la combinaison de **plusieurs arbres de décision**. Il fait partie de la famille des modèles dits « bagging » (Bootstrap Aggregating), qui visent à améliorer la stabilité et la performance en agrégeant plusieurs modèles faibles.

- **Principe de base** :
  L'idée est de créer une **forêt** composée de plusieurs arbres de décision, chacun entraîné sur un **sous-échantillon** aléatoire du jeu de données d'entraînement. Lorsqu'une prédiction est demandée, **chaque arbre donne un vote**, et la **classe majoritaire** est choisie comme prédiction finale.
- **Création des arbres** :
  Chaque arbre est construit en :
  - Choisissant un échantillon aléatoire (avec remise) du dataset.
  - À chaque nœud, en ne considérant qu'un **sous-ensemble aléatoire de variables** pour déterminer la meilleure séparation. Cela réduit la corrélation entre les arbres et améliore la diversité du modèle.
- **Avantages** :
  - **Réduction du surapprentissage** par rapport à un arbre unique.
  - **Robustesse** face aux données bruitées ou aux valeurs aberrantes.
  - **Capacité à estimer l'importance des variables** (ce que nous avons exploité dans notre projet pour faire du feature selection).
- **Décision finale** :
  En classification, chaque arbre produit une prédiction (0 ou 1 dans notre cas), et la forêt choisit la classe ayant obtenu le plus de votes.
- **Complexité maîtrisée** :
  Bien que chaque arbre soit relativement simple, la combinaison de nombreux arbres permet d'obtenir un modèle très performant, sans qu'il soit nécessaire de comprendre chaque arbre individuellement.

Conclusion

Chaque algorithme fonctionne selon une logique différente, ce qui nous permet d'avoir une vision complète des forces et limites de chacun dans le contexte du diagnostic de maladie cardiaque. La **régression logistique** mise sur l'interprétabilité et la simplicité, **KNN** sur la proximité locale dans l'espace des caractéristiques, et **Random Forest** sur la puissance d'un ensemble de modèles hétérogènes.

Résultats obtenus

L'objectif de cette partie est de comparer les performances de trois modèles de classification supervisée – **K-Nearest Neighbors (KNN)**, **Random Forest (RF)** et **Régression Logistique (RL)** – pour prédire la présence d'une maladie cardiaque à partir des caractéristiques cliniques de 100 patients. Les résultats ont été compilés dans un tableau de comparaison qui indique, pour chaque patient, si la prédiction de chaque algorithme était correcte (1) ou incorrecte (0).(figure)

Méthodologie d'évaluation

Pour chaque modèle :

- Le **seuil de classification** a été fixé à 0.5 pour les modèles probabilistes (RF, RL), ce qui signifie que toute probabilité ≥ 0.5 est considérée comme une prédiction positive (malade).
- Le modèle a été **entraîné et testé sur les mêmes 100 patients** dans un objectif de comparaison directe.
- La **métrique utilisée** ici est l'exactitude (accuracy) :

  Accuracy = Nombre de bonnes prédictions / Nombre total de prédictions

Résultats chiffrés

Modèle	Bonnes prédictions	Accuracy (%)
K-Nearest Neighbors	86 / 100	86.00 %
Random Forest	85 / 100	85.00 %
Régression Logistique	86 / 100	86.00 %

Analyse comparative

1. K-Nearest Neighbors (KNN)

- **Performance brute** : 86 % de bonnes prédictions, soit l'un des meilleurs résultats du test.

  Avantages :

  - Méthode non paramétrique, donc utile en présence de données non linéaires.
  - Facile à implémenter, interprétation intuitive.

  Limites :

  - Sensible au **choix de k,** et à la **mise à l'échelle des données** (Standardisation).

- Peut sur-apprendre (overfitting) si le bruit est important, ou mal se généraliser si k est trop petit.
- Peu efficace en cas de données très déséquilibrées.

2. Random Forest

- **Performance brute** : 85 % de bonnes prédictions.

Avantages :

- Robuste aux variables bruitées et aux corrélations, grâce à l'agrégation de plusieurs arbres de décision.
- Permet une estimation de l'importance des variables (non exploitée ici mais intéressante dans le domaine médical).

