ΟΜΑΔΙΚΗ ΑΝΑΦΟΡΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟΥ PROJECT

ΗΛΕ47	ΠΛΗΠΡΟ
Ακαδημαϊκό έτος	2021-2022
Ομάδα Α	Αρβανιτίδου Μαρία Σμπώκος Γεώργιος Χατζηεμμανουήλ Αντώνιος

ΤΙΤΛΟΣ PROJECT:
COVID-19 Data Dashboard (No. 13)

ΕΙΣΑΓΩΓΗ

Η εξάπλωση της πανδημίας COVID-19, με λίγο περισσότερα από 418 εκατομμύρια κρούσματα παγκοσμίως, έχει οδηγήσει στον θάνατο, περίπου, έξι εκατομμύρια ανθρώπους. Αυτό δημιούργησε την επιτακτική ανάγκη για καταγραφή, εις βάθος ανάλυση και κατανόηση των επιδημιολογικών δεδομένων, η οποία, μέσω της αποτελεσματικής μοντελοποίησης, θα συμβάλει στην σωστή λήψη αποφάσεων από την πολιτεία, για την αντιμετώπιση του ιού και την προστασία των πολιτών της χώρας.

Στόχος του συγκεκριμένου ομαδικού Project είναι η δημιουργία μιας γραφικής διεπαφής (dashboard), η οποία θα παρουσιάζει με απλό και περιεκτικό τρόπο την εξάπλωση της νόσου στην χώρα, καθώς και την κατάσταση ίασης και εξέλιξης του εμβολιασμού.

Η παρούσα αναφορά περιγράφει, αναλυτικά, όλη τη διαδικασία που ακολουθήσαμε για να φέρουμε εις πέρας το Project, τις προκλήσεις που αντιμετωπίσαμε και τον τρόπο διαχείρισης αυτών, καθώς και το χρονοδιάγραμμα επιτυχούς ολοκλήρωσης του έργου.

ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ (Απαιτήσεις πελάτη)

Το πρόβλημα το οποίο έχουμε αναλάβαμε να φέρουμε εις πέρας είναι η κατασκευή ενός Python Dashboard που θα αποτυπώνει την καθημερινή, αλλά και την διαχρονική εξέλιξη της πανδημίας στην Ελλάδα, με απλό κι εύχρηστο τρόπο. Αντλώντας τα επιδημιολογικά στοιχεία από την επίσημη Κρατική Ιστοσελίδα, ζητείται να παρουσιαστούν αναλυτικά τα στοιχεία της τρέχουσας ημέρας, με πληροφορίες σχετικά με το πλήθος των νέων κρουσμάτων, θανάτων, εμβολιασμών, καθώς και σύγκριση με εκείνα της προηγούμενη ημέρας. Επιπλέον, απαιτείται η δημιουργία πίνακα ο οποίος θα παρουσιάζει την εξέλιξη των επιδημιολογικών στοιχείων από την έναρξη καταγραφής επίσημων στοιχείων έως και σήμερα, όπως επίσης και η αναφορά σε επιδημιολογικούς δείκτες, όπως ο Rt (Effective Reproduction Number). Συμπληρωματικά, το πρόγραμμα θα ενισχύεται με τη διαδικασία μελλοντικής πρόβλεψης της εξέλιξης της πανδημίας με τη γραμμική μέθοδο «Linear Regression». Κοντολογίς, ο κώδικας χρειάζεται να διαβάζει ένα csv αρχείο και να υλοποιεί σε ένα γραφικό περιβάλλον τα παραπάνω.

Χρήστης της συγκεκριμένης εφαρμογής μπορεί να είναι ο οποιοσδήποτε διαθέτει έναν ηλεκτρονικό υπολογιστή με browser και σύνδεση στο Internet. Το μόνο που απαιτείται, επιπλέον, είναι η επιλογή ενός συγκεκριμένου URL, το οποίο ανακατευθύνει απευθείας στη

σελίδα με τα αποτελέσματα. Ο χρήστης δεν έχει μεγάλα περιθώρια αλληλεπίδρασης με την εφαρμογή, γεγονός που την κάνει αρκετά σταθερή και στιβαρή.

Δεν χρειάστηκε να γίνουν συγκεκριμένες παραδοχές κατά την περιγραφή του προβλήματος, καθώς το Project υλοποιήθηκε, ακριβώς, με βάση τις αρχικές προδιαγραφές. Υπάρχουν, όμως, περιθώρια βελτίωσης, κυρίως σε σχέση με την ποιότητα των δεδομένων της Κρατικής Ιστοσελίδας, στα οποία υπάρχει η ανάγκη να γίνει καθαρισμός λανθασμένων εγγραφών (Data Cleansing). Ακόμη, για τη μελλοντική πρόβλεψη, πέρα από τη Γραμμική προσέγγιση, θα είχε αξία να υπάρξει, σε επόμενο στάδιο, και Πολυωνυμική.

ΟΡΙΣΜΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΡΟΔΙΑΓΡΑΦΩΝ (Σχεδιασμός)

Όσον αφορά στο σχεδιασμό του συστήματος του Dashboard δεν υπάρχει κάποια ιδιαίτερη πολυπλοκότητα. Αρχικά, εισάγονται τα δεδομένα (είσοδος) που βρίσκονται στο .csv αρχείο (https://raw.githubusercontent.com/Sandbird/covid19-Greece/master/cases.csv). Η διαχείριση των δεδομένων γίνεται μέσω dataframe της pandas (δομή δεδομένων) και συναρτησιακό προγραμματισμό με τη βοήθεια, βέβαια, επιπλέον βιβλιοθηκών. Οι τύποι των μεταβλητών που χρησιμοποιούνται είναι κατά κανόνα integers, με εξαίρεση τους επιδημιολογικούς δείκτες, οι οποίοι προκύπτουν έπειτα από υπολογισμό μέσω πράξεων και είναι τύπου float. Η οπτικοποίηση της πληροφορίας και των διαγραμμάτων που εμφανίζονται στο χρήστη στο τελικό στάδιο, γίνεται μέσω της Streamlit (https://share.streamlit.io/chatziemmanouil-antonios/eap-plhpro/main/data.py). Για την καλύτερη εμφάνιση της πληροφορίας, υπάρχει δυνατότητα της δημιουργίας πιο ομαλής καμπύλης στα διαγράμματα, η οποία προκύπτει από νέες τιμές στις στήλες του pandas dataframe, οι οποίες εκφράζουν το μέσο όρο των προηγούμενων ακριβώς 10 τιμών.

