Beenamics

Stratégie météo

et

Equipe 4 :

BOUGEROLLES Jean
CHOMEL Louis
DENIAU Pierre
GOURC Mathis
LABAT Valentin
ROYER Romain

Référents :

BARRAU Nelly
LOUBIERE Peio

Objectifs

Les objectifs sont d'implémenter dans le modèle HOPOMO les variables météorologiques suivantes :

- La température
- La pluie
- Le vent
- L'humidité

Pour cela nous avons utilisé les équations déjà existantes pour la pluie et la température. Puis nous les avons adaptées aux données d'entrées des API (Astral et OpenWeatherMap) et ajouté les variables de vent et d'humidité.

Les entrées

Pour réaliser la prédiction suivant le modèle nous utilisons l'API **OpenWeatherMap** qui nous permet d'obtenir les prédictions météorologiques sur les 5 prochains jours.
Nous utilisons également **Astral** qui nous permet d'obtenir l'heure exacte du lever et du coucher de soleil pour chaque jour.

Cependant, ces données existent aussi dans les exports OWM, mais pour tester notre modèle nous avions besoin de récupérer toutes les données d'une année passée, ce qui est payant sur cette dernière API.

De même pour tester notre modèle nous avions besoin de données complètes d'une année passée. C'est pourquoi nous avons utilisé l'export csv fourni par Lorenzo, lui-même exporté via OWM. Nous avons ainsi récupéré les données météorologiques de 2020 en format "heure par heure" (*meteo_gelos_owm_2020.csv*).

Traitement des entrées

Nous avons donc travaillé sur le modèle météorologique à travers 4 indicateurs différents : la température, la pluviométrie, l'humidité et le vent. Pour tester ces indicateurs, que nous expliquerons ci-dessous, il était nécessaire de retraiter les entrées de la façon suivante :

- Récuper des variables du fichier csv (*meteo_gelos_owm_2020.csv)*
  `'date','main.temp','rain.1h','main.humidity','wind.speed'`
- Convertir de la colonne date/heure en deux listes composés d'entiers.
  - Il existe une méthode datetime pour effectuer cette opération réalisée pour le moment naïvement.
- Exporter les données de lever et coucher de soleil sur toute une année puis les convertir en une liste de liste [[date, heure du lever, heure du coucher]]
- Implémenter deux fonctions utilitaires pour : convertir le jour (int) en une date (liste d'entiers) et récupérer uniquement les données comprises entre le lever et le coucher du soleil (retourne une liste).

Le traitement des données de prédictions (qui sont en format JSON) se fera ultérieurement. Néanmoins les variables et la méthode resteront identiques. En effet, il suffira de transformer les données JSON en format CSV.

Implémentation

L'implémentation est actuellement adaptée à des données heure par heure. Cependant, pour la prédiction, les données seront toutes les 3h. Ce qui ne pose pas de problème puisqu'il suffira d'adapter le coefficient des indicateurs. Nous aurons donc logiquement des résultats moins discrets.

Température

Après avoir récupéré les données de température avec la fonction **TEMP(t)** nous calculons un index de la manière suivante :

- Calculer l'index d'une valeur (pour 1h et plus tard pour 3h)
- Ajouter l'index dans une liste.
- Sommer tous les index calculés
- Diviser le tout par le nombre d'heures d'ensoleillement. Donc le nombre de données de température pour des valeurs toutes les heures (qui sera multiplié par 3 pour la prédiction).

Nous avons par ailleurs fait le choix de diviser le résultat par le nombre d'heures d'ensoleillement où nous possédons les données (plutôt que d'utiliser le résultat d'une soustraction entre l'heure du coucher et du lever du soleil via Astral). Pour la raison assez logique d'obtenir des résultats les moins biaisés possible (s'il n'y a pas le même nombre de données par jour, ce qui est le cas).