Limites :

- Légère sous-performance ici comparée à KNN et RL.
- Résultats parfois biaisés si certaines classes sont surreprésentées.
- Moins transparent/interprétable qu'un modèle linéaire dans un cadre médical.

3. Régression Logistique

- **Performance brute** : 86 % également, au même niveau que KNN.

Avantages :

- Simplicité, interprétabilité des coefficients (utile en contexte clinique pour expliquer les décisions).
- Moins de risque de surapprentissage si le nombre de features est maîtrisé.

Limites :

- Suppose une **relation linéaire** entre les variables explicatives et le log-odds de la sortie → possible perte d'information si cette hypothèse est violée.
- Moins performant que les modèles non linéaires dans certains cas complexes.

Interprétation et choix du modèle

Même si KNN et la Régression Logistique affichent la même **exactitude de 86%**, le **choix du meilleur modèle dépend du contexte d'application** :

- Si l'on privilégie la **performance pure**, **KNN** peut être retenu.
- Si l'on privilégie l'**interprétabilité et la rigueur clinique**, la **régression logistique** est préférable.
- Si l'on vise un **compromis robuste sur différents types de données**, **Random Forest** reste un bon candidat, malgré 1% de moins en précision.

Application interactive : rendre le modèle utile et accessible

Afin de donner une dimension concrète et accessible à notre projet, nous avons conçu une application Python et un site web interactif permettant aux utilisateurs — qu'il s'agisse de patients à risque ou de professionnels de santé — d'estimer leur probabilité d'être atteints d'une maladie cardiaque à partir de leurs données médicales personnelles.

Ce dispositif vise à :

- Sensibiliser les utilisateurs présentant des facteurs de risque,
- Encourager la vigilance et les consultations préventives,
- Faciliter l'accès au modèle prédictif, sans nécessiter de compétences techniques.

L'interface a été pensée pour être simple et intuitive, avec des champs cliniques classiques à remplir (âge, tension, cholestérol, etc.), un bouton de prédiction, et un affichage immédiat du résultat sous forme de probabilité et de message d'alerte si le risque est élevé.

Tableau de résultats :