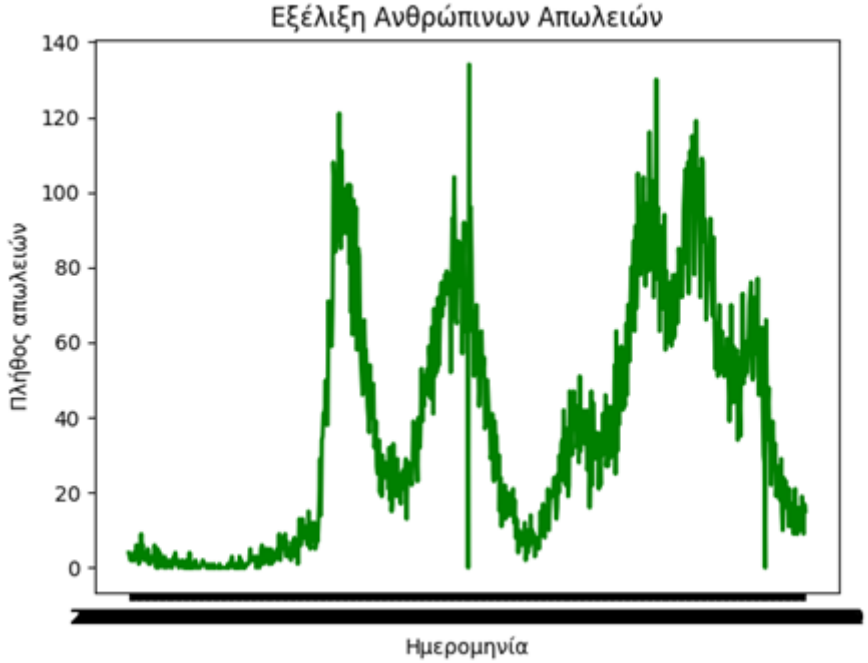
Οι μοναδικές απαιτήσεις του συστήματος σε επίπεδο hardware είναι ένας Η/Υ, σύνδεση στο διαδίκτυο και σε επίπεδο Software λειτουργικό σύστημα με εγκατεστημένο έναν περιηγητή. Για την δημιουργία του Dashboard ο κώδικας αντλείται μέσα από το λογαριασμό της Streamlit (όπου υπάρχει διασύνδεση στη συγκεκριμένη περίπτωση με λογαριασμό στο Github).

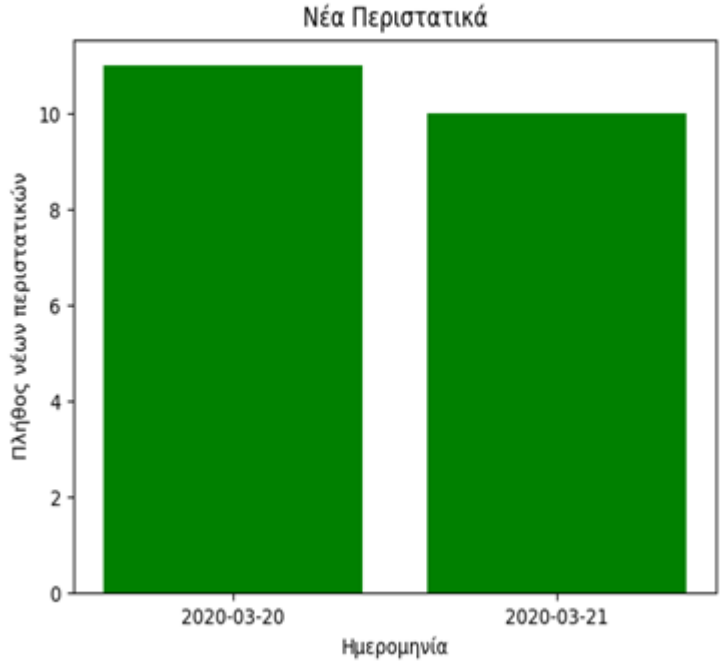
Για την διευκόλυνση της υλοποίησης δεχόμαστε πως τα δεδομένα τα οποία τροφοδοτούμε τον κώδικα είναι έγκυρα και μάλιστα όχι ελλιπή (πράγμα που φυσικά δεν ισχύει). Για τον υπολογισμό κάποιων επιδημιολογικών δεικτών ύστερα από μελέτη βιβλιογραφίας τους ορίσαμε με σχετικά λογικό τρόπο, χάριν αποφυγής της πολυπλοκότητας στον κώδικα.

ΤΕΧΝΙΚΕΣ ΛΕΠΤΟΜΕΡΕΙΕΣ ΥΛΟΠΟΙΗΣΗΣ (Ανάπτυξη)

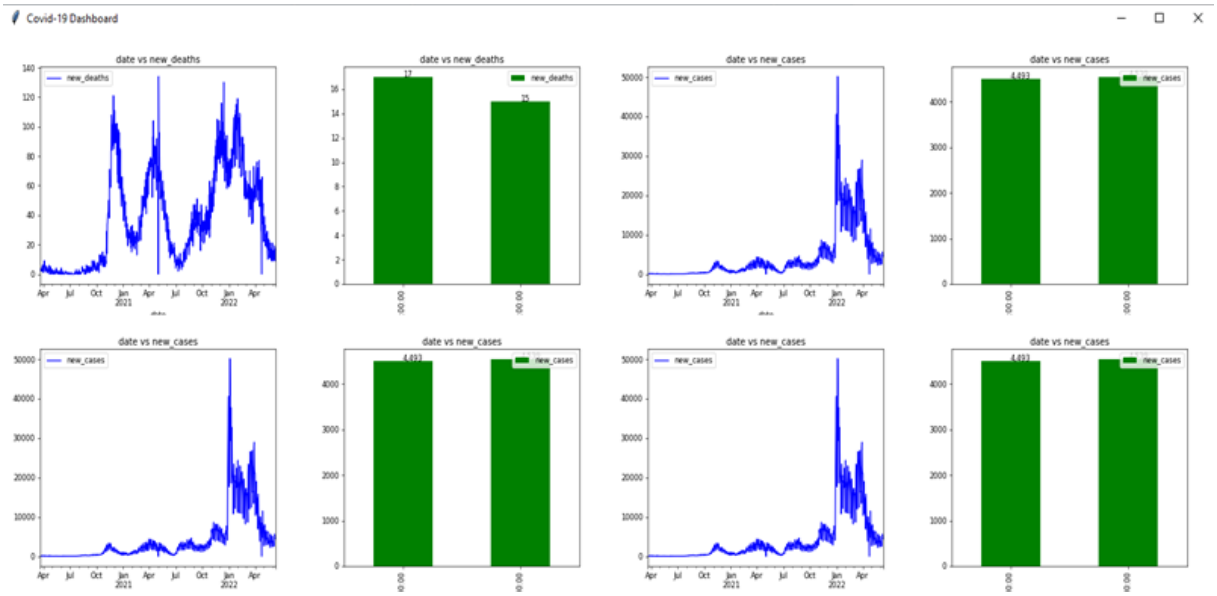
Tkinter

Ξεκινώντας τον προγραμματισμό του Project, καταπιαστήκαμε με την βιβλιοθήκη TKinter για την οπτικοποίηση των δεδομένων. Το βασικό πρόβλημα με το οποίο ήρθαμε αντιμέτωποι ήταν το γεγονός, ότι το κάθε διάγραμμα εμφανιζόταν σε ξεχωριστό figure και για να εμφανιστεί το επόμενο, ήταν απαραίτητο να κλείσει το προηγούμενο. Επιπλέον, είχαμε δεχθεί ότι τα επιδημιολογικά στοιχεία θα τα αντλούμε από ένα csv αρχείο μιας συγκεκριμένης ημερομηνίας (στατική βάση δεδομένων). Πρόβλημα υπήρχε και με το Format της ημερομηνίας, κάτω από τα Bar Charts, η οποία ήταν αντεστραμμένη, καθώς η αρχική μορφή της στην βάση ήταν text.





