علوم گیاهی

این اپلیکیشن جهت آنالیز و تخمین شاخص های تنش غیرزیست گیاهی طراحی شده است و می تواند به طور گسترده در برنامه های زراعت و اصلاح نباتات به عنوان یک رابط کاربر پسند برای کشاورزان و پرورش دهندگانی که با حجم زیادی از داده ها سروکاردارند ، استفاده شود.

پیش نیازهای استفاده :

لازم است که کاربر از یک مرورگر وب برای تعامل و دسترسی سریع به برنامه استفاده کند

استفاده از این نرم افزار آسان است و نیازی به یادگیری اولیه دانشی ندارد

مرورگری که از برنامه های کاربردی وب و جاوا اسکریپت پشتیبانی می کند.

تفاوتی ندارد که کامپیوتر شخصی باشد یا موبایل یا تبلت یا ...

این نرم افزار با معماری M.V.C کاملا جهانی ، بهینه و حرفه ای طراحی شده و تمامی کدها تا حد امکان ساده، خوانا، تمیز نوشته شده اند.

بررسی اجمالی نیازمندیها :

مجموعه داده های بزرگ را در حداقل زمان تجزیه و تحلیل کند.

در برنامه‌های اصلاح گیاهان ، ارزیابی عملکرد عمده در شرایط بهینه و تنش برای انتخاب ژنوتیپ‌های مقاوم، به عنوان یک شاخص کلیدی مورد استفاده قرار می‌گیرد. در چهار دهه گذشته، چندین شاخص برای ارزیابی تحمل به تنش در محصولات کشاورزی پیشنهاد شده است . با وجود استفاده خوب از این شاخص‌ها در کشاورزی و اصلاح نباتات، هنوز نرم‌افزاری کاربرپسند برای دسترسی به این روش‌ها وجود ندارد و یا با سرعت پایینی برخوردار است .

شاخص های آماری – ریاضی که تعریف شده اند و در پروژه به کار گرفته شده اند :

Index	Formula	Pattern of selection	Reference
Tolerance	$TOL= Y_P - Y_S$	Minimum value	Rosielle and Hamblin (1981)
Mean productivity	$MP = \frac{Y_P + Y_S}{2}$	Maximum value	Rosielle and Hamblin (1981)
Geometric mean productivity	$GMP= \sqrt{Y_S \times Y_P}$	Maximum value	Fernandez (1992)
Harmonic mean	$HM= \frac{2(Y_S \times Y_P)}{(Y_S + Y_P)}$	Maximum value	Bidinger et al. (1987)
Stress susceptibility index	$SSI= \frac{1-(Y_S / Y_P)}{1-(\bar{Y}S / \bar{Y}P)}$	Minimum value	Fischer and Maurer (1978)
Stress tolerance index	$STI= \frac{Y_S \times Y_P}{(\bar{Y}P)^2}$	Maximum value	Fernandez (1992)
Yield index	$YI= \frac{Y_S}{\bar{Y}_S}$	Maximum value	Gavuzzi et al. (1997)
Yield stability index	$YSI= \frac{Y_S}{Y_P}$	Maximum value	Bouslama and Schapaugh (1984)
Relative stress index	$RSI= \frac{(Y_S / Y_P)}{(\bar{Y}S / \bar{Y}P)}$	Maximum value	Fischer and Wood (1979)

رابط های کاربری :

کاربران پس از دسترسی به نرم افزار (استفاده از وب سایت) به راحتی می توانند با ویژگی های اپلیکیشن ارتباط برقرار کنند. در مرحله آغازین : کاربر برای استفاده از اپلیکیشن ، باید دارای فایل دیتا ست مانند فایل مثال داخل وبسایت که در قالب خاصی تعبیه شده است ، جهت پردازش آنان ، باشد . پس از ارسال و بارگذاری داده ها به وبسایت فقط کافیست دکمه process را زده و پس از گذشت کسری از ثانیه ، نتایج آنالیز آنان قابل نمایش و کاربر به صفحه نتایج فرستاده می شود و کاربر میتواند خروجی ها و نتایج را برای خود دانلود کند. در صفحه اصلی، کاربر به موارد زیر دسترسی دارد : 1. ارسال فایل داده ها (دیتا ست) 2. دستور العمل ها و نحوه استفاده از نرم افزار 3. نمایش فرمول های استفاده شده جهت پردازش 4. ارتباط با تیم توسعه دهنده 5. دسترسی به فایل پیشفرض (مثال)