	Vraie_classe	KNN_Prediction	KNN_Prob_Sain	KNN_Prob_Malade	RF_Prediction	RF_Prob_Sain	RF_Prob_Malade	RL_Prediction	RL_Prob_Sain	RL_Prob_Malade
669	0.0	0.0	80.3921568627451	19.607843137254903	0.0	64.0	36.0	0.0	89.08793239810345	10.912067601896563
30	1.0	1.0	45.098039215686278	54.90196078431373	1.0	42.0	57.99999999999999	0.0	57.10863945890434	42.89136054109565
377	1.0	1.0	1.9607843137254901	98.0392156862745	1.0	20.0	80.0	1.0	4.179883949070044	95.82011605092956
536	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	24.0	76.0	1.0	7.742860190492906	92.2571398095071
807	0.0	0.0	100.0	0.0	0.0	92.0	8.0	0.0	94.19903750679991	5.800962493200091
793	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	18.0	82.0	1.0	8.498341035800328	91.50165896419968
363	1.0	1.0	15.686274509803921	84.31372549019608	1.0	26.0	74.0	1.0	20.023517129354694	79.9764828706453
584	1.0	1.0	1.9607843137254901	98.0392156862745	1.0	21.0	79.0	1.0	1.8191178058145852	98.18088219418541
165	1.0	1.0	37.2549019607843 16	62.745098039215684	1.0	39.0	61.0	1.0	20.00561327709952	79.99438672290047
484	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	25.0	75.0	1.0	5.178808196953877	94.82119180304612
774	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	22.0	78.0	1.0	9.337641735490132	90.66235826450986
552	1.0	1.0	17.647058823529413	82.35294117647058	1.0	34.0	66.0	1.0	17.058197373844575	82.94180267615542
769	0.0	0.0	92.15686274509804	7.8431372549019605	0.0	72.0	28.000000000000004	0.0	97.82845526453823	2.17154473546417745
695	1.0	1.0	5.88235294117647	94.11764705882352	1.0	23.0	77.0	1.0	8.26044123517221	91.73955876482779
719	1.0	1.0	13.725490196078432	86.27450980392157	1.0	28.999999999999996	71.0	1.0	14.209167613741746	85.79083238625826
312	1.0	0.0	50.98039215686274	49.01960784313725	0.0	56.99999999999999	43.0	1.0	48.410972271452266	51.589027728547734
714	0.0	0.0	98.0392156862745	1.9607843137254901	0.0	97.0	3.0	0.0	97.79960513471798	2.200394865282018
309	1.0	1.0	21.568627450980394	78.43137254901961	1.0	32.0	68.0	1.0	23.159230119755236	76.84076988024476
846	1.0	1.0	15.686274509803921	84.31372549019608	1.0	21.0	79.0	1.0	18.58175113402185	81.41824886597814
617	1.0	0.0	100.0	0.0	0.0	92.0	8.0	0.0	93.07602462058162	6.923975379418383
355	1.0	1.0	39.21568627450981	60.78431372549019	1.0	49.0	51.0	1.0	47.64674696774237	57.35325303225763
39	0.0	0.0	11.764705882352941	88.23529411764706	1.0	28.999999999999996	71.0	1.0	23.254101654227767	76.74589834577223
231	0.0	0.0	92.15686274509804	7.8431372549019605	0.0	79.0	21.0	0.0	94.24896933930597	5.751030660694021
867	1.0	0.0	64.70588235294117	35.294117647058826	1.0	39.0	61.0	0.0	76.37482227949543	23.625177720504567
604	0.0	0.0	62.745098039215684	37.254901960784316	0.0	75.0	25.0	0.0	58.900107652292874	41.099892347707126
63	1.0	1.0	9.803921568627452	90.19607843137256	1.0	25.0	75.0	1.0	7.651911038366199	92.3480889616338
192	0.0	0.0	100.0	0.0	0.0	92.0	8.0	0.0	94.55438817320413	5.445611826795 87
482	1.0	0.0	50.98039215686274	49.01960784313725	0.0	65.0	35.0	0.0	54.063388183847777	45.936611816152222
866	0.0	0.0	88.23529411764706	11.76470588235294	0.0	74.0	26.0	0.0	95.57378703577552	4.426212964224473
67	0.0	0.0	98.0392156862745	1.9607843137254901	0.0	92.0	8.0	0.0	96.30138370061047	3.6986162993895366
72	1.0	1.0	19.607843137254903	80.3921568627451	1.0	21.0	79.0	1.0	22.234152440571254	77.76584755942875
656	0.0	0.0	80.3921568627451	19.607843137254903	0.0	76.0	24.0	0.0	90.53222778499143	9.467772215008562
680	1.0	1.0	11.764705882352941	88.23529411764706	1.0	27.0	73.0	1.0	8.368510408483354	91.63148959151664