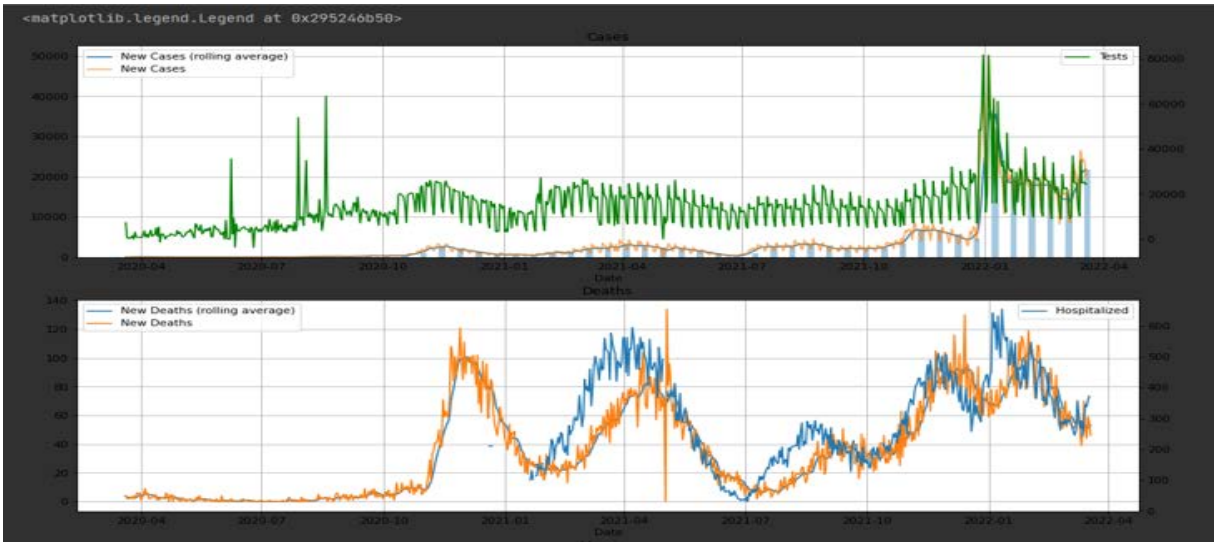
Αναζητώντας λύσεις για τα παραπάνω, καταφέραμε να εμφανίσουμε στο ίδιο figure όλα τα διαγράμματα και να αντλούμε τα στοιχεία απευθείας από την ιστοσελίδα του Υπουργείου (δυναμική βάση δεδομένων).



Το βασικότερο πρόβλημα που είχαμε, σε αυτό το στάδιο, να αντιμετωπίσουμε ήταν το γεγονός ότι δεν βρίσκαμε τον τρόπο να περάσουμε στο figure ένα scroll bar, με αποτέλεσμα να χρειάζεται είτε να μεγαλώσει κατά πολύ το παράθυρο, ώστε να χωρέσουν να εμφανιστούν αρκετά διαγράμματα ή να μικρύνει πολύ το μέγεθος των διαγραμμάτων. Επιπλέον, δυσκολευτήκαμε να βρούμε τον τρόπο να διορθώσουμε το Format στο οποίο εμφανίζονται οι ημερομηνίες κάτω από το Bar Chart. Αν και χρησιμοποιήθηκε συνάρτηση μετατροπής της ημερομηνίας των δεδομένων από text σε date, δεν βρέθηκε τρόπος να περιστραφεί από κάθετη σε οριζόντια θέση και να αφαιρεθεί η ώρα. Καθώς το αποτέλεσμα δεν ήταν το επιθυμητό για την ομάδα μας και καθώς παράλληλα αναζητούσαμε λύσεις και σε άλλες βιβλιοθήκες, επιχειρήσαμε στη συνέχεια να οπτικοποιήσουμε τα δεδομένα με τη χρήση του Matplotlib.

Matplotlib

Η δεύτερη προσπάθεια απεικόνισης των αποτελεσμάτων επιχειρήθηκε με τη χρήση της Matplotlib βιβλιοθήκης της Python, η οποία χρησιμοποιείται για την οπτικοποίηση διαγραμμάτων. Η σύνταξη του κώδικα για την δημιουργία των bar charts και line graphs πραγματοποιήθηκε, κατόπιν μελέτης εκπαιδευτικού υλικού διαθέσιμου στο διαδίκτυο. Το κύριο πρόβλημα ήταν η αδυναμία δημιουργίας checkbox ή selectbox, έτσι ώστε να υπάρχει κάποιου είδους αλληλεπίδραση με τον χρήστη, η οποία θα του επιτρέπει να επιλέξει την πληροφορία που επιθυμεί να προβληθεί στο διάγραμμα. Οι δοκιμές πραγματοποιήθηκαν σε περιβάλλον Jupyter Notebook (μέσω της Anaconda). Παρατίθεται το αποτέλεσμα:



Pyechart

Η τρίτη προσπάθεια απεικόνισης των αποτελεσμάτων έγινε με τη χρήση της Pyechart, μιας ακόμη βιβλιοθήκη της Python, η οποία χρησιμοποιείται για την δημιουργία διαδραστικών διαγραμμάτων. Η συγγραφή του κώδικα για την δημιουργία των bar charts και line graphs ήταν σχετικά εύκολη, μιας και υπήρχε διαθέσιμο υλικό και βίντεο στο διαδίκτυο. Παρόλα αυτά, η τελική υλοποίηση δεν ήταν εφικτή λόγω προβλήματος στη δημιουργία της τελικής html σελίδας. Αναλυτικότερα, μέσω της Pyechart μπορεί να γίνει η συγγραφή του κώδικα απεικόνισης, ο οποίος στο τελικό στάδιο δημιουργεί μια html σελίδα στον φάκελο προορισμού του αρχείου, η οποία περιέχει το επιθυμητό διαδραστικό περιβάλλον. Δυστυχώς, αυτό το βήμα δεν στάθηκε δυνατό να ολοκληρωθεί, αλλά στο μεταξύ, ανακαλύφθηκε η βιβλιοθήκη Streamlit, η οποία , τελικά, συμφωνήθηκε ομόφωνα να αποτελέσει το εργαλείο υλοποίησης του έργου.

Streamlit

Κατόπιν ενδελεχούς έρευνας στο διαδίκτυο, με σκοπό να βρεθεί ένα εύχρηστο και συνάμα δωρεάν εργαλείο, μέσω του οποίου θα μπορούσε κάποιος να προβάλλει διαγράμματα και, γενικά, πληροφορία σε περιηγητή μέσω της Python, χωρίς να έχει ιδιαίτερες γνώσεις σε front-end web designing, ανακαλύψαμε την Streamlit.

Ο κώδικας που συντάχθηκε, δημιουργήθηκε έπειτα από πολύ προσεχτική μελέτη του κώδικα των δημοσίων project που μπορεί κανείς να δει ελεύθερα στη Streamlit gallery **(https://streamlit.io/gallery)** και επίσης, αφού ακολούθησαν πολλές δοκιμές που αφορούσαν, κυρίως, στις λεπτομέρειες της εμφάνισης του Dashboard**.**