توجه: کاربر برای استفاده از خدمات اپلیکیشن باید به فایلی که حاوی داده می باشد ، در قالب مثال زیر باشد :

	A	B	C	D
1	genotype	Yp	Ys	
2	G1	66.12	53.51	
3	G2	86.85	75.18	
4	G3	98.65	84.38	
5	G4	74.23	35.11	
6	G5	63.35	54.70	
7	G6	74.43	49.42	
8	G7	63.66	49.17	
9	G8	85.61	59.08	
10	G9	86.34	59.89	
11	G10	74.24	56.07	

صفحات و جزئیات مختلف در اپلیکیشن :

-صفحه اصلی: [شامل : ارسال کننده دیتا ، آموزش نحوه کار با وبسایت ، دسترسی به فایل دیتا پیشفرض ، نمایش فرمول های استفاده شده در اپلیکیشن]

-نتایج : [نمایش نمودار ها و دانلود آنها: { pca ، Frequencies ، Correlation Plots ، 3D Plot } ، نمایش جدول نتایج و دانلود آنها]

رابط های فنی:

در برنامه، دیتا های استخراج شده و فایل های دیتا ست (excel file) داده‌های مهم ما هستند که فرآیندها بر روی آنها اعمال می‌شوند . پس از دریافت فایل دیتا ست توسط وبسایت ، شروع به محاسبه و اعمال فرمول ها بر روی دیتا ها می کند . دیتا ها توسط دانشجویان ، معلمان و اساتید و محققان تولید و میتوانند از این وبسایت استفاده کنند . ساختار این داده ها بر اساس دیتا های طبقه بندی شده در فایل xlsx می باشند که دیتا خام شامل Yp و Ys هستند . دیتا ارسالی ، توسط کتابخانه ها و توابع نوشته شده به زبان پایتون پردازش و محاسبه شده و توسط زبان سمت سرور node js نتایج را به سمت کلاینت ارسال میکند .

توابع مهم در برنامه :

پس از اجرا می توان از تابع Calculate برای بدست آوردن نتایج استفاده کرد. یک دیتافریم را به عنوان ورودی می گیرد و نتیجه را به عنوان یک شی شامل دیتافریم های result شاخص ها، result رتبه ها و ماتریس های همبستگی correlations results pearson و spearman correlations results را برمی گرداند.

Calculate : این تابع تمامی محاسبات را یکجا انجام داده و در خروجی در قالب **Dictionary** و **Array** به خروجی می فرستد .

پارامتر هایی که توسط این تابع به خروجی می روند :

indices ، ranks ، correlations : [pearson , spearman] ، pca : [correlation_based covariance_based]

Math's : مجموعه توابع آماری و ریاضی جهت محاسبه و انجام عملیات شاخص ها برروی دیتا

شاخص های محاسباتی در این پروژه ، در جدول صفحات بالا آمده است

GetPCA : دارای **eigenvals** (مقادیر ویژه) ، **eigenvecs** (بردارهای ویژه) ، بارگیری (بارگیری عامل) ، مشارکت (مشارکت متغیرها) ، امتیازها (امتیازهای عامل) می باشد.

Ranks : مقدار رتبه بندی دیتا های محاسبه شده را بر میگرداند

Download sample file : تابعی می باشد که فایل دیتا ست از پیش تعریف شده جهت نمایش کارکرد وب سایت را برای کاربر دانلود می کند

اطلاعات اپلیکیشن :

- ساختار نرم افزار :
- **Activity**
- **Use case**

- کنوانسیون ها:

پلتفرم : وب اپلیکیشن ، **web base**

برای اجرای اپلیکیشن از پلتفرم های وب استفاده می شود . اپلیکیشن در محیط وب سایت (وب اپلیکیشن) توسعه می یابد

سمت سرور: **Node js (express)**

به منظور پیاده سازی منطق و رفتارهای سمت سرور، در این پروژه از زبان های برنامه نویسی جاوا اسکریپت و نود جی اس استفاده شده است زیرا سرعت، امنیت و قابلیت مدیریت داده مناسبی دارند.

سمت کلاینت : **EJS – Css - js**

برای سمت کاربر و طراحی طاهری از ejs و زبان های پایه طراحی وب : css و jsاستفاده می شود. به این دلیل که جهت ارتباط با سمت server (node js) از سرعت نسبتا خوبی برخوردار می باشد و جهت سئو نیاز به زبان دیگری ندارد .