139	1.0	1.0	5.88235294117647	94.11764705882352	1.0	28.999999999999996	71.0	1.0	5.5862165724633766	94.41378342753663
733	0.0	1.0	45.098039215686278	54.90196078431373	0.0	52.0	48.0	0.0	75.28488019741168	24.715119802588333
824	0.0	0.0	54.901960784313725	45.098039215686278	1.0	37.0	63.0	0.0	61.698195879225295	38.301804120774705
174	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	28.999999999999996	71.0	1.0	5.105800168051222	94.89419983194878
896	0.0	0.0	92.15686274509804	7.8431372549019605	0.0	79.0	21.0	0.0	93.99313897144901	6.006861028550976 5
500	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	25.0	75.0	1.0	4.873515377980 28664	95.12648462201972
70	1.0	1.0	23.529411764705 88	76.47058823529412	1.0	36.0	64.0	1.0	28.502995386736508	71.4970046132635
717	0.0	0.0	80.3921568627451	19.607843137254903	0.0	76.0	24.0	0.0	92.41194953903147	7.588050460968529
23	1.0	1.0	11.764705882352941	88.23529411764706	1.0	33.0	67.0	1.0	20.654784911426556	79.34521508857344
542	1.0	1.0	9.803921568627452	90.19607843137256	1.0	28.999999999999996	71.0	1.0	19.460623773863 24	80.53937622613677
799	0.0	0.0	78.43137254901961	21.568627450980394	0.0	56.00000000000001	44.0	0.0	82.69244265599228	17.307557344007 72
673	1.0	1.0	23.529411764705 88	76.47058823529412	1.0	28.999999999999996	71.0	1.0	29.251266277825206	70.74873372217479
826	0.0	1.0	35.294117647058826	64.70588235294117	1.0	44.0	56.00000000000001	1.0	19.138643018580726	80.86135698141928
250	1.0	1.0	31.372549019607842	68.62745098039215	1.0	21.0	79.0	1.0	26.56478763022624	73.43521236977377
753	0.0	0.0	100.0	0.0	0.0	95.0	5.0	0.0	99.01405876062881	0.9859412393711936
350	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	18.0	82.0	1.0	7.841979892193307	92.15802010780669
759	1.0	0.0	88.23529411764706	11.76470588235294	0.0	74.0	26.0	0.0	95.20451461310661	4.795485386893396
760	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	28.999999999999996	71.0	1.0	3.964914823696064	96.03508517630394
107	0.0	0.0	98.0392156862745	1.9607843137254901	0.0	84.0	16.0	0.0	95.2605689966283	4.739431003371703
445	1.0	1.0	21.568627450980394	78.43137254901961	1.0	31.0	69.0	1.0	21.208987174446182	78.79101282555382
141	1.0	1.0	5.88235294117647	94.11764705882352	1.0	31.0	69.0	1.0	4.761322229093512	95.23867777090649
651	1.0	1.0	25.490196078431 37	74.50980392156863	1.0	47.0	53.0	1.0	37.9981644985260 66	62.001835501473934
545	0.0	0.0	74.50980392156863	25.490196078431 37	0.0	70.0	30.0	0.0	71.43423054541825	28.565769454581 75
110	0.0	0.0	86.27450980392157	13.725490196078432	0.0	59.0	41.0	0.0	75.63883184034816	24.361168159651836
594	1.0	1.0	21.568627450980394	78.43137254901961	0.0	53.0	47.0	1.0	16.564016181974807	83.43598381802519
520	0.0	1.0	13.725490196078432	86.27450980392157	1.0	26.0	74.0	1.0	15.987266321798831	84.01273678201 17
907	1.0	1.0	9.803921568627452	90.19607843137256	1.0	27.0	73.0	1.0	13.031205002441236	86.96879499755876
676	1.0	1.0	15.686274509803921	84.31372549019608	1.0	28.999999999999996	71.0	1.0	25.052346223136713	74.94765377686329
280	0.0	0.0	80.3921568627451	19.607843137254903	0.0	52.0	48.0	0.0	90.28544438776252	9.714555612237476
136	0.0	0.0	100.0	0.0	0.0	89.0	11.0	0.0	98.60791307990486	1.3920869200951356
422	1.0	1.0	0.0	100.0	1.0	21.0	79.0	1.0	2.4061509624788924	97.59384903752111
208	0.0	0.0	90.19607843137256	9.803921568627452	0.0	74.0	26.0	0.0	96.223979107066538	3.776020892946284
442	1.0	1.0	3.9215686274509802	96.07843137254902	1.0	21.0	79.0	1.0	2.352889754547949	97.64711024545205