- **Επεξήγηση Κώδικα**

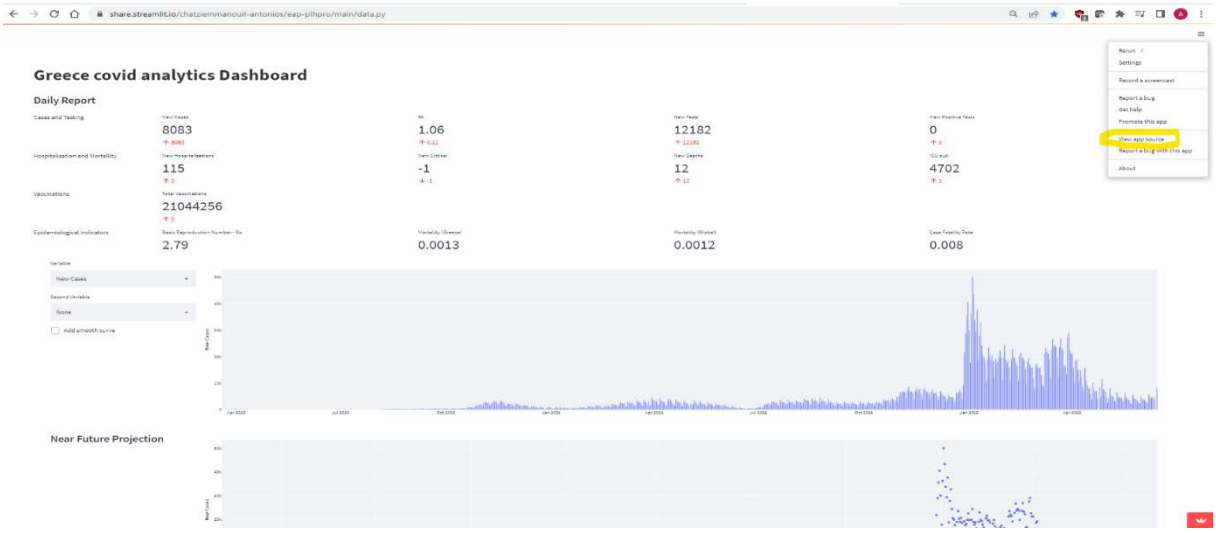
Για τον σχεδιασμό του Dashboard ήταν, αρχικά, απαραίτητη η δημιουργία λογαριασμού στο https://streamlit.io/. Η Streamlit είναι ένα εργαλείο με το οποίο μπορεί κανείς να μετατρέψει data scripts σε web εφαρμογές με δυνατότητα κοινής χρήσης. Όλα αυτά γίνονται με τη χρήση γλώσσας Python (η Streamlit είναι βιβλιοθήκη της Python), χωρίς να χρειάζεται προηγούμενη γνώση (front-end) σχεδιασμού ιστοσελίδων. Στη Streamlit υπάρχει η δυνατότητα δημιουργίας project το οποίο μπορεί να είναι είτε δημόσιο ή ιδιωτικό. Υπάρχει, ακόμα, η δυνατότητα συνεργασίας – διασύνδεσης του project με το GitHub https://github.com/. Για τις ανάγκες της παρούσας παρουσίασης, επιλέχθηκε να δημιουργηθεί ένα δημόσιο project στο GitHub (https://github.com/Chatziemmanouil-Antonios/EAP-PLHPRO) με το όνομα EAP-PLHPRO το οποίο, εν συνεχεία, διασυνδέθηκε ως δημόσιο project με το ίδιο όνομα στη Streamlit. Ο κώδικας, δηλαδή, που χρησιμοποιήθηκε, αρχικά αντλείται από το GitHub, έπειτα εκτελείται και μέσω της Streamlit παράγεται το τελικό εξαγόμενο αποτέλεσμα της οπτικοποίησης του Dashboard σε περιηγητή στον παρακάτω σύνδεσμο https://share.streamlit.io/chatziemmanouil-antonios/eap-plhpro/main/data.py**.**

Για την υλοποίηση του Dashboard στη Streamlit είναι απαραίτητη η χρήση των βιβλιοθηκών:

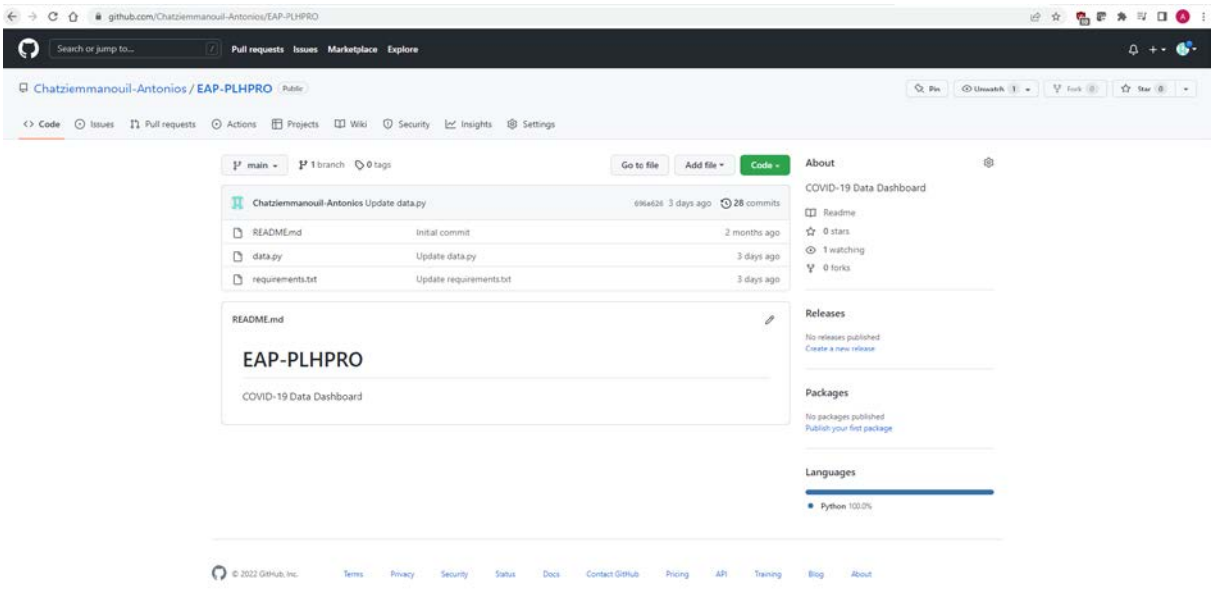
numpy==1.21.2, pandas==1.3.5, plotly==5.7.0, scipy==1.7.3, streamlit==1.8.1, protobuf~=3.19.0

Οι παραπάνω βιβλιοθήκες με τις συγκεκριμένες εκδόσεις τους αναφέρονται στο Project με το όνομα **EAP-PLHPRO** στο **Github** στο αρχείο **requirements.txt**. Στο Project στο **Github** βρίσκεται και το αρχείο **data.py,** όπου περιέχεται ο κώδικας της Python. Μόνο την πρώτη φορά που τρέχει ο κώδικας στη **Streamlit** φορτώνονται οι προαναφερθείσες βιβλιοθήκες και αυτό διαρκεί λίγα λεπτά. Ακολούθως, αυτές αποθηκεύονται για το συγκεκριμένο Project και δεν υπάρχει ιδιαίτερη καθυστέρηση στην εμφάνιση του **Dashboard** στον περιηγητή.

Ο χρήστης έχει τη δυνατότητα να δει τα αρχεία του Project με το όνομα EAP-PLHPRO στο Github μέσω του Streamlit, εφόσον επιλέξει στην πάνω δεξιά γωνία ‘View app source’:



Αρχεία στο GitHub:



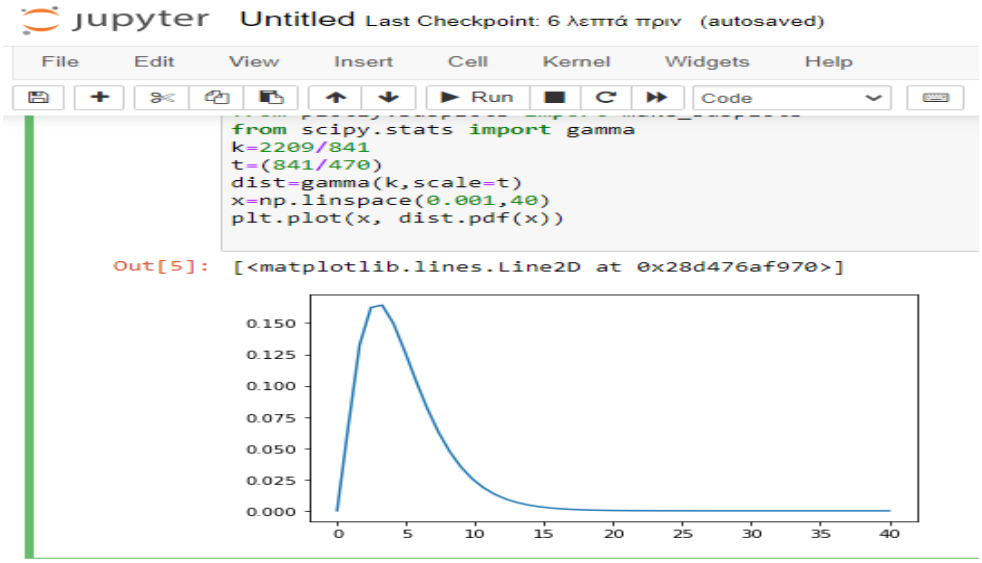
Στον κώδικα, καταρχάς, δηλώνονται οι βιβλιοθήκες και τα module βιβλιοθηκών που θα χρησιμοποιηθούν.