مدیریت پروژه: **Git – Github**

تمامی فایل های این پروژه با کمک دستورات گیت برای دسترسی تمامی توسعه دهندگانی که در این پروژه فعالیت می کنند به صورت جزئی در گیت هاب ذخیره می شود.

سرویس ها و ماژول ها : **python**

جهت پردازش دیتا و محاسبات لازمه پروژه از جمله (data science) ، از زبان پایتون استفاده شده و با سرعت مناسبی که این زبان در زمینه علوم تحقیقاتی و کار با دیتا دارد ، فعالیت های اصلی این پروژه مانند کار با دیتا های ورودی با آن انجام می شود .

- معماری نرم افزار M.V.C pattern :

به منظور توسعه بیشتر این برنامه، توسعه دهندگان باید از ساختار: M.V.C و الگوی Client-Server برای مدیریت بهتر و دسترسی آسان تر به کدها استفاده کنند. الگوی Client-Server برای اتصال کلاینت ها با سرور در پروژه استفاده می شود. توسعه‌دهندگان باید در مورد هر فعالیتی که برای پروژه انجام می‌دهند نظر بدهند، چه سمت سرور، چه سمت کلاینت یا بقیه توسعه‌دهندگان. توسعه دهندگان سمت سرور باید به طور کامل دلیل استفاده از API ها را هنگام استفاده از آنها توضیح دهند.
نکته ای که همه توسعه دهندگان این پروژه باید بدانند این است که بهینه بودن و سرعت بالای این اپلیکیشن رکن اصلی این ساختار است که باید رعایت شود.

- Test Strategy : System Tests - Integration Tests - Unit Tests :

1. //تست واحد : تست عملکرد صحیح بلوک های کوچک کد منبع نرم افزار که بر اساس آن عملکرد صحیح هر قسمت از کد ارزیابی میشود.

2. // تست یکپارچه سازی : ادغام بلوک های کوچک کد با یکدیگر و تشکیل یک سری ماژول، سپس ادغام چندین ماژول با یکدیگر و آزمایش عملکرد کلی آنها، به طوری که در نهایت اطمینان حاصل شود که اجزا یا ماژول هایی که توسط توسعه دهندگان مختلف در یک تیم نرم افزاری به درستی با یکدیگر کار می کنند. ارتباط برقرار می کنند.

3.//تست سیستم : توسط آن یک نرم افزار یا اپلیکیشن کامل در دست بررسی است تا بررسی شود که آیا نرم افزار ما در زمانی که در دسترس کاربرانش قرار می گیرد به درستی کار می کند و نیازهای آن ها را برآورده می کند.

! این فرآیند در مرحله اتمام توسعه و بروزرسانی تکمیل خواهد شد !

-زمان اجرا:

این بخش شامل جزئیات و تکامل برنامه با کمک نمودار است.

به طور کلی تمامی فاکتورهای این اپلیکیشن شامل موارد زیر است:

-رفتار کاربر: شخص واقعی که اپلیکیشن برای او طراحی شده است . کاربر با ارسال دیتا خود به وبسایت ، می تواند به نتایج محاسبات آنان توسط شاخص های تعریف شده را مشاهده نماید.

-اپلیکیشن کلاینت: این عامل شامل پلتفرم های این اپلیکیشن (وب اپلیکیشن) است که کاربر می تواند با آن ها تعامل داشته باشد.

-سرور: سروری که شامل بررسی، محاسبات منطقی و ارتباط بین پایگاه داده، برنامه کلاینت و APIها می باشد.

-رابط ها و API ها: سرویس و رابط هایی ک جهت محاسبات و پردازش دیتا استفاده شده اند

-پایگاه داده: -

- Sequense

- استقرار :

نحوه نصب اپلیکیشن و قسمت های مختلف آن در ساختارهای فیزیکی در نمودار زیر طراحی شده است. کدهای نوشته شده و تست شده در سمت سرور + کدهای سمت کلاینت ، پس از تکمیل بر روی هاست مستقر می شوند. اجزای خارجی و API ها از طریق واسط ها به هاست متصل شده و به پروژه اضافه می شوند.

- Deployment

- همکاران:
- مدیر پروژه : دکتر جلیلی
- **Full stack developer & Software Engineer** : محراب آقایی