86	1.0	1.0	1.9607843137254901	98.0392156862745	1.0	23.0	77.0	1.0	1.315734670831 0813	98.68426532916892
44	1.0	1.0	5.88235294117647	94.11764705882352	1.0	25.0	75.0	1.0	7.237729747167487	92.76227025283251
532	1.0	1.0	5.88235294117647	94.11764705882352	1.0	20.0	80.0	1.0	6.014075245519923	93.98592475448008
913	1.0	1.0	35.294117647058826	64.70588235294117	1.0	41.0	59.0	1.0	43.93184357773029	56.06815642226971
635	1.0	1.0	9.803921568627452	90.19607843137256	1.0	31.0	69.0	1.0	3.822851951245731 5	96.17714804875426
290	0.0	0.0	90.19607843137256	9.803921568627452	0.0	79.0	21.0	0.0	91.92132440302547	8.078675596974534
338	1.0	1.0	35.294117647058826	64.70588235294117	1.0	41.0	59.0	1.0	26.951088698702673	73.04891130129732
357	1.0	0.0	64.70588235294117	35.294117647058826	1.0	47.0	53.0	0.0	52.78700196067702 4	47.212998039322976
292	0.0	0.0	68.627450980392 15	31.372549019607842	0.0	69.0	31.0	0.0	71.781122462203 65	28.218877537790355
227	1.0	1.0	9.803921568627452	90.19607843137256	1.0	28.999999999999996	71.0	1.0	5.765522424039 86	94.23447757596014
592	1.0	1.0	9.803921568627452	90.19607843137256	1.0	26.0	74.0	1.0	4.272067068834665	95.72793293111563
425	1.0	1.0	31.372549019607842	68.62745098039215	1.0	38.0	62.0	1.0	33.355362900392 33	66.64463709960768
789	0.0	0.0	66.66666666666666	33.333333333333333	0.0	51.0	49.0	0.0	93.93193428504203	6.068065714957973
523	1.0	1.0	7.8431372549019605	92.15686274509804	1.0	25.0	75.0	1.0	5.5657026722888908	94.43429732771109
915	1.0	1.0	9.803921568627452	90.19607843137256	1.0	25.0	75.0	1.0	5.537173339195844	94.46287666080415
352	1.0	1.0	17.647058823529413	82.35294117647058	1.0	31.0	69.0	1.0	21.806339621761637	78.19366037823836
494	1.0	1.0	9.803921568627452	90.19607843137256	1.0	28.999999999999996	71.0	1.0	3.744308451468969 5	96.25569154853103
60	0.0	0.0	100.0	0.0	0.0	92.0	8.0	0.0	95.03181654928888	4.968183450711 12
599	1.0	1.0	1.9607843137254901	98.0392156862745	1.0	18.0	82.0	1.0	4.41874272772268	95.58125727227733
723	1.0	0.0	54.901960784313725	45.098039215686278	0.0	50.0	50.0	0.0	52.16738789914348	47.832612100856515
426	0.0	0.0	96.078431372549 02	3.9215686274509802	0.0	84.0	16.0	0.0	92.00335311029066	7.996646889709333
469	0.0	1.0	47.058823529411 76	52.941176470588 24	1.0	42.0	57.99999999999999	1.0	21.501978198707526	78.49802180129247
66	0.0	0.0	82.352941176470 58	17.647058823529413	0.0	76.0	24.0	0.0	92.062140954429 89	7.937859045570102
332	0.0	0.0	92.156862745098 04	7.8431372549019605	0.0	79.0	21.0	0.0	96.35471950685 66	3.645280493143394
375	1.0	0.0	80.392156862745 1	19.607843137254903	0.0	79.0	21.0	0.0	95.97659373614427	4.023406263855 73
381	1.0	1.0	9.803921568627452	90.19607843137256	1.0	25.0	75.0	1.0	9.726104370864174	90.27389562913582
558	1.0	1.0	19.607843137254903	80.3921568627451	1.0	31.0	69.0	1.0	20.482881903881 6	79.51711809611 84
260	0.0	0.0	98.0392156862745	1.9607843137254901	0.0	84.0	16.0	0.0	89.03885738719222	10.961142612807787
235	0.0	0.0	62.745098039215684	37.254901960784316	1.0	35.0	65.0	1.0	38.204185441553854	61.795814558446146
218	0.0	0.0	100.0	0.0	0.0	92.0	8.0	0.0	93.33351701907655	6.666482980923445
382	1.0	1.0	9.803921568627452	90.19607843137256	1.0	28.999999999999996	71.0	1.0	6.184827202354737	93.81517279764526
439	0.0	0.0	82.352941176470 58	17.647058823529413	0.0	79.0	21.0	0.0	86.83725548112639	13.162744518873607
763	1.0	0.0	70.588235294117 65	29.411764705882355	0.0	54.0	46.0	0.0	79.32001508861018	20.679984991 38982
305	1.0	1.0	35.294117647058826	64.70588235294117	0.0	60.0	40.0	1.0	48.62919318174952	51.37080681825048