Έπειτα γίνεται εισαγωγή των δεδομένων από το .csv αρχείο: `https://raw.githubusercontent.com/Sandbird/covid19-Greece/master/cases.csv` στο οποίο όμως αναγνωρίζονται ως ημερομηνίες η στήλη “date” που περιέχεται στο pandas dataframe (πηγή: https://www.adamsmith.haus/python/answers/how-to-import-dates-in-a-csv-file-as-datetimes-in-a-pandas-dataframe-in-python).

Δημιουργούνται δύο νέες στήλες στο pandas dataframe **“new_positive_tests”** και **“new_vaccinations”**. Οι τιμές για την στήλη **“new_positive_tests”** υπολογίζονται ως διαφορές μεταξύ δύο διαδοχικών ημερομηνιών της στήλης **“positive_tests”.** Με τον ίδιο τρόπο υπολογίζονται και οι τιμές της στήλης **“new_vaccinations”** με βάση τις τιμές στην στήλη **“total_vaccinations”.**

Στη συνέχεια γίνεται υπολογισμός του επιδημιολογικού δείκτη Case Fatality Rate (CFR) τον οποίο ορίζουμε: ως το πηλίκο της διαίρεσης των συνολικών θανάτων προς τα συνολικά κρούσματα.

Έπειτα ορίζεται η συνάρτηση, η οποία υπολογίζει τον επιδημιολογικό δείκτη **Effective Reproduction Number – Rt (t)** (ο δείκτης περιγράφει με απλά λόγια τη μεταδοτικότητα του ιού σε έναν συγκεκριμένο χρόνο). Σύμφωνα με την βιβλιογραφία που μελετήθηκε υπάρχει η Γάμμα κατανομή [με mean (μ.ο.) = k*θ = 4.7 και variance (διακύμανση) = κ*θ^2 = 2.9^2], της οποίας η συνάρτηση πυκνότητας πιθανότητας [probability density function (PDF)] προσεγγίζει ικανοποιητικά την χρονική περίοδο που μεσολαβεί μεταξύ της μόλυνσης κάποιου από τον ιό μέχρι και το σημείο να θεωρηθεί αυτός μολυσμένος (χρόνος επώασης) (συνήθως δηλαδή μέχρι αυτός να εμφανίσει συμπτώματα). Υπολογίσαμε από τις σχέσεις k*θ = 4.7 και κ*θ^2 = 2.9^2 τις παραμέτρους κ και θ της Γάμμα κατανομής (στην συνάρτηση Rt το θ είναι το t). Με τη βοήθεια του **https://www.wolframalpha.com/** βρίσκουμε πως κ=2209/841 και θ=841/470. Ο δείκτης Rt, με μία απλή προσέγγιση, ορίζεται ως το πηλίκο των κρουσμάτων σήμερα, προς το σύνολο των κρουσμάτων των προηγούμενων <<x>> ημερών. Καθώς ο αριθμός των test που πραγματοποιούνται ημερησίως δεν είναι σταθερός, γεγονός το οποίο επηρεάζει το δείκτη Rt, θεωρήσαμε ότι ένας μέσος όρος των προηγούμενων τιμών των κρουσμάτων σε ένα διάστημα βάθους 10 ημερών θα εξομάλυνε τις τιμές του Rt (θεωρούμε ότι στα διαστήματα των 10 ημερών τα συνολικά τεστ που πραγματοποιούνται είναι, περίπου, τα ίδια). Έτσι, στον υπολογισμό του Rt δημιουργούμε προσωρινά στο pandas dataframe μία προσωρινή στήλη ιδίου μεγέθους με την new_cases στην οποία όμως οι τιμές της στήλης new_cases είναι <<διορθωμένες>> με το μέσο όρο των 10 προηγούμενων τιμών. Αν παρατηρήσουμε στην κατανομή Γάμμα την συνάρτηση πυκνότητας πιθανότητας [probability density function (PDF)] για κ=2209/841 και θ=841/470 παρατηρούμε, ότι μετά τις 20 μέρες περίπου, η καμπύλη σταθεροποιείται και η πιθανότητα να έχει μολυνθεί κάποιος την 20η ημέρα σχεδόν μηδενίζεται.



Λαμβάνοντας αυτά υπόψιν, καταλήξαμε, ότι είναι δόκιμο να ορίσουμε την Rt (t) συνάρτηση για μία μέρα t ως το πηλίκο των κρουσμάτων την ημέρα t προς το άθροισμα των προηγούμενων 20 ημερών, στο οποίο όμως άθροισμα υπάρχει για κάθε μία μέρα <<ένας συντελεστής βαρύτητας>> μέσω της Γάμμα κατανομής.

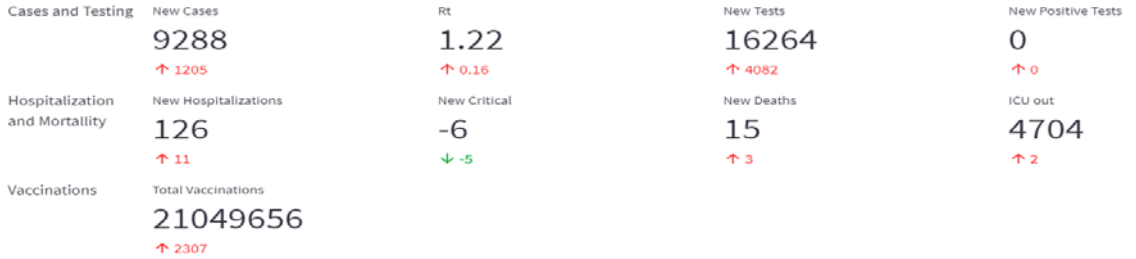
Στη συνέχεια, προσθέτουμε στο Dataframe μία νέα στήλη, κενή αρχικά, με την ονομασία Rt και την <<γεμίζουμε>> με τιμές για κάθε μία μέρα, μέσω της συνάρτησης Rt (t).

Για την εμφάνιση των επιδημιολογικών δεικτών έχουν χρησιμοποιηθεί μετρικές της Streamlit (πηγή:https://docs.streamlit.io/library/api-reference/data/st.metric).

Για την εμφάνιση των επιδημιολογικών δεικτών, όσον αφορά στις τρεις πρώτες γραμμές αυτών, έχει χρησιμοποιηθεί για την μετάφραση των ετικετών των δεικτών το λεξιλόγιο value_labels.