Ainsi on calcul l'index d'une valeur en suivant les valeurs précédemment données dans le modèle HOPOMO, à savoir :

- En dessous de 14 degrés les abeilles ne volent pas, index à 0
- Entre 14 et 22 degrés on calcul un index entre 0 et 1
    - Normalisation de la valeur : (valeur - min)/(max-min)
- Entre 22 et 32 l'index est optimal est vaut 1
- Entre 32 et 40 on calcul un index entre 0 et 1
    - Normalisation de la valeur : (max - valeur)/(max-min)
- Au dessus de 40 degrés les abeilles ne volent pas, index à 0

Pluie

Concernant la pluie, nous avons le nombre de millimètres d'eau tombé par heure. Ainsi nous pouvons calculer le nombre d'heures où il a plu avec la fonction **HOURSraining_during_daylight(t).**

Celle-ci récupère également les données de pluviométrie. Ensuite nous avons besoin des heures d'ensoleillement de la fonction **HOURSdaylight(t).** Comme expliqué précédemment on retourne simplement le nombre de données pour avoir le nombre d'heures d'ensoleillement (puisqu'on a déjà récupéré les données uniquement entre le lever et le coucher du soleil).

On obtient ainsi le taux de pluie sur la journée en divisant le nombre d'heures de pluie pour le jour t par le nombre d'heures d'ensoleillement du même jour.

Finalement on obtient la normalisation inverse en soustrayant à 1 ce taux. Plus cet index final sera proche de 1, plus les abeilles n'auront pas été impactées par la pluie. On obtient le résultat inverse si cet index est proche de 0.

Humidité

Sur le même principe de la température, la fonction **INDEXhumidity(t)** récupère les données d'humidité puis réalise le même traitement : normalisation des valeurs, puis somme des index divisée par le nombre d'heures (de données). Ainsi on obtient un indicateur sur la journée.

L'humidité minimum requise est de 60%, ainsi nous avons créé une nouvelle variable dans les paramètres appelée **min_humidity** initialisée à 60 et utilisée de la manière suivante :

- Entre 60% et 80% d'humidité, l'index est optimal, fixé à 1
- Entre 80% et 100% d'humidité, l'index est une normalisation inverse de la valeur (plus on se rapproche de 100% plus l'index sera petit)
- En dessous de 60% les abeilles sont fortement affectées, l'index est égal à 0

Une étude a montré qu'un taux d'humidité trop élevé affecte négativement la qualité et l'obtention de nectar (Silva et al., 2013; Alves et al., 2015). C'est pourquoi nous effectuons une normalisation inverse entre 80% et 100% d'humidité (sachant qu'avec 100% d'humidité il pleut probablement).

Cependant, cet index influence trop fortement (et négativement) les abeilles dans notre modèle. C'est pourquoi nous avons temporairement mis de côté cette variable pour nos sorties graphiques afin de réfléchir à une autre solution d'implémentation (probablement plus sur les fonctions de récolte du nectar que sur le vol en lui-même des abeilles).

Vent

Enfin, la fonction **INDEXwind(t)** réalise exactement les mêmes actions que pour l'humidité. Cette fois-ci nous avons créé une variable appelée **max_wind,** fixée à 45km/h.

- Au dessous de 45km/h, l'index est une normalisation inverse de la valeur (plus on est proche de 0 km/h plus l'index est proche de 1).
- Au-dessus de 45km/h, les abeilles ne volent pas. L'index est donc 0.

Conclusion

Ces quatres index représentent ainsi notre modèle météorologique et sont utilisés dans la fonction **INDEXflight(t)**. Celui-ci permet d'obtenir une valeur entre 0 et 1 des conditions climatiques. Plus celui-ci est proche de 1, plus les conditions climatiques sont favorables aux abeilles. Comme précisé précédemment, l'index d'humidité a été temporairement écarté. Ainsi l'index de vol est la multiplication des index de température, de pluie et de vent.