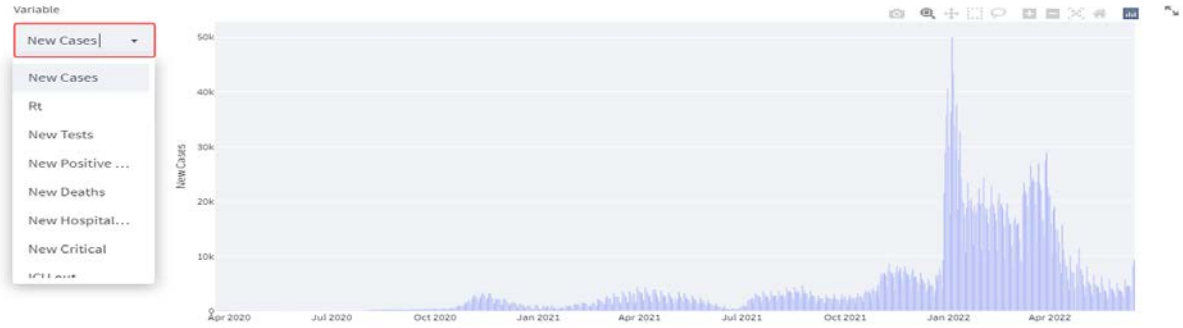
Για την κατηγοριοποίηση των επιδημιολογικών δεικτών των τριών πρώτων γραμμών και την εμφάνιση τους σε σειρά με συμμετρική απόσταση μεταξύ τους, έχει χρησιμοποιηθεί το λεξιλόγιο Rows, όπου το key είναι η περιγραφή της κατηγορίας και το value είναι μία λίστα με τα ονόματα των επιδημιολογικών δεικτών (χρησιμοποιήθηκαν σε λίστα σαν στοιχεία τα string ‘ ‘ για λόγους συμμετρίας στην εμφάνιση).

Για την ταυτόχρονη διαχείριση της εμφάνισης των μετρικών της Streamlit (για τις τρεις πρώτες γραμμές των δεικτών) χρησιμοποιείται διπλή επαναληπτική εντολή for σε συνδυασμό με τα δύο λεξιλόγια και έτσι εμφανίζονται οι ημερήσιες τιμές και η μεταβολή τους σε σχέση (κατά κανόνα) με την προηγούμενη ημέρα. Επειδή, όμως, υπάρχει περίπτωση μέσα στο dataframe να υπάρχει σε στήλη NaN value, πιο σωστό θα ήταν να πούμε πως οι διαφορές που εμφανίζονται αφορούν τη διαφορά των δύο τελευταίων μη NaN value, πράγμα το οποίο δεν σημαίνει απαραίτητα πως πρόκειται για διαφορά στις τιμές δύο διαδοχικών ημερών.



Για την εμφάνιση των επιδημιολογικών δεικτών της τέταρτης σειράς, αυτοί απλά δηλώνονται σαν μετρικές της Streamlit, χωρίς όμως να υπολογίζεται κάποια μεταβολή (οι τιμές τους έχουν αντληθεί από την βιβλιογραφία με εξαίρεση την Case Fatality Rate η οποία υπολογίζεται).

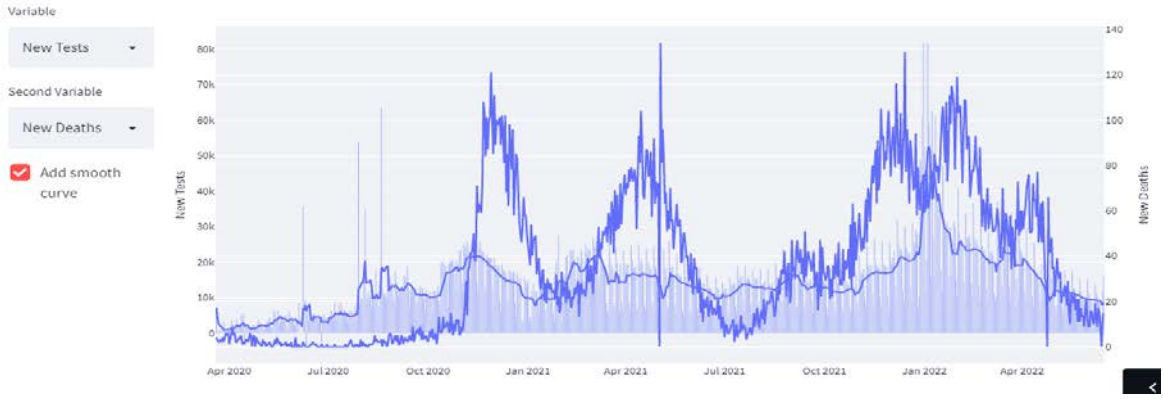
Δημιουργήσαμε δύο selectbox όπου ο χρήστης μπορεί να δει την μεταβλητή που κάθε φορά επιθυμεί στο διάγραμμα το οποίο βρίσκεται ακριβώς δίπλα. Για τις επιλογές που υπάρχουν στα δύο selectbox χρησιμοποιήθηκε το λεξικό value_labels.



Υπάρχει η δυνατότητα για τον χρήστη να επιλέξει και το <<Add smooth curve>> checkbox για μία μεταβλητή στο διάγραμμα για λόγους αισθητικής στην εμφάνιση (εφαρμόζεται μόνο για μεταβλητή του πρώτου selectbox). Αυτό σημαίνει πως εμφανίζεται διάγραμμα συνεχούς καμπύλης των <<διορθωμένων>> τιμών της μεταβλητής, όπου κάθε τιμή που βρίσκεται στη

στήλη του dataframe έχει αντικατασταθεί προσωρινά με το μέσο όρο των 7 προηγούμενων τιμών.

Να σημειωθεί πως για λόγους αισθητικής στην εμφάνιση εάν επιλεχθεί να εμφανιστεί δεύτερο διάγραμμα από το δεύτερο selectbox από το χρήστη αυτό γίνεται πάντα με την μορφή καμπύλης και όχι ραβδογράμματος.

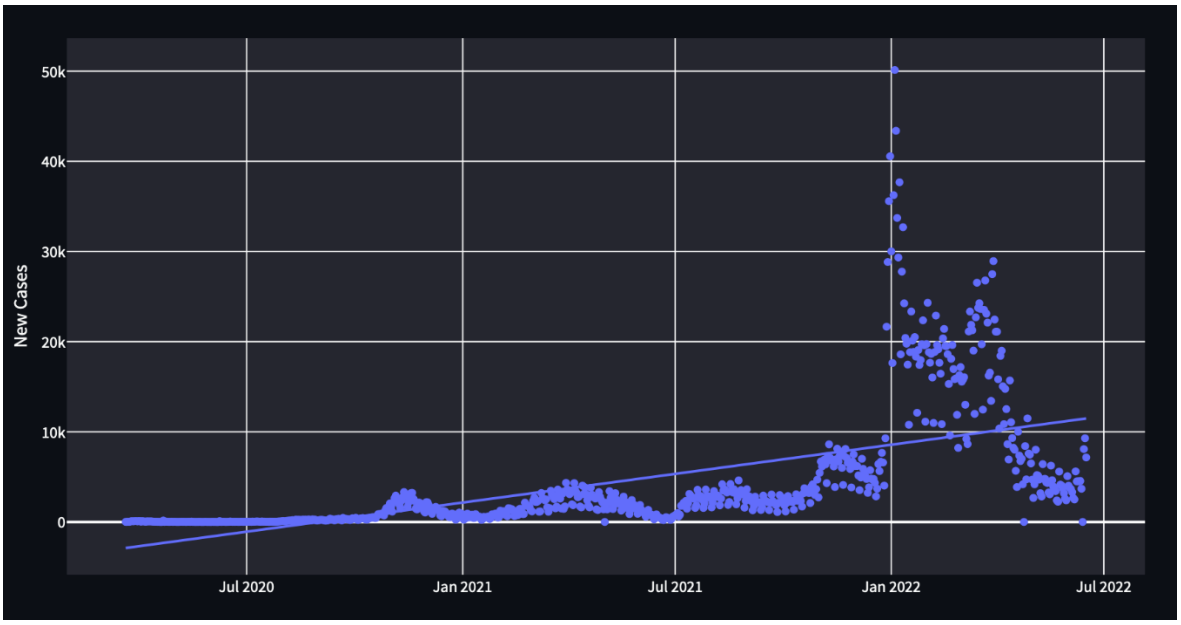


Το τελευταίο τμήμα του Dashboard αφορά την προβολή της εκτίμησης των κρουσμάτων στο κοντινό μέλλον μέσω της μεθόδου <<γραμμικής παλινδρόμησης>>.

Τέλος, δίνεται η δυνατότητα στον χρήστη επιλέγοντας το checkbox ‘Display dataset’ να δει κομμάτι των δεδομένων του dataframe.

LinearRegression

Μία από τις προδιαγραφές του λογισμικού ήταν και η μελλοντική πρόβλεψη της πανδημίας στη χώρα μας με χρήση της γραμμικής παλινδρόμησης. Η αρχική προσπάθεια έγινε στο Visual Studio Code χρησιμοποιώντας Matplotlib/pyplot και sklearn, μέσω της οποίας εκπαιδεύτηκε και εξετάστηκε το δείγμα (train_test_split, test 20%, train 80%) και στην συνέχεια, υπολογίστηκε η κλίση b1 της ευθείας (LinearRegression.coeff_) και ο σταθερός παράγοντας b0 της ευθείας (LinearRegression.intercept_), όπως και το άθροισμα των ελαχίστων τετραγώνων (LinearRegression.score(x, y)). Επίσης, έγινε μια προσπάθεια υλοποίησης της πολυωνυμικής προσέγγισης (numpy.polyfit(x, y, a)) για καλύτερα και πιο ακριβή αποτελέσματα, αλλά δυστυχώς δεν κατέστη εφικτή. Συνδυαστικά με αυτή τη συνάρτηση, χρησιμοποιήθηκε η Seaborn για απεικόνιση, αλλά και πάλι αντιμετωπίσαμε προβλήματα κατά την εφαρμογή της. Στη συνέχεια, η βιβλιοθήκη Pyecharts χρησιμοποιήθηκε με στόχο την δημιουργία διαδραστικής διεπαφής, και όχι μόνο την απεικόνιση των δεδομένων. Όπως προαναφέρθηκε, η υλοποίηση αυτή δεν πραγματοποιήθηκε, εξαιτίας προβλήματος στο τελευταίο βήμα δημιουργίας του html αρχείου (html=X.render). Εν τούτοις, μετά από πολλές δοκιμές η υλοποίηση του Linear Regression έγινε στο Streamlit με χρήση της PlotLy Express.



ΕΛΕΓΧΟΣ ΚΑΙ ΑΠΟΔΟΧΗ ΛΟΓΙΣΜΙΚΟΥ

Κατά τη διάρκεια και μέχρι και την ολοκλήρωση του project γινόταν έλεγχος της λειτουργίας του λογισμικού σε διάφορα επίπεδα, με σκοπό να φτάσουμε στα επιθυμητά τελικά αποτελέσματα.

Αρχικά, πραγματοποιήθηκε ο λειτουργικός έλεγχος του λογισμικού, σύμφωνα με τον οποίο δόθηκε έμφαση στη λειτουργία του λογισμικού, σύμφωνα με τις προδιαγραφές των απαιτήσεων του πελάτη. Οι προδιαγραφές αυτές αποτελούν την ορθή απεικόνιση και πρόβλεψη της εξέλιξης της πανδημίας μέσω χρήσης ενός dashboard, με το οποίο ο χρήστης αλληλεπιδρά.

Επίσης, κατά την διάρκεια διαμόρφωσης του τελικού dashboard γινόταν και έλεγχος διεπαφής. Δηλαδή, επιβεβαιωνόταν η ορθή επικοινωνία των σχετιζόμενων τμημάτων του κώδικα που γράφτηκε για αυτή την διεπαφή, ώστε να μεταβάλλονται σύμφωνα με τις επιλογές του χρήστη. Συγκεκριμένα, η επιλογή “Νέα κρούσματα” απεικόνιζε όντως γραφικά τα νέα κρούσματα και η αλλαγή της επιλογής σε “Νέοι θάνατοι” έδειχνε τα αντίστοιχα δεδομένα του δείγματος. Όλη η προαναφερθείσα διαδικασία ελέγχου του λογισμικού, αποτέλεσε το alpha testing του λογισμικού και πραγματοποιήθηκε από εμάς. Στη συνέχεια, πραγματοποιήθηκε και beta testing από άτομα του οικογενειακού/φιλικού περιβάλλοντος μας, με στόχο να αποκτήσουμε περαιτέρω κριτική πάνω στην χρησιμότητα και ευχρηστία του dashboard.

Επίσης, έγινε αξιολόγηση του συστήματος όσον αφορά στην ασφάλεια και απόδοση του. Όπως έχει προαναφερθεί, η συγκεκριμένη εργασία χρησιμοποιεί ένα αρχείο csv ως την πηγή πληροφοριών για την δημιουργία του Dashboard. Εξαιτίας δυναμικών μεταβολών της πηγής δεδομένων (εισαγωγή νέων δεδομένων στο δείγμα καθημερινά), επηρεάζεται και η απόδοση του συστήματος. Μέχρι στιγμής το σύστημα έχει καλή απόδοση, λόγω μικρού μεγέθους δείγματος, αλλά υπάρχει περίπτωση μειωμένης απόδοσης σε περίπτωση εισαγωγής μεγάλου όγκου δεδομένων. Αναφορικά με την ασφάλεια του συστήματος, έχει αξιολογηθεί ως καλή, αλλά μελλοντικά, είναι πιθανόν να χρειαστεί να μετατραπεί το GitHub repo σε private.

Συνολικά, τα κριτήρια αποδοχής του λογισμικού φαίνεται να πληρούνται στην παρούσα εργασία και να είναι σύμφωνα με τις προδιαγραφές των απαιτήσεων του πελάτη, όπως και με τις ανάγκες του χρήστη.

ΠΑΡΑΤΗΡΗΣΕΙΣ ΕΠΙ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ & ΤΟΥ ΘΕΜΑΤΙΚΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ

Η επιστήμη των δεδομένων έχει υπάρξει ένας γρήγορα αναπτυσσόμενος τομέας των επιστημών των μαθηματικών και της πληροφορικής. Μέσω αυτού του project είχαμε την ευκαιρία να κάνουμε μια εισαγωγή στην επιστήμη των δεδομένων και παράλληλα να αρχίσουμε τα πρώτα βήματα στον προγραμματισμό με την χρήση της γλώσσας Python. Ο πειραματισμός με τις διάφορες βιβλιοθήκες, ασχέτως αν η τελική εφαρμογή ήταν επιτυχής, μας βοήθησε να μάθουμε περισσότερα, πέραν του τυπικού πανεπιστημιακού διαβάσματος, καθώς επίσης και να κατανοήσουμε την πρόκληση της υλοποίησης λειτουργιών σε προγραμματιστικό επίπεδο. Τέλος, είχαμε την ευκαιρία να δουλέψουμε ομαδικά και να μοιραστούμε χρήσιμες πληροφορίες με όλη την ομάδα στην προσπάθειά μας για ουσιαστική επικοινωνία μεταξύ μας.

ΙΔΕΕΣ ΕΠΕΚΤΑΣΕΙΣ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ

Το project ολοκληρώθηκε σύμφωνα με τις απαιτήσεις του πελάτη (ΕΑΠ/ΣΕΠ) αλλά υπάρχει η δυνατότητα βελτίωσης και επέκτασης κάποιων στοιχείων της συγκεκριμένης δουλειάς.

Η εύρεση πιο αξιόπιστων πηγών δεδομένων θα έδινε καλύτερα αποτελέσματα. Η κακή ποιότητα των δεδομένων κατέστησε δύσκολο τον καθαρισμό του δείγματος και δυστυχώς, οι προσπάθειες καθαρισμού που έγιναν δεν απέδωσαν, υποχρεώνοντας μας να κάνουμε χρήση όλων των δεδομένων.

Αυτή η χαμηλή ποιότητα δεδομένων αντικατοπτρίζεται επίσης και στην απεικόνιση της γραμμικής παλινδρόμησης, όπου πολλά δεδομένα απομακρύνονται από το ιδανικό σενάριο του R2>0.9. Παρόλα αυτά, μια προσπάθεια καλύτερης πρόβλεψης αποτελεί η πολυωνυμική παλινδρόμηση, στην οποία γίνεται μια πολυωνυμική προσέγγιση του δείγματος, ακολουθώντας την τάση του δείγματος σε μεγαλύτερο βαθμό. Η προσπάθεια χρήσης της πολυωνυμικής παλινδρόμησης, δυστυχώς δεν ολοκληρώθηκε στα πλαίσια της παρούσας παράδοσης της εφαρμογής..

ΣΧΟΛΙΑΣΜΟΣ ΤΗΣ ΓΛΩΣΣΑΣ PYTHON ΩΣ ΕΡΓΑΛΕΙΟ ΥΛΟΠΟΙΗΣΗΣ

Η Python ως γλώσσα προγραμματισμού ανοιχτού κώδικα θεωρείται υψηλού επιπέδου και μια από τις καλύτερες γλώσσες για διάφορα έργα ή εφαρμογές. Εκτός από το γεγονός ότι παρέχει μεγάλη λειτουργικότητα για την αντιμετώπιση μαθηματικών, στατιστικών και επιστημονικών προβλημάτων, παρέχει και αμέτρητες βιβλιοθήκες για να χρησιμοποιήσει, όποιος καταπιάνεται με την εφαρμογή της επιστήμης δεδομένων. Η ευκολία στη χρήση και η σχετικά απλή σύνταξη της, βοηθάει αρκετά, ώστε άτομα χωρίς ιδιαίτερο τεχνικό υπόβαθρο (προπτυχιακοί φοιτητές λόγου χάρη), να κατορθώνουν να προσαρμοστούν στις απαιτήσεις προγραμματισμού. Αυτός είναι και ο κυριότερος λόγος που χρησιμοποιείται ευρέως από σειρά επιστημονικών και ερευνητικών κοινοτήτων.

Στη δική μας περίπτωση, η δύναμη της πληθώρας των βιβλιοθηκών έγινε φανερή, αφού για να καταλήξουμε στην τελική μας επιλογή, είχαμε τη δυνατότητα να πειραματιστούμε με άλλες τέσσερις διαφορετικές εκδοχές του ίδιου αποτελέσματος, δοκιμάζοντας μερικά από τα άφθονα εργαλεία που παρέχονται δωρεάν. Στο τέλος, συνεχίσαμε με αυτό που ταίριαζε καλύτερα στο αποτέλεσμα, όπως νοητικά το είχαμε σχεδιάσει. Επιπροσθέτως, το γεγονός ότι επιτρέπεται στο χρήστη να εκτελεί τον κώδικα οπουδήποτε, συμπεριλαμβανομένων των Windows, MacOS X, UNIX και Linux, προσέφερε μεγάλη διευκόλυνση, καθώς τα μέλη της προγραμματιστικής μας ομάδας δουλεύαμε σε διαφορετικά OS.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. https://dafarry.github.io/tkinterbook (Βιβλίο με όλες τις μεθόδους του Tkinter)
2. https://www.youtube.com/watch?v=8JJ101D3knE&t=4s (GIT Tutorial for Beginners)
3. https://www.youtube.com/results?search_query=corey+schafer+git (GIT & Python)
4. https://www.youtube.com/watch?v=JrWHyqonGj8 (GUI with Python's Tkinter, by Robert Jomar Malate)
5. https://www.youtube.com/watch?v=AP4FalhHvUo&list=PLCQT7jmSF-LrwYppkB3Xdbe6QC81-ozmT (8 Video Series for Tkinter)
6. https://datatofish.com/matplotlib-charts-tkinter-gui/ (Tkinter)
7. https://www.youtube.com/watch?v=hSPmj7mK6ng (Dash Dashboard)
8. https://www.youtube.com/watch?v=xE95tIzCuKM (Pyechart)
9. https://medium.com/codex/powerful-visualization-using-python-pyecharts-code-included-29cb1135a0d3 (Pyechart)
10. https://programmer.ink/think/pyecharts-easily-draw-30-super-practical-and-exquisite-charts.html (Pyechart)
11. https://seaborn.pydata.org/ (Seaborn)
12. https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html (polyfit)
13. https://docs.streamlit.io/ (Streamlit)
14. https://plotly.com/python/linear-fits/ (Plotly)
15. Nishiura H., Linton N.M., Akhmetzhanov A.R. Serial interval of novel coronavirus (COVID-19) infections. Int J Infect Dis. April 2020;93:284–286. (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7128842/)
16. https://en.wikipedia.org/wiki/Gamma_distribution
17. Wang, H., Paulson, K. R., Pease, S. A., Watson, S., Comfort, H., Zheng, P., Aravkin, A. Y., Bisignano, C., Barber, R. M., Alam, T., et al. (2022). Estimating excess mortality due to the covid-19 pandemic: a systematic analysis of covid-19-related mortality, 2020–21. The Lancet. (https://www.sciencedirect.com/science/article/pii/S0140673621027963)
18. Liu Y, Gayle AA, Wilder-Smith A, Rocklöv J. The reproductive number of COVID-19 is higher compared to SARS coronavirus. J Travel Med. 2020;27(2):taaa021. doi:10.1093/jtm/taaa021 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7074654/)
19. https://apothesis.eap.gr/bitstream/repo/43246/1/115716_%CE%A0%CE%91%CE%A4%CE%A3%CE%9A%CE%91%CE%A3_%CE%A7%CE%A1%CE%97%CE%A3%CE%A4%CE%9F%CE%A3.pdf
20. http://ikee.lib.auth.gr/record/319055/files/%CE%A0%CF%84%CF%85%CF%87%CE%B9%CE%B1%CE%BA%CE%AE%20%CE%9C%CE%B1%CF%81%CE%BF%CF%85%CE%BB%CE%AF%CE%B4%CE%B7%CF%82%20%CE%A0%CE%B1%CE%BD%CE%B1%CE%B3%CE%B9%CF%8E%CF%84%CE%B7%CF%82%202431.pdf?version=1 https://covid19.gov.gr/covid19-live-analytics/
21. https://www.w3schools.com/python/ref_func_zip.asp
22. https://www.adamsmith.haus/python/answers/how-to-import-dates-in-a-csv-file-as-datetimes-in-a-pandas-dataframe-in-python
23. https://www.adamsmith.haus/python/answers/how-to-import-dates-in-a-csv-file-as-datetimes-in-a-pandas-dataframe-in-python
24. https://docs.streamlit.io/library/api-reference/data/st